Discovery and Explainability of Fine-Grained Structures in Large-Scale User-Generated Data Sets

Dissertation zur Erlangung des Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat)

der Fakultät für Angewandte Informatik
der Universität Augsburg

Johannes Kastner
University of Augsburg

1. Referee: Prof. Dr. Peter M. Fischer

2. Referee: Prof. Dr. Elisabeth André

Day of Defense: 04.01.2024

Discovery and Explainability of Fine-Grained Structures in Large-Scale User-Generated Data Sets

Johannes Kastner
University of Augsburg

Ich brauche keine Melodie
Und keinen Takt
Um weiter zu tanzen (in meinem Film)
Das Drehbuch ist vertrackt

Callejon - *Kind im Nebel*

To my family.

Abstract

Since the beginning of the 21st century, the exploration of ever-growing data sets has gained more and more attention in research and application-related data analysis. In particular, recommender systems-related use cases due to social networks and media, as well as increasingly popular services in online shopping and marketing, became of specific interest. Moreover, entertainment media such as music and video streaming services and many associated communities and forums are also services that collect large amounts of user-based data. Analyzing correlations, structures, and groups based on various characteristics in sometimes enormous data sets, e.g., place targeted, user-based advertising or making recommendations for music and movies, is essential. In addition, analyzing user behavior and interactions in social networks and media is a crucial area in research to create traceability and understanding of behavior. Also, changes in user profiles over time and correlations between user behavior and news propagation paths are other significant areas in research. Giving structure to this amount of data and extracting relevant results requires human expertise. However, this is very expensive for humans, as it is very time-consuming to identify commonalities and differences in structures and patterns for individual specific data sets.

In particular, identifying specific user roles in social media and networks has taken on a special significance in the last 20 years, as the proportion of bots, spammers, or users who otherwise engage in harmful behavior has increased significantly. Moreover, in addition to these user roles, many other classes of users exist that are distinguished from other users by their behavior on the network and fine-grained characteristics. While the focus in research and practice has so far been on generalized user roles, such as detecting harmful user behavior, fine-grained identification has largely fallen by the wayside due to the need for expert input and transferability to other data sets and the associated effort. Furthermore, taking the rise of influencers as an example, the development of user roles over time, among other things, is a worthwhile but still largely unexplored topic.

In this work, the use of human expertise for the recognition and transferability of patterns and structures in the context of known Machine Learning (ML) methods will

now be applied and gradually reduced. In particular, the refinement and classification of generalized user roles into fine-grained structures benefit from a largely automated and scalable process. Furthermore, traceability aspects serve as substantial knowledge gains, especially at the beginning of the analysis, to enable transferability to new scenarios. In the process, users distinguished by many conspicuous, partly complementary characteristics, such as their actions in the social network, their position, and their ability to influence other users, are first grouped comprehensibly. Subsequently, a trained and supervised classifier assigns each cluster a probability to the existing user roles. The method excites as it can be successfully applied to datasets that are temporally and thematically distinct from the original dataset. Further research also shows that transferability to completely new datasets with a different origin is possible with little effort. Different sampling strategies are investigated to successfully analyze datasets in terms of scalability and stability of user roles and are combined probabilistically afterward. Moreover, a transition model is presented, which can make predictions for users in previously unexamined datasets in a temporal context to investigate longer-term trends regarding user role migrations.

The evaluation results show that many stable distinct user roles are reliably detected, that transferability concerning topical and temporal influences is possible with small cutbacks, and that transferability to entirely new data sets can be successfully implemented with moderate effort. The results of the transition model also show that a large number of users can be predicted reliably to a large extent. Ultimately, all of these aspects also ensure that the approach can cope with a wide variety of data sets in terms of scalability and, with minor drawbacks, hardly relies on the need for expert input.

In addition, the transferability of the approach to datasets representing cascades of user messages as a graph is also carried out in the context of this work. Compared to user role analysis, similar graphs are summarized by various largely hidden properties, with the difference that a Deep Learning (DL) procedure is performed. The evaluation of this use case also shows that parts of the model work on entirely different scenarios and that knowledge can also be extracted and analyzed based on patterns. Furthermore, the transferability also allows an enormous saving of human resources.

Moreover, an approach is presented to minimize the costly and tedious data preparation process by integrating normalization and standardization into a clustering procedure. Again, as with fine-granular user analysis, the primary goal is to cluster common structures and abstract them from others to save human resources.

This thesis presents methods for recognizing fine-grained structures in diverse scenarios, abstracting them successfully, and analyzing them with minimal expert input. In particular, the gain in knowledge and the traceability of how structures emerge during the analysis confirm the usefulness of the methods. Furthermore, these approaches are strengthened in their significance by scalability and transferability.

Zusammenfassung

Die Erforschung von riesigen, immer weiter wachsenden Datensätzen hat seit Beginn des 21. Jahrhunderts durch die sozialen Netzwerke und Medien, sowie auch durch immer populärer werdende Dienste im Bereich des Online Shoppings und Marketings immer mehr Beachtung in der Forschung aber auch in der anwendungsbezogenen Datenanalyse in Zusammenhang mit Empfehlungssystemen gefunden. Auch Unterhaltungsmedien wie Musik- als auch Video-Streaming Dienste und viele damit verbundene Communitys und Foren sind Dienste, die große Mengen an nutzerbasierten Daten sammeln. Dabei ist es essenziell, anhand vieler sehr unterschiedlicher Eigenschaften in teils sehr großen Datensätzen Zusammenhänge, Strukturen und Gruppen zu analysieren, um beispielsweise gezielt und nutzerbasiert Werbung zu platzieren oder Empfehlungen für Musik als auch Filme zu unterbreiten. Darüber hinaus ist auch die Analyse von Nutzerverhalten und Interaktionen in sozialen Netzwerken und Medien ein sehr ausschlaggebender Bereich in der Forschung um Nachvollziehbarkeit und Verständnis für das Verhalten zu schaffen. Auch die Veränderungen von Nutzerprofilen im Laufe der Zeit, sowie die Korrelationen zwischen Nutzerverhalten und Ausbreitungspfaden von Nachrichten zu verfolgen sind weitere bedeutende Bereiche in der Forschung. Um dieser Menge an Daten Struktur zu geben und relevante Ergebnisse zu gewinnen wird menschliche Expertise benötigt, die jedoch für Menschen sehr teuer ist, da es sehr zeitaufwendig ist, Gemeinsamkeiten und Unterschieden in Form von Strukturen und Mustern für einzelne spezifische Datensätze ausfindig zu machen.

Insbesondere die Identifikation von spezifischen Nutzerrollen in sozialen Medien und Netzwerken hat in den letzten 20 Jahren einen besonderen Stellenwert eingenommen, da der Anteil an Bots, Spammern oder Nutzern, die anderweitig schädliches Verhalten an den Tag legen, sehr stark zugenommen hat. Darüber hinaus existieren neben diesen Nutzerrollen auch viele andere Klassen von Nutzern, die sich durch ihr Verhalten im Netzwerk und zu anderen Nutzern von diesen durch feingranulare Eigenschaften abheben. Während in der Forschung und Praxis der Fokus bislang auf generalisierten Nutzerrolen, wie beispielsweise der Erkennung von schädlichem Nutzerverhalten lag, blieb die feingranulare Identifikation aufgrund der Notwendigkeit des Einsatzes von

Experten und der Übertragbarkeit auf andere Datensätze und dem damit verbundenen Aufwand, bislang weitestgehend auf der Strecke. Ferner ist am Beispiel des Aufstiegs der Influencer unter anderem auch die zeitliche Entwicklung von Nutzerrollen ein sehr interessantes, aber noch weitestgehend unerforschtes Thema.

Im Rahmen dieser Arbeit soll nun der Einsatz von menschlicher Expertise für die Erkennung und Übertragbarkeit von Mustern und Strukturen im Zusammenhang mit bekannten ML Verfahren eingesetzt und schrittweise reduziert werden. Insbesondere die Verfeinerung und Klassifikation von generalisierten Nutzerrollen in feingranulare Strukturen profitiert von einem weitestgehend automatisierten und skalierbaren Prozess. Darüber hinaus dienen vor allem zu Beginn der Analysen die Aspekte der Nachvollziehbarkeit für wichtige Erkenntnisgewinne um eine Übertragbarkeit auf neue Szenarien zu ermöglichen. Im Prozess werden zunächst Benutzer, die sich durch eine Vielzahl von auffälligen, teils komplementären Eigenschaften wie deren Aktionen im sozialen Netzwerk, deren Position, sowie deren Eigenschaft andere Nutzer zu beeinflussen in nachvollziehbarer Art und Weise zusammengefasst. Im Anschluss daran erhält jeder dieser Cluster mithilfe eines trainierten und überwachten Klassifikators eine Wahrscheinlichkeit zu den vorhandenen Nutzerrollen. Das Verfahren besticht dadurch, dass es auf Datensätze, die sich temporal und thematisch vom Ursprungsdatensatz abheben, erfolgreich angewendet werden kann. Weitere Untersuchungen zeigen auch, dass die Übertragbarkeit auf komplett neue Datensätze mit anderem Ursprung mit geringem Aufwand möglich ist. Um auch Datensätze erfolgreich im Hinblick auf Skalierbarkeit und Stabilität von Nutzerrollen analysieren zu können, werden verschiedene Sampling- und Kombinationsstrategie untersucht. Außerdem wird ein Transitionsmodell vorgestellt, welches im temporalen Kontext in der Lage ist, Vorhersagen für Nutzer in bislang nicht untersuchten Datensätzen eine Vorhersage für erwartete Nutzerrolen zu treffen, um auch längerfristige Trends hinsichtlich Nutzerrollenwanderungen untersuchen zu können.

Die Ergebnisse der Evaluation zeigen, dass eine Vielzahl an stabilen unterschiedlichen Nutzerrollen zuverlässig erkannt werden, dass die Übertragbarkeit hinsichtlich thematischer und zeitlicher Einflüsse mit kleinen Abstrichen möglich ist, sowie dass die Übertragbarkeit auf komplett neue Datensätze mit moderatem Aufwand erfolgreich umgesetzt werden kann. Auch die Ergebnisse des Transitionsmodell zeigen, dass eine Vielzahl an Nutzern weitestgehend zuverlässig vorhergesagt werden können. Letztendlich sorgen all diese Aspekte auch dafür, dass der Ansatz hinsichtlich Skalierbarkeit mit unterschiedlichsten Datensätzen zurechtkommt und mit geringen Abstrichen kaum auf die Notwendigkeit des Einsatzes von Experten angewiesen ist.

Außerdem wird im Rahmen dieser Arbeit auch die Übertragbarkeit des Ansatzes auf Datensätze, die Ausbreitungsgraphen von Nachrichten repräsentieren, vollzogen. Verglichen mit der Nutzerrollenanalyse werden hier ähnliche Graphen durch eine Vielzahl von weitestgehend verborgenen Eigenschaften zusammengefasst, mit dem

Unterschied, dass hier ein Deep Learning Verfahren vollzogen wird. Auch die Auswertung dieses Anwendungsfalles zeigt, dass Teile des Verfahrens auf komplett anderen Szenarien funktionieren und dass ebenfalls Wissen anhand von Strukturen extrahiert und analysiert werden kann. Darüber hinaus ermöglicht die Übertragbarkeit ebenfalls eine enorme Einsparung von menschlichen Ressourcen.

Ferner wird unter anderem ein Ansatz vorgestellt, um den sehr aufwendigen und langwierigen Prozess der Datenaufbereitung zu minimieren, indem Aspekte der Normalisierung und Standardisierung in ein Clusteringverfahren integriert werden. Auch hier ist, wie bei der feingranularen Nutzeranalyse das vorrangige Ziel, gemeinsame Strukturen zusammenzufassen und von anderen zu abstrahieren sowie, dass dadurch menschliche Ressourcen eingespart werden können.

Die in dieser Arbeit vorgestellten Verfahren zeigen allesamt, dass es möglich ist in mannigfaltigen Szenarien mit teilweise komplett unterschiedlichen Ausgangssituationen hinsichtlich der verfügbaren Datensätze, feingranulare Strukturen zu erkennen, diese erfolgreich voneinander zu abstrahieren, sowie mit möglichst wenig Zeitaufwand von Experten zu analysieren. Insbesondere der Erkenntnisgewinn und die Nachvollziehbarkeit, wie Strukturen im Laufe der Analysen entstehen, bestätigen den Nutzen der Verfahren. Darüber hinaus werden diese Ansätze durch die Aspekte der Skalierbarkeit und Übertragbarkeit in ihrem Stellenwert verstärkt.

Acknowledgments

The composition of a dissertation is an almost endless journey and a rollercoaster of emotions, which prompts the author to express gratitude to companions for their support. First and foremost, I sincerely thank Prof. Dr. Peter M. Fischer for granting me the opportunity to work in his research group and under his guidance to create this dissertation. I have always valued his helpful feedback and beneficial discussion sessions. Without his assistance and his visions, this would not have been possible. Furthermore, I would also like to thank Prof. Dr. Elisabeth André for taking on the second evaluation for this dissertation.

A big thanks go to Prof. Dr. Werner Kießling and Prof. Dr. Markus Endres, who provided me with the opportunity to take my first steps in the world of science and work in their research groups. Through their involvement, I gained the motivation to embark on an academic career. In addition, I would like to express my gratitude to Dr. Florian Wenzel, who, as a supervisor, sparked my interest in scholarly work during numerous projects throughout my studies.

Also, a great thanks goes to my family, especially my parents and siblings, who supported and encouraged me throughout the creation of this dissertation. Without their involvement, the realization of this work would not have been possible. I would also like to thank my friends who, despite the limited time during the creation of this dissertation, provided encouragement and helped shift my focus when necessary.

Lastly, I would also like to extend my gratitude to all my (former) colleagues and companions at the Chair of Databases and Information Systems, Prof. Dr. Bernhard Möller, Prof. Dr. Martin E. Müller, Dr. Lena Rudenko, Dr. Andreas Zelend, Dr. Patrick Roocks, Felix Mack, Jennifer Neumann, Vincent Le Claire, and Elisabeth Czerwenka for their collaborative work, pleasant conversations, and engaging discussions throughout the last years.

Contents

Acronyms

AI Artificial Intelligence

AL Active Learning

DL Deep Learning

DM Data Mining

ET Extremely Randomized Trees

GBM Gradient Boosted Decision Trees

IMDB Internet Movie Database

KD Knowledge Discovery

KNN K-Nearest Neighbor

LDA Linear Discriminant Analysis

ML Machine Learning

PCA Principal Component Analysis

RL Reinforcement Learning

SL Supervised Learning

SSL Semi-Supervised Learning

SVM Support Vector Machines

USL Unsupervised Learning

Part I

Motivation and Background

Chapter 1

Introduction

I'm swinging hard for all my life
Reaching across this great divide
Until I see the other side
Until I die, until I die

Parkway Drive - *Darker Still*

1.1 Motivation

User-generated data has become more and more significant in the last few years. Especially the purpose of sharing personal and public information and interacting with other users on social media platforms led to a continuously growing number of users and their produced content, leading to a vast amount of data to analyze. The importance and significance of user-generated data are reflected in online social networks and users in the real world, as information is discussed and spread offline in daily routines: Whether discussions of significant sports events, political elections and topics, and the latest movies regularly reach and affect a vast part of today's society. This massive amount of user-generated data allows experts to analyze and categorize users by their behavior and interaction in digital platforms, which is a significant component of this work.

The behavior of users in those kinds of data sets became an extensive area in research and several regions of the economy, as analyzing it is an eminently significant benefit. Keeping track of the immense, ever-growing data is challenging, making it difficult to understand individuals, groups, and products in the data sets provided. Experts,

researchers, and data scientists often struggle with this vast amount of data as they must process and structure it before an analysis is feasible.

In a commercial setting, social media benefited from the trends beginning in the late 2000s, as user-generated data also exists in many other parts of digital daily routines, such as e-commerce. For example, micro-targeting, describing a segmentation process of a group of users with similar characteristics such as specific interests, demography, or residence to deliver them custom-tailored content, is very important in online shopping and social networks. The granularity adjustment is significant in this use case, as users could get bored by coarse-grained non-personalized ads and get frightened by too fine-grained personalized ads (cp. [Bar14]). Moreover, detecting fine-grained user roles is also applicable to the behavior and interaction of users in social networks to distinguish between different kinds of user roles. The work presented in this thesis picks up this trend. It aims to find fine-grained and hierarchically built structures using independent dimensions. A major, but not the only, contribution is to detect stable and distinguishable user groups starting from coarse-grained structures. In literature, granularity in terms of granular computing is present in both ways, from coarse to fine-grained and vice versa, being a prominent area in data clustering and cluster analysis, which is discussed in the survey of Ding et al. [DDZ15].

Valuable strategies for the aims of this thesis can be addressed with both Unsupervised Learning (USL) and Supervised Learning (SL) techniques from the area of Machine Learning (ML) as some steps need further human interaction and intervention, while others only need them as catalysts in the beginning. Even though many areas in ML do not require human intervention, e.g., the plain unsupervised user group detection, the analysis and classification of structures need at least human assistance in the beginning and supervisory authority in the learning process of the classification. Thus, the use of human expertise respectively intervention in the different and manifold areas of ML is a significant aspect, which will be discussed later in Section 1.4. Both the human intervention for amendments in Knowledge Discovery (KD) approaches and expertise in model building or training models are helpful but expensive in terms of time and effort. This thesis will examine human expertise's benefits, challenges, and drawbacks in each step of the approach.

The individual aspects of this work will now be explained and motivated, considering how their interaction creates a unique set of challenges.

First, large-scale user-generated data sets consist of user information, information considering behavior to other users as well as to non-human entities, such as (digital) products, e.g., movies. In [KDN08], user-generated content is described as content that has its origin from regular people who voluntarily interact with (non)-human entities and contribute data and information, e.g., people rate movies, videos, music, or interact with the content of other people. This data is then utilized in social networks,

online shopping platforms, and other places. Frequently, these user-generated data sets are massive in terms of data points and distinct features and, thus, difficult to interpret from the view of humans. These aspects outline ambitious tasks and cause challenges for researchers and data scientists, such as finding groups, structures, patterns, connections, and differences, often leading to overwhelming situations and challenges in the analysis process. Since it is impossible to avoid human effort, one of the main aspects of this work is to minimize the effort to unburden experts. These tasks will be discussed later in related work in Section 4.8.

In this work, data sets from the Social Media platforms Twitter[1], which was recently renamed to X, and Telegram[2], which contain user information and user behavior between pair of users considering their interaction and content, will be analyzed. Twitter and Telegram are well-known social media platforms that deliver vast amounts of user-generated data. Analyzing data sets from two distinct and miscellaneous kinds of social networks to analyze user features representing different user behaviors and interactions substantiates the variety and adaptability of the approach presented in this thesis. Furthermore, a data set from the Internet Movie Database (IMDB)[3], which contains data on movies, actors, directors, or user ratings, is part of an approach to reduce human intervention in terms of preprocessing.

Second, analyzing and detecting fine-grained structures also play an essential part in this work. Fine-grained structures can be distinguished only by a few characteristics or features, sometimes only negligible, from others. These aspects play an essential role in cluster analysis as the way from coarse to fine-grained structures is a manifold research area, discussed in the survey of Ding et al. [DDZ15]. This work analyzes data sets with a variety of features in most of the occurring use cases, where a widespread comprehension of the given data and their patterns is necessary. Ensuring a more fine-grained level considering the structures in the analysis process, straightforward and complex features, but also latent and explicit features, are considered. These features are needed to describe users and summarize similar groups, leading to modeling user roles. Many well-known user roles in the literature have a manifold description, which must be mapped on those fine-grained structures covering as many features as possible, creating known and new user roles. While most of the related research (cp. Section 2.1.2 and 4.8) focuses only on a few coarse-grained structures, which incorporates only the recognition of limited features or characteristics, this work targets explicitly the identification of fine-grained structures as well as their understanding and the comprehensibility w.r.t. their origin and composition within a clear hierarchical structure among the classes. Finding these structures in a fine-grained manner compared to a few coarse-grained classes also distinguishes the number of classes,

[1] https://twitter.com
[2] https://telegram.org
[3] https://www.imdb.com

which is, in most cases, a lot larger depending on features and use cases. Summarizing data objects to structural mapping of groups can also lead to two or more best-matching classes due to character deviation. Thus, it is also necessary to consider a probabilistic n-class problem for each pattern to map. Distinguishing from the previously mentioned approach of micro-targeting, where users are characterized distinctly by a set of features, i.e., whether a feature is fully present, this approach aims no hard allocation as allocating user roles may often not be precise. Since the use of human expertise is scarce and expensive in terms of time and effort and forces increased demand, in this work, the use of human knowledge will take place in a limited way when it is valuable and necessary aiding to reduce human expertise over time to facilitate an automated model-based process as far as possible.

Third, a vital component in this research is the significance of explainability and discovery of fine-grained structures, which emphasizes the additional value of fine-grained structures in large-scale user-generated data sets. Understanding how results are achieved is essential for researchers and analysts. ML techniques often provide only results for a given input, which are difficult to comprehend. In contrast to classic KD approaches, the approach presented in this thesis appropriates all steps of a transparent strategy to understand the process and the outcome. Since fine-grained structures may only differ measurably in a few features, it is necessary to aid the analyst reasonably and well-structured through each analysis step. In most aspects, human analysts and experts' central part involves supervision during and after the analysis, especially at the beginning of analyzing new data sets. Onwards, while exploring more and more data sets, the central part of analysis consists of monitoring and comparing results against a ground truth, which is associated with less effort in time and resources compared to the previous steps.

Finally, the challenges of this work, mentioned in the paragraphs before, considering the benefits of a fine-grained structural analysis in large-scale user-generated data sets, are summed up briefly. This work provides a cohesive and thus comprehensible approach to understanding each step of the novel KD process, which will be introduced in the following section. Furthermore, it aims to recognize even fine-grained patterns in vast and complex data sets. A pre-eminent aspect, which is emphasized more or less in each part of this approach, is the need for human expertise and the opportunity to conserve resources and time the more data sets are analyzed. Moreover, it is of significant interest to focus on the behavior of users in several aspects of today's digital routines, such as online shopping or online social communities, in particular, social networks and media and other platforms.

While the previous aspects concentrate mainly on the analysis of disconnected data sets within a social network, it is also of peculiar interest to transfer the model to other data sets stemming from other social media platforms and social communities, such as

video and music streaming portals, with the benefit of recommending specific content to fine-grained user roles. In addition to the approach presented in this thesis, the ability to track user role changes over time in data sets stemming from the same source also provides the possibility to transfer such kinds of models to other scenarios in the social-political area, such as election forecast and shift of votes, which is a well-known scope in aspects of political elections. The insights of detecting fine-grained structures in this approach may also pave the way into other directions, such as online shopping and e-commerce, as the possibility to customize promotions to specific user groups depending on their behavior in the platform is possible.

Not only does the comprehension of social behavior or interaction with other users require the need for fine-grained structures such as user roles, but also the diffusion of messages in information cascades and their correlation with fine-grained structures are worthwhile research topics. The ability to cluster users and whole cascades of user messages, represented as graphs, is a worthwhile research area. This topic significantly enhances the related research regarding coarse-grained user role analysis, as cascade structures also exhibit different graph shapes representing a lucrative research area.

1.2 Overview of Approach & Methodology

To specify the problem statement and the contributions of this thesis more in detail, a brief overview of the approach and the methodology is presented in this section, as can be seen in Fig. 1.1. At first glance, the approach resembles the classic KD process for databases as presented in [FPSS96], yet there are some crucial extensions and innovations.

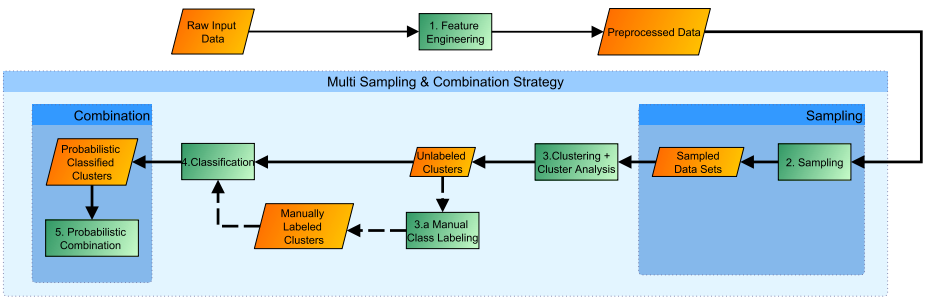


Figure 1.1: Flowchart of the novel KD approach.

In the first and most conventional step of the approach, on a *raw data set* consisting of

several data points with feature values, all kinds of *Feature Engineering* are addressed, such as filtering data or normalization and standardization techniques to reach accurate and comparable data. Regardless, preprocessing is a time-consuming but vital step, requiring careful observation and analysis. Moreover, general aspects of preprocessing are also easily adaptable to new data sets.

Sampling is a cross-cutting technical aspect encompassing the following stages of the approach, which will be presented after the remaining steps. The third step of the model deals with *Clustering* and *Cluster Analysis* to group similar items and delimit them from other clusters. Thus, each cluster remains *unlabeled* until the *Classification* step. While human intervention in *Clustering* is initially vital, as finding and tuning parameters needs to be done only in the beginning, *Cluster Analysis* is a step that requires careful observation, as *Clustering* does not produce a straightforward output each time. Once a clustering strategy and a sense for cluster evaluation metrics are found, the adaptability to other data sets is easily manageable. Delimitating from those approaches, finding more fine-grained instead of coarse-grained structures, and comprehending each step, especially in *Cluster Analysis*, break new grounds in this work.

After *Clustering* and producing a set of *unlabeled* clusters, an optional step, *Manual Class Labeling*, is followed, especially when applying the pipeline to entirely new data sets. After that, the manually labeled and unlabeled cluster deal as input for the *Classification* step where unlabeled clusters receive a label. *Classification* is the most expensive step in terms of both human intervention and transferability, as training data and a ground truth need to be specified, and the classifiers need to be trained. Transferability is necessary depending on the data source, as data sets stemming from the same source can get around with training data as features and specifications are similar. Thus, much time is saved when reusing several training data sets for topically related or close-in-time data sets, making the proposed approach valuable and effective. While most traditional approaches in KD focus on either *Clustering* or *Classification*, this approach considers both techniques consecutively. Moreover, classifying fine-grained structures as a part of a multi-class problem leading to probabilistic vectors is not prevalent in most related procedures as it requires a lot of human resources. Significantly, SL techniques benefit from more recent techniques, such as Active Learning (AL) and Semi-Supervised Learning (SSL), due to a significant reduction of human involvement, as experts conduct as a corrective.

As a cross-cutting concern, *Sampling* represents the first significant extension of the proposed approach using several strategies, which will be introduced later in Section 2. While *Sampling* is described in work published since 2010, such as [ZAL14], as the purpose for scalability issues for the continuously growing data sets in social media use cases, the lack of lower accuracy and validity is a clear drawback. Even though

sampling has become an increasingly important strategy, it reveals new challenges as data sets become more difficult to manage. Sampling is beneficial in this work, preventing the drawbacks of classical sampling, as the whole process from Steps 3 to 4 is applied to each sample as part of the novel *Multi-Sampling and Combination Strategy*, revealing clear advantages in certainty and stability of classes. Saving time in the whole process, mainly due to reducing the size of data objects to representative samples in terms of the quadratic complexity of hierarchical clustering, is the most critical aspect of sampling in this thesis. In addition, a unique feature of the novel *Multi-Sampling and Combination Strategy* is to guarantee coverage of the whole data set, which is a well-known problem in massive data sets, as such data sets cannot be processed in one step due to the complexity of running runtime and memory. The most crucial aspect of this work is the possibility that sampling, and combination provide certainty and stability in classifying classes. Human experts only need to choose valuable strategies that can easily be adapted to all kinds of data sets. Once convenient methods are found, human intervention is not required anymore. Only finding suitable sample sizes is substantial for the following steps, as too large and too small data sets can cause issues.

The output of the *Classification* step for each sample delivers a probabilistic labeling for each cluster in terms of a vector. These probabilistic class vectors for each data point represented in clusters from different samples are combined in the combination step to achieve stability. It is possible to reach for each data point of the input a stable probabilistic classification in the output. This final step of the pipeline is an entirely new attempt at classifying objects, ensuring more stability and certainty in finding patterns for specific data objects, as each can have a slightly different probabilistic classification in varying samples.

Finally, this approach has a lot of common steps as traditional KD and Data Science pipelines but also strikes a new path, especially considering the novel *Multi-Sampling and Combination Strategy* that guarantees to find stable and specific patterns, which is essential for the analysis of fine-grained structures, the evolution of those as well as the portability of this approach to new data sets. Especially the latter aspect is particularly substantial, as the transferability of single steps such as clustering and classification and the whole approach leads to minimizing human effort and expertise when applying it to (entirely) new data sets.

1.3 Contributions

After introducing the overall direction of this thesis, giving an overview of the approach and methodology, and motivating the content of this work, the most important contributions will be presented and discussed briefly in this section.

- The main goal of this work is to provide a framework for fine-grained structural analysis of data sets stemming from social media. While the coarse-grained analysis was a very established topic in related research (cp. Section 4.8), this work extends the detection of coarse-grained classes by refining them comprehensibly and traceably. Nevertheless, some approaches concentrate on finer-grained analysis. However, most of them consider only a limited number of classes or focus on analyzing a few features. The approach presented in this thesis allows a lucid, well-structured, and comprehensible procedure by applying a hierarchical and probabilistic strategy that enables refinements for several clusters but also classes that need to be pre-defined. Moreover, some results of the classification step show affinity to several classes, which augments a fine-grained structural analysis to define them precisely. Both learning the structure of fine-grained patterns, such as user roles as well and the assignment process in terms of suitable labels are covered in this thesis.

- Furthermore, a significant contribution of this work is the ability to adapt the proposed framework with moderate effort, employing human intervention in new unknown data sets. The first scenario is the adaptability of scalable and complementary data sets from the same source. The ability to adapt the model on (entirely) new data sets only with fewer and more suitable variations illustrates the feasibility of the recognition and transferability of knowledge over time as well as topic variations. As scalability addresses the variation of sizes of data sets, sampling plays a pivotal step in this thesis. The comprehensive experiments in Chapter 5 approve the functionality and transferability of the approach and the benefits of fine-grained probabilistic classes considering topic and time variations.

- As the framework works successfully on a specific data set, it is of significant interest that it also can handle data sets stemming from other sources, which augments the contribution of the adaptability of the proposed model as a second scenario. This thesis provides, on the one hand, a transfer of the model to a new social network with similar but varying features, on the other hand, a transfer to a completely different scenario, where cascade shapes are identified and classified. In this scenario, the challenge in the process is that features are latent and elusive for humans and thus can hardly be tracked and comprehended. The experiments and analysis in Chapter 6 and 7 confirm the transferability on ultimately other data sets, show the versatility, and amplify this approach's previously mentioned benefits.

- Last, one fundamental contribution of this work is the reduction of human intervention. This thesis provides an approach that needs human experts,

especially initially, for supervision and learning. However, the more data sets were supervised, the less human intervention and maintenance were required for the following analysis. In particular, experts can focus more on understanding the process and the subsequent analysis. A rather eminent aspect is simulating and predicting user roles using models based on analyzing users and user roles in terms of a time series of related events in Section 5.8. This model-driven approach can cut short the whole process of analyzing users and user roles from Fig. 1.1 as the extensive analysis in Section 5.9 reveals.

- Another somewhat orthogonal approach for clustering data sets is a novel clustering approach exploiting the Borda Social-Choice voting rule in the allocation process instead of allocating data objects to clusters using traditional distance metrics in Chapter 3. The most outstanding benefit is that a normalization and standardization of data sets are not required, and thus, experts can save much time in the step preprocessing. Moreover, the extensive experiments show the suitability of this approach in terms of a Pareto-optimal use case such as recommendations.

1.4 Assigning Contributions to ML Areas

The KD approach from Section 1.2, as well as the contributions stated in the previous section, include several techniques and approaches from the area of ML. Thus, first, a general overview of KD will be presented. After that, the focus will be set more detail on the contributions of this thesis, and their interaction with the most essential areas of ML will be examined and brought into line, while the most crucial areas of ML will be introduced later in Section 2.5.

Maimon and Rokach describe KD in Databases as an "*automatic exploratory analysis and modeling of large data repositories*" to identify "*valid, novel valuable and understandable patterns from large and complex data sets.*" Data Mining is crucial for KD. It uses algorithms to explore datasets, build models, and identify unknown structures and patterns to understand phenomena and make predictions. The analysis of massive datasets requires techniques, approaches, and algorithms from the field of Data Mining (DM) [MR10]. In Awad and Kanna, KD is defined as an "*extraction process where knowledge is gathered from structured and unstructured data sources to create a knowledge database for identifying meaningful and useful patterns from underlying large and semantically fuzzy data sets.*" KD combines algorithms and concepts of ML with statistical metrics and methods to solve user-oriented queries and issues. The extracted knowledge is used repeatedly as input for new data to produce a new output, which can also be reused in the knowledge base. In most cases, the procedures of

KD are applied on huge scalable data sets to achieve patterns and structures using several concepts and algorithms of ML such as clustering, classification, dimensionality reduction, and much more, to name only a few of them, which will be introduced in the following section. [AK15]. Finally, scalability, automation, and the ability to find, comprehend, and transfer fine-grained structures and patterns, pivotal in this work, are present in KD and ML.

Considering the fine-grained structural analysis of data sets from Fig. 1.1 focusing on a general KD approach, the first area needed to provide a hierarchical and probabilistic procedure is USL using hierarchical agglomerative clustering to refine coarse-grained structures. Furthermore, cluster analysis is a pivotal follow-up technique that guides the cluster hierarchy to find the best possible clustering. Besides USL, this approach also affords an interaction with SL techniques, such as classifiers, to map each clustered and analyzed result to a set of given user roles in a probabilistic way. This interaction of hierarchical clustering and probabilistic classification substantiates the success of finding fine-grained classes.

Regarding the adaptability and transferability of the approach on data sets stemming from the same source and other sources, mostly preprocessing and data preparation from the area of DM are needed. The expert has to incrementally select and adjust the features using statistical measures if a completely new data set stemming from another social network is analyzed. Suppose the source of the data set is related to already analyzed data sets. In that case, there are only minor adjustments needed in the sampling strategy, clustering, and cluster analysis (USL), and some circumstances in the classification step (SL), as significant deviations in terms of time and topic between the data sets sometimes need manual adjustments. Besides SL, there exist approaches from the area of SSL and AL to support the user in creating training data to reduce human involvement, as manual labeling of entirely new data objects is only needed in the beginning.

Suppose there is a sufficient pool of already analyzed data sets. In that case, using statistical models, it is also possible to predict user role distributions in terms of a long-time analysis for new data sets. Especially for analyzing and processing user and user role movements beyond distinct data sets, Markov Chains deal as a foundation for a model-building process, which may shorten the analysis of new data sets in an existing time series. As Markov Models are not a general step of a typical KD approach, the essential steps and methodology will be introduced later in Section 5.9.

Since reducing human intervention is one pivotal aspect of this work, it is very present in most of the given steps. While the savings of resources in preprocessing and data preparation (DM) are very manageable, as these steps are only needed again if entirely new data sets are analyzed, the clustering and cluster analysis (USL) need some more attention, as the technique is based on statistical values, which needs to be adjusted

based on the granularity of user roles, which can slightly differ between data sets. More human effort is needed to generate training data for the classification step (SL) and ground truth to validate the classifiers. Especially if hardly trained classifiers are available, human resources are needed to create, enrich, and validate new training data.

This work also involves a study to minimize preprocessing steps to save time and human resources. The approach in chapter 3 combines the time-consuming preprocessing step and k-means clustering, which is not part of Fig. 1.1. This almost automatic approach aims to reduce time and human intervention by combining a clustering algorithm from the area of USL with a social choice voting role.

1.5 Thesis Structure

The remainder of this thesis is structured as follows. After the motivation, an overview of the proposed approach, the contributions, and their allocation to areas of ML in Chapter 1, in Chapter 2, relevant definitions are clarified, and the most significant background knowledge of ML techniques, algorithms as well as (statistical) measures are introduced as the last section in Part I.

The main part of this thesis (Part II) includes in Chapter 3 a technique to minimize human intervention in terms of preprocessing right before clustering will be introduced and underpinned with many experiments considering a movie based recommendation scenario. The aim to reduce human intervention is also part of the following Chapter 4 where a model-building process is presented. This process deals with several ML strategies from Chapter 2 in a kind of pipeline to (semi)-automatically cope with massive data sets in a user-labeling process. This chapter is followed by Chapter 5, where data sets from Twitter are analyzed within single data sets, and results are compared to other topically and temporal (non) related data sets. Moreover, a comprehensive long-term analysis of users and user roles is performed, leading to a model-building process to simulate and predict user roles. After that, in Chapter 6, the ability to transfer the model-building process described in Chapter 4 from the Twitter data sets to Telegram data sets is displayed and discussed. The last chapter of the main part (Chapter 7) deals with the analysis of cascade shapes from Twitter data sets and the ability to deploy parts of the methodology from Chapter 4 on them.

In Part III, this work concludes with a summary and classification of the most significant contributions as well as a short outlook on future work in Chapters 8 and 9.

Chapter 2

Background

Ich hebe den Hammer
Und schlage den Meißel
Ich suche nach Erzen
Tief in deinem Herzen

Callejon - *Unter Tage*

This chapter introduces the most important terms and definitions considering social networks and social media as well as user roles. After the terminology, a broad range of statistical and mathematical metrics and measures are presented, which are eminent for the approach presented in Chapter 4. Furthermore, general algorithmic approaches addressing the main steps of the Knowledge Discovery (KD) approach from Fig. 1.1 in Section 1.2 are introduced.

2.1 Definitions

First, conceptual definitions essential for this thesis' work will be presented and clarified. Especially for detecting user roles in social media, it is necessary to introduce social media, the different aspects, and characterizations of the most typical representatives and their purposes. As user roles play a significant role in this work, a basic definition

for the term of a user role is also introduced in this section, while specific kinds of user roles will be presented in Chapter 5 for the Twitter and in Chapter 6 for the Telegram use cases.

2.1.1 Social Networks & Social Media

Social networks and social media play a crucial role in this work since the data stems from social networks or social media. Data sets from social media services are often characterized by several features, which provide many possibilities in terms of analyzing user roles as well as information diffusion of message cascades using Machine Learning (ML) as mentioned in Section 1.4.

In [OW15], *social media services* are defined as "*Web 2.0 Internet-based applications, where user-generated content is created by users and shared with others*". Each user and sometimes user group is represented by a user-specific *profile* maintained by the social media service. Moreover, traditional social media services ensure that profiles are connected *bilaterally* to other users' profiles, such as Facebook, while the focus in Twitter and more recent social media services, such as Instagram or TikTok, is more *follower*-based and thus *unilateral*. Some of the most common social media services these days are the social networks Facebook and Instagram, the microblogging service Twitter, and the social video hosting services YouTube and TikTok [Sta].

Many social media services, such as Facebook, complicate access to data sets due to data protection. Moreover, services such as Instagram or TikTok are not considered because they are driven by non-transparent algorithms, which compound the transparency of connections between users and interactions on messages by users. Only certain services like the microblogging service Twitter and the messaging service Telegram allow analyzing data sets, as manifold data is available.

Another definition of social media services was provided by [May08] back in 2008 as "*services, where users can spread information across societies worldwide within minutes*", as messages can be spread within a few seconds around the world today. Content such as messages, pictures, and videos can be shared with others. However, the interaction with other users, their messages in the network, and the organization of users in different kinds of social structures are also very characteristic of social media. In today's society and daily life, social media services are essential as information can be spread to family and friends, but also to many others around the world within seconds, enabling reactions and discussions that are vital for the analysis of users and their behavior in social media.

In particular, Twitter allows users to *create*, *react*, and *share* content in terms of bare *text*, *images*, or *videos*, leading to conversations that experts can analyze for *user behavior* and *interaction* with others and their *content* [Jav+07]. Unlike many other

social networks, such as Facebook or MySpace, Twitter is not based on *symmetric friendships*, as users can *follow* others. However, the following users can skip the following back. Twitter was also the first social media service that provided *mentions* of users as well as *hashtags*. Also, the possibility to *retweet* content was a new feature in online social networks, which allowed users to spread information more easily beyond the creators' environment. These features characterize the social media service Twitter as more likely to be a news source than a traditional social network [Kwa+10].

In contrast to Twitter, Telegram, released in 2013, is not a typical traditional social media service but more an *instant messaging* platform where users can communicate with other users using *text messages*, *pictures*, or *multimedia services* such as *video* or *voice chats* between two individuals. Besides, it is also possible to create *groups* or *channels* where a single user can share content with many other users, while on Twitter, *messages* are shared *globally*. Compared to Twitter, user-based features are scarce, as there is no information about the *verified* status of a user nor the possibility of adding a *URL*. However, the possibility of communicating with other users, such as responding to messages and forwarding them but also mentioning other users, is given in Telegram, too. These aspects also substantiate the suitability as a data source for analysis, as many features from Telegram are similar to those of Twitter, as seen in the Telegram API[4]. Also, the possibility of creating *bots* for channels that can respond to users' messages is widespread in Telegram. While Twitter has problems with bots, which are flooding spam messages to specific topics, bots in Telegram are used as chat-bots[5].

Considering the data sets in this work, the main reasons to choose Twitter and Telegram are the availability of manifold data, described by user profile features, the position of users in the network as well as interaction with other users, which is all given in Twitter and Telegram.

2.1.2 User Roles

After introducing social media services, it is essential to define user roles, which play a crucial part in social media, and in a fine-grained structural analysis in this work. Users in online social networks share similar *features* but differ in others. Before defining user roles, focusing on specific roles and intentions for using social media services like Twitter is essential. Analyzing intentions provides insights for understanding user roles. Twitter has evolved in the past 15 years with new features like *threads* and *user mentions*. Researchers studied user behavior to improve specific features. Critical insights on finding different types of users will be presented chronologically to describe

[4] https://core.telegram.org
[5] https://telegram.org/blog/bot-revolution

the evolution and specification of user roles. In contrast, depending on the specific use cases, specific user roles will be introduced later in Chapter 5 and 6.

In [Jav+07], the author describes how different kinds of users have shaped the microblogging platform Twitter, especially the *communication* between users and the *intention* of users. Based on the users' intentions, specific user roles can be adapted. One of the main directions of this paper is to understand how and why specific kinds of users use Twitter by studying *topological* and *geographical* properties. Especially in terms of *activity*, considering *following* other users or being *followed* were central aspects of the analysis of [Jav+07]. The distributions for in-degrees, i.e., followers, and out-degrees, i.e., followees, show a relatively similar power-law distribution. This distribution leads the authors to consider Hubs and Authorities using the HITS algorithm by [Kle98]. Users with a high Authority value often have a low Hubs value and thus are defined as users with many followers but only follow fewer other users. In contrast, low Authority and high Hub value define users with fewer followers and follow many others. There are also many users in between, which leads the authors to group the users into different communities using Modularity by [GN02], finding, e.g., users influenced by Hubs. This aspect leads to many different user roles, which can also vary, as users may behave differently in different communities.

Besides the work of [Jav+07], published in 2007 as Twitter did not have the scope of today and thus several properties, e.g., *replies* were not yet possible, there are more definitions of specific user roles such as in [Smi15]. *Self-presentation*, considering athletes, is very common in social media, as they want to improve their *range* and *popularity* by sharing verbal and non-verbal messages to express their identity. This behavior is also noticeable on Twitter, as personal insights into their life using *text messages* or *images* are very common for rising stars and celebrities.

To expand the user intentions made in [Jav+07], the work of [HH09] analyzes the benefits of user exchanges using the possibility of *tagging* them with the @ sign. This sign is vital to identify users who participate a lot in *conversations* after they are *mentioned* or users who are *active* and try to motivate other users to participate in conversations after they are *mentioned*. [Tin+12] focuses on their work on user roles, which have significant participation in conversations w.r.t distribution of messages.

In [BD+14], the focus is also on community detection, as the authors use an extended Markov Stability approach by [DYB10], which can deal with directed networks [LDB08]. They scan the network for communities using a continuous time diffusion process, which reveals different kinds of fine to coarse-grained communities, creating also several user role-alike groups. The flow of information and the degree of reinforcement is a central point in their work. It overviews the different communities built and characterized by location, profession, or topic. Also, patterns of interest considering incoming and outgoing interests in the communities are crucial in this work for defining different

user roles. Moreover, they also consider the degree of *retweets* for the community nodes to define user roles more precisely.

All of the previously presented related work led to various characterizations of user types with entirely different intentions in social networks and social media. This insight is precious to get an overview of these distinct roles by grouping similar users. So, analysis needs to summarize users with similar features to understand how often users are represented in a social network or social media, how users evolve, or the impact on information diffusion of cascade shapes. Nevertheless, defining them by the given features is pivotal before user roles can be summarized. The central aspect of the summation in several groups is to consider not only many individuals but only a few groups in terms of analysis. The term user role does not have a general definition in literature, but primarily, representatives are described by their features and characteristics. [Zyg17] also describes the problem of defining a general definition of user roles, as significant features from data sets need to be extracted to create a valuable summary of the original data set, which is highly dependent on the data set and the purposes of the analysis. Considering user roles, [Zyg17] describes this as "a tool for simplifying patterns of action, distinguishing between different types of users, and understanding human behavior." Thus, for this work, a *user role* is built by a group of *similar users* and is *relevant* and *stable* if this role occurs in an adequate number in a single data set and regularly *reoccurs* in sufficient data sets. For this work, a *user role* is defined as a group of users who share *similar feature values* and are *well separated* from other user groups.

As mentioned in Section 1.3, this work targets the detection of coarse-grained user roles such as spammers or malicious acting users and, furthermore, fine-grained user roles. The general definition of a user role and the specific definitions of different roles in the literature pave the way for a comprehensible, fine-grained structural analysis of user roles.

2.2 Statistical Metrics

Several approaches in this thesis rely on statistical metrics as they are essential tools to describe correlation and similarities of features or whole data sets. Moreover, they help to visualize distributions of features or data sets. Thus, statistical metrics are helpful tools to confirm data sets and their suitability after preprocessing steps such as normalization and stabilization.

2.2.1 Correlation

While preprocessing in a KD model, choosing features carefully for the following steps is essential. In most Supervised Learning (SL) and Unsupervised Learning (USL) techniques, correlation of features, w.r.t. to input data matters, as features that influence others in an intense way sophisticate the results. Therefore, features have to be chosen depending on their correlation to each other. The *Correlation Coefficient* is a well-known statistical measure that indicates a statistical relationship between 2 variables. Thus, it is a pivotal technique that helps data analysts decide whether a feature is needed, depending on the relationship to other features.

There are several methods to specify correlation, especially the term correlation coefficient. The most common is the *Pearson Correlation Coefficient*, used in several implementations, e.g., pandas or numpy, defined as in [Weib].

Definition 1 (Correlation Coefficient)

$$cor(X, Y) \equiv \frac{cov(X, Y)}{\sigma_X \sigma_Y} \tag{2.1}$$

for two variables X and Y. Where $cov(X, Y)$ is the covariance and σ_X and σ_Y is the standard deviation of these two variables.

The *Covariance* for two variables X, Y with sample size N is defined as in [Weia]:

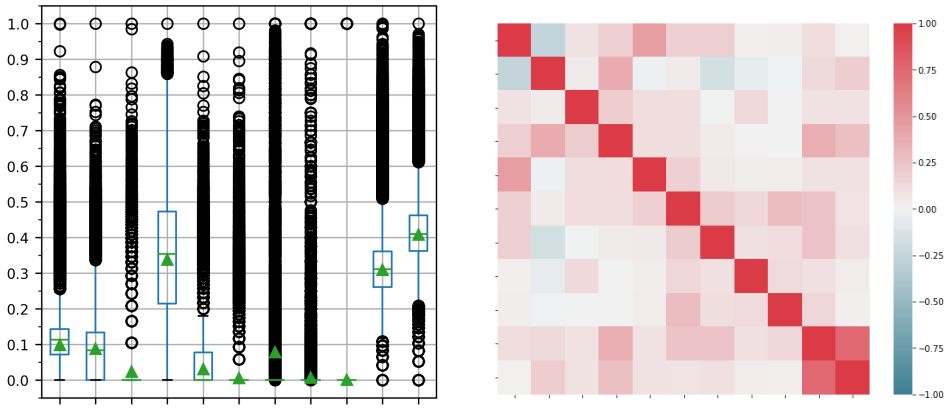
Definition 2 (Covariance)

$$cov(X, Y) = \langle (X - \mu_X)(Y - \mu_Y) \rangle = \langle XY \rangle - \mu_X \mu_Y = \sum_{i=1}^{N} \frac{(x_i - \bar{x})(y_i - \bar{y})}{N} \tag{2.2}$$

where μ_X, μ_Y are the respective means.

It is essential to choose features in a way that they correlate to only a few other features and that there is no high anti-correlation. Both features with a high correlation and a high anti-correlation do not bring benefits using SL, as well as USL algorithms, as they can be distorted, e.g., can embarrass the process of clustering and classification. If too many features strongly (anti-) correlate, the separation in hierarchical clustering can lead to unusable clustering hierarchies, as similar structures can be found in several subtrees of the dendrogram. In Hall ([Hal99]), the importance of correlation in terms of feature selection for classification is discussed as a strong (anti-) correlation can lead to overfitting effects, as these features have no additional value to the model. Good features of a set of features correlate with a specific class while uncorrelated with any other class. Fig. 2.1b shows a correlation matrix of several features on the x-

and y-axis, whereas the diagonal stands for the correlation of identical features. A correlation coefficient of 1 means that features are strongly correlated, i.e., the features of y are predicted fully by the features of x, whereas -1 means they are strongly anti-correlated, i.e., the features of y are predicted wholly negatively by the features of x. A correlation coefficient of 0 means that features are not correlated and thus not influenced by other features.



(a) Boxplot with several features.

(b) Correlation matrix.

Figure 2.1: Boxplot and correlation matrix.

2.2.2 Effect Size

Effect sizes are also robust statistical measures to evaluate a set of data objects against another to specify their relationship based on the standard deviation. A ubiquitous effect size measure is *Cohen's d*, which is very valuable to show, e.g., the explanatory power of features between two data sets. Thus, it can be a helpful tool for cluster analysis by finding those salient features that are significant for a specific cluster. Furthermore, *Cohen's d* is also an auspicious method to validate representative sampling strategies.

Cohen's d, especially a *pooled* version where the number of objects is considered, was defined in [Coh88] and further specified in [Saw09] as follows:

Definition 3 (Pooled Cohen's d)

$$d = \frac{\bar{x_1} - \bar{x_2}}{s} \tag{2.3}$$

where $\bar{x_1}$ and $\bar{x_2}$ are the two means of each set, and s is the pooled standard deviation of two sets.

$$s = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \tag{2.4}$$

where the variance for s_i^2 is defined as in Eq. (2.12).

For the interpretation of pooled Cohen's d, the following specifications for the effect size were made. A value of 0.01 is a very small effect, a value of 0.2 delivers a small effect, 0.5 is a medium effect, 0.8 is a large effect, while values of 1.2 are described as very large and 2.0 is as huge effect.

2.2.3 Box & Whisker Plot

Box & Whisker Plots is a further tool for analyzing data sets, briefly called boxplots. Boxplots enable the possibility to have a closer look at the features and their deviation. Thus, it is an essential tool for reviewing data transformation and normalization steps in data preparation, which will be introduced in the following section. These steps can also be discerned in boxplots, potent methods to depict statistical values like mean and median, but also skewness and variance to facilitate data analysis of data sets' features. An example of a boxplot using pythons matplotlib[6] can be seen in Fig. 2.1a. Each boxplot consists of standardized values like the *minimum*, which is the lowest data point in a data set excluding outliers; the *maximum*, which is the highest data point without any outliers; the *median*, which describes the middle value and the *first* and the *third quartile*, which describe the *median* of the *lower* respectively the *upper half* of the data set. The box of the *Box & Whisker* plot is represented by the values between the *first* and *third quartile*, in which the *median* is drawn as a line. The *mean* (green triangle) value is sometimes drawn in the box. The *whiskers* do not have a standardized definition, but in some cases, the *lower* and the *upper whiskers* are represented by the *minimum* and *maximum* values. If many outliers are present in the data set, a more standard definition for the lower and upper whiskers is the *first quartile* minus 1.5 times the *interquartile range*, respectively, the *third quartile* plus 1.5 times the interquartile range. The *interquartile range* is the difference between the *third* and the *first quartile*. All data points that do not lie between the *whiskers* are depicted as *circles* [MTL78].

So many statistical values introduced in this section can be comprehended in a boxplot. A tiny box with short whiskers represents a feature with a low *standard deviation* around the mean. In contrast, larger boxes are interpreted as features with a higher

[6] https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.boxplot.html

standard deviation. If the mean of a feature is quite balanced in the middle of a box, i.e., the whole box and whiskers are symmetric, the *skewness* delivers a *symmetric distribution* of the data. *Right-skewed* distributions of features are depicted as boxplots of features where the mean value is entirely at the bottom of the box, while in *left-skewed* distributions, the mean value drifts to the top of the box. Thus, boxplots deliver many possibilities for processing and interpreting features while preprocessing. Especially the suitability of data sets for following steps, such as clustering, where a consistent distribution of data is crucial in terms of distance measures, shows the benefits of boxplots.

2.3 Distance Measures

Distance metrics deal with comparing pairwise objects, such as single data points and sets. For the evaluation of the novel Borda clustering approach in Chapter 3, as well as several other clustering approaches in Chapter 4 and the model building process in Section 5.8 distance measures are essential. Nevertheless, before specific *Distance Measures* between objects are presented, the definition of *Distance Measures*, which quantifies the distance between pairwise objects, is introduced as stated in [LRU20].

Definition 4 (Distance Measure)
Given a set of points, defined as a space, a distance measure on this space is a function $d(x, y)$ between two points x and y, which produces a real number as output and satisfies the following axioms.

1.) $d(x, y) \geq 0$ which states that there are no negative distances.

2.) $d(x, y) = 0$ if and only if $x = y$. Distances are always positive, except for the distance between a point and itself.

3.) $d(x, y) = d(y, x)$ Distances are always symmetric.

4.) $d(x, y) \leq d(x, z) + d(z, y)$ which describes the triangle inequality. This axiom defines distances between two data points as shortest paths, as travel between two points x and y via a third point z has no benefits.

Distance measures are crucial for clustering algorithms such as k-means and their variants, but distances between sets can also be calculated. Distance and similarity measures describe how close data objects or sets of objects are to each other. Well-known traditional measures like the *Euclidean Distance* or *Canberra Norm* will be used with the basic k-means clustering to evaluate against the Borda clustering approach in Chapter 3 defined in [BS13; ES00; LRU20].

The basic *Euclidean Distance* represents the linear distance between two data points and thus is very common in k-means clustering as the variance between the data points and the centroids are minimized and thus the Euclidean distance maps well according to the target function of k-means, and centroids are easy to determine. In addition, the calculation is stable and produces a low overhead.

Definition 5 (Euclidean Distance)
Given two points $x_i = (x_{i_1}, \ldots, x_{i_d})$ and $x_j = (x_{j_1}, \ldots, x_{j_d})$ with d dimensions, the particular squared distances regarding each dimension are summed up and rooted after that, i.e.,

$$dist(x_i, x_j) = \sqrt{\sum_{l=1}^{d} (x_{i_l} - x_{j_l})^2} \tag{2.5}$$

To also set the focus on distances using small domain ranges, the *Canberra Norm* is a very supplemental measure. As Canberra simulates a kind of normalization, as the local domain ranges of the data points are considered separately, it is a valuable distance measure, especially to evaluate the basic k-means along with the Canberra distance against the Borda Social Choice clustering approach. Similar to the *Euclidean Distance*, the *Canberra Norm* also has low overhead and good stability in terms of calculation.

Definition 6 (Canberra Norm)
It sums up the absolute fractional distances of two d-dimensional points x_i, x_j concerning the range of the focused dimension for all dimensions:

$$dist(x_i, x_j) = \sum_{l=1}^{d} \frac{|x_{i_l} - x_{j_l}|}{(x_{i_l} + x_{j_l})} \tag{2.6}$$

Definition 7 (Weighted Manhattan Metric)
Given two transition tables as matrices $x_{i,j}$ and $y_{i,j}$ both with user roles as sources i and targets j as well as their transition probabilities, the *Weighted Manhattan Metric* is defined as follows:

$$d_{i,j} = |x_{i,j} - y_{i,j}| \times \frac{x_{i,j} + y_{i,j}}{\sum_{i=0}^{n} \sum_{j=0}^{m} x_{i,j} + y_{i,j}} \tag{2.7}$$

Definition 8 (Jaccard Coefficient)
Jaccard is a measure of the similarity of sets. To handle categorical values in data sets in terms of clustering, they must be expressed with numerical values. Given two d-dimensional sets of objects A and B, the similarity of these two sets is calculated as follows, as presented in [WX08].

$$J(A,B) = \frac{A \cap B}{A \cup B} \tag{2.8}$$

where $A \cap B$ represents the number of simultaneous presence in both sets, while $A \cup B$ represents the union set of the two sets. Values of the Jaccard Coefficient lie between $[0, 1]$, whereas a value of 1 signalizes a complete intersection of both sets, and 0 stands for sets without any common objects.

2.4 Data Preprocessing

Before data sets can be processed, such as clustering or classification (cp. Fig. 1.1), it is necessary to observe the characteristics of user features, such as the *distribution*. As most data sets have various features with diverse *bounds*, *variance*, and *outliers*, processing the data to gain a more successful output in later steps of the pipeline, which rely on traditional distance metrics, is essential. *Data Preprocessing* is a significant step of the approach presented in Chapter 4, which will be applied to several uses in Chapter 5 for Twitter data sets and in Chapter 6 for the Telegram data set.

First, data sets must be carefully analyzed using the *Correlation Coefficient* from Section 2.2.1 to select features sufficient for the analyst. After that, the features cannot be used, as they have different *bounds* and *deviations*, which would complicate typical ML approaches in the following steps, such as clustering. Especially regarding social media analysis *distributions* such as the well-known *power-law* matter [New04]. Several well-known data transformation techniques are considered to provide a successfully performed clustering, whereas the most suitable approaches are presented.

Standardization *Skewness* is a vital aspect of preprocessing, which describes a feature's distribution in a data set. In most real-world networks, e.g., the number of followers or friends is a very striking feature for *Skewness*, as many users have a few friends and a handful of users who have thousands of friends, which is described as a *power-law distribution* in [ZAL14; Jav+07]. This distribution is also well-known regarding clicked sites on the internet, as only a few sites are clicked very often, and many sites generate only a few clicks, or e-commerce considering products for sale, where cheap products are exponentially provided more often than expensive ones. Visualizing extremely distributed data sets is easier by utilizing *Skewness* to cope with the distribution. The shape of a distribution of a feature x, which can be *symmetric* or *asymmetric*, so-called *left* or *right-skewed*, is defined by their *Skewness* as in the implementation of Scipy[7], which relies on [ZK00]:

[7] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.skew.html

Definition 9 (Skewness)

$$G_1 = \frac{\sqrt{N(N-1)}}{N-2} \times \frac{m_3}{{m_2}^{3/2}} \tag{2.9}$$

where the adjusted *Fisher-Pearson Standardized Moment Coefficient* for $i = 2, 3$ is given by

Definition 10 (Fisher-Pearson Standardized Moment Coefficient)

$$m_i = \frac{1}{N} \sum_{n=1}^{N} (x[n] - \bar{x})^i \tag{2.10}$$

Kuhn and Johnson deliver an interpretation of the *Skewness* as follows [KJ13]:

- Skewness = 0: Symmetric distribution
- Skewness > 0: Asymmetric distribution, where the mass of the distribution of the feature is concentrated on the left, a so-called right-skewed distribution.
- Skewness < 0: Asymmetric distribution, where the mass of the distribution of the feature is concentrated on the right, a so-called left-skewed distribution.

The *Skewness* of features is a compelling metric to justify data transformations, as extremely *asymmetric distributions* can distort essential values like the *mean* or *median* and thus are no longer representative of the data set. Especially when using distance metrics in clustering, the data sets must have an almost *symmetric distribution*. However, the distribution should be kept intact, as specific characteristics of features can get lost. Thus, a *Standardization* matching the *distribution* must be chosen. Occasionally, choosing a suitable transformation is difficult as distributions can be opaque. To handle this aspect in terms of clustering, a novel clustering approach where preprocessing steps such as normalization and standardization are not needed anymore is introduced in Chapter 3.

After motivating reasons for data transformation using *Skewness*, the most important technique used in this work is presented. As mentioned, extreme outliers can be a massive problem in clustering, as they could negatively influence the clustering process. A compelling method to reduce extremely skewed data is the *Logarithmic Transformation*, which is defined as follows [ZC18]:

Definition 11 (Logarithmic Transformation)

$$x' = \log_{10}(x) \tag{2.11}$$

Features of data sets with an extreme *exponential distribution*, caused by, e.g., outliers, are compressed more powerfully than using other transformation functions like square-root or cube-root transformation, as the distance between outliers and the mean compared to other data points and the mean is getting closer.

Normalization Besides data transformation, it is also essential to consider data normalization because massive data sets with many features often manifest different bounds, which can negatively affect the clustering as they work with distance measures. Considering friend relationships in social networks again, which was described in [ZAL14], the *power-law* distribution of those data sets often causes a mean very close to those data objects which have, e.g., few friends as those users often represent up to 80 percent of all data points. In contrast, the rest is distributed over an enormous range, as stated in the *Pareto Principle*, also known as the 80/20 rule, as discussed in [New04]. Thus, it is also imperative to consider the variance of features because the variance indicates how values in a data set are distributed around the mean. The data points are closer to their mean if the variance has a smaller value. So, it is necessary to ensure that features of a data set have similar bounds and thus have almost equal variance. The *Sample Variance* of a feature x is determined based on a sample set of size N as follows [FPT04]:

Definition 12 (Sample Variance)

$$s^2 = \frac{1}{N} \sum_{i=1}^{N-1} (x_1 - \bar{x})^2 \tag{2.12}$$

Based on the variance, the well-known *Standard Deviation* is calculated as the square root of the variance.

Several methods bring features of a data set into the same bounds discussed in [Osb10], such as the division by greatest value, *MinMax* normalization, or z-score normalization. The first two methods transform the data into bounds between 0 and 1, while division by greatest value does not have mandatory values for 0. Moreover, negative values must be shifted, and extreme outliers significantly influence the transformation, as all values are divided through the highest value. It is advisable to use the MinMax normalization, which leads to an equal range between 0 and 1 for each feature and thus is a *linear transformation* where the ratio of the data is not altered. In contrast to these two techniques, the z-score normalization delivers different bounds for all features, which can lead to problems in clustering since features with an unequal domain can negatively influence the clustering process. Both normalization techniques are sensitive to outliers, so it is mandatory to use a transformation method to reduce

Skewness and, thus, extreme outliers. The *MinMax* normalization for a single feature value x_i and a set of feature values $x_1, ...x_N$ is defined as in [JNR05]:

Definition 13 (MinMax Normalization)

$$x_i^{'} = \frac{x_i - \min\{x_1, ..., x_N\}}{\max\{x_1, ..., x_N\} - \min\{x_1, ..., x_N\}} \tag{2.13}$$

2.5 Machine Learning in Knowledge Discovery

In Section 1.4, the contributions of this thesis were mapped to the most crucial areas of ML, which will be presented in this section in more detail. ML is a manifold area of Artificial Intelligence (AI), in which computers can analyze structures in data sets using algorithms and models and thus can learn from these structures. This learning aspect is a central point of KD because it is also vital to comprehend the output of data sets. Nevertheless, not each model or algorithm can deal without the supervision of human experts, as they convey only specific, typically restricted types of information and knowledge. On the one end of the spectrum, there is USL, which describes how a model can handle data structures without human intervention. The model analyzes patterns in a data set and groups similar data points together using traditional distance measures and statistical metrics. The most important representatives in USL are clustering techniques (cp. Section 2.6) and, consequently, cluster analysis (cp. Section 2.6.2). Furthermore, dimensionality reduction such as Principal Component Analysis (PCA) as well as Linear Discriminant Analysis (LDA) is a considerable technique in USL, where multi-dimensional data sets are processed to present a lucid structure (cp. Section 2.8). Common dimensionality reduction strategies are often used after clustering in cluster analysis to comprehend multi-dimensional clusterings straightforwardly [Mar11; WX08; GMW07].

On the other end of the spectrum, many algorithms and models rely on human supervision, such as classifiers (cp. Section 2.7). In SL, the human expert is pivotal since models cannot learn and address from scratch in labeling real-world data. It is necessary to provide manufactured training data as a ground truth for inputting the SL models. The most established technologies deal with classifying unlabeled unknown data sets, such as Support Vector Machines (SVM), K-Nearest Neighbor (KNN), or decision trees, to name only a few of them. The main goal of SL is to approximate the input data, which should be labeled using training data to the given ground truth. While moderate human expertise is needed at the beginning of SL, human supervision is reduced as more training data is available [Mar11].

Since creating satisfying training data can be tedious, further tasks in ML characterize the procedure. Semi-Supervised Learning (SSL) lays in between SL and USL and

requires a small amount of human-labeled data as input for the classification models. Subsequently, an iterative process starts, in which unlabeled data is classified using the human-labeled data as training data. Until a stable training data set is reached, the training data is enriched with the most valuable results from the classification step. SSL is an approach to quickly reach adequate training data, even though there is no guarantee that all classes are covered coherently considering the quantity. Also, the risk of over-fitting is given if the best-matching labeled data populates only a few classes. The so-called self-training is a process to reduce human resources. However, a fully automatic process is not recommendable since training progress should be examined time by time so that extreme outliers can be removed and inadequately represented classes can be refined specifically [EH19; Zhu08].

A particular case of SSL is called Active Learning (AL), an iterative learning strategy, which can be seen in Fig. 2.2. In contrast to SSL, this approach focuses on a human expert, the so-called oracle. Nevertheless, before the model can query the user with data objects to be labeled, the oracle has to pick a few data objects that must be manually labeled. After that, these labeled data objects enrich the training data, which is used as input for the model to query the following unlabeled data objects. The iterative process continues with the oracle, which has to label these objects manually until all unlabeled data objects are labeled. In this approach, the oracle can precisely control the training progress, especially if inadequately represented classes need to be enriched with further training data. In contrast to SSL, AL needs more human support, especially during the progress of training, while SSL needs human expertise, especially in terms of supervision [Set10].

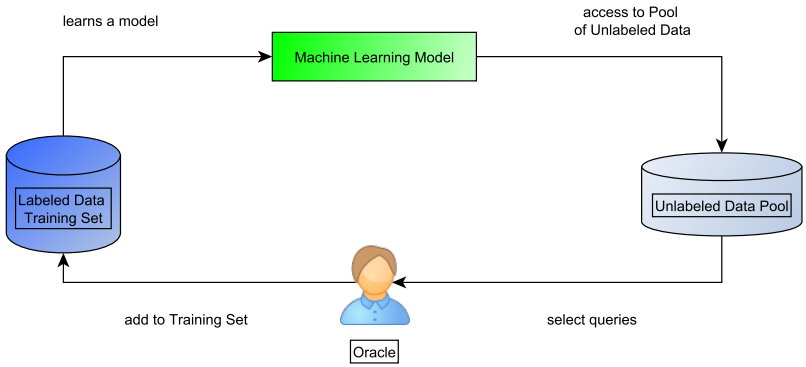


Figure 2.2: Active learning process from [Set10].

After introducing two established areas of ML, including two in-between strategies, Deep Learning (DL) is the following well-known big area in terms of ML. DL provides a range of algorithms and models that can be used for data sets, including USL, SSL, and SL. Neural Networks can classify data, and recent efforts to utilize DL for clustering have yielded encouraging outcomes [Alj+18]. However, achieving the explainability of the output data can be difficult. It is crucial to prioritize explainability when constructing models, particularly during the initial stages where human expertise is indispensable. Parameter tuning and a thorough comprehension of individual algorithms are crucial for effectively utilizing DL.

Reinforcement Learning (RL) is a well-known area of research in ML that aims to maximize the cumulative reward over all actions through an iterative decision-making process. Although unsuitable for model-building, RL's basic approach can be defined as a Markov decision process based on Markov Chains. The Markov Model, based on the same concept, is a promising approach for analyzing data sets. [SB18] and [Mar11] provide insights into this approach, which will be discussed in Section 5.9.

Data Mining (DM) is not a part of ML but has a lot of similarities and intersections with ML. Both areas focus on finding patterns in data sets, while in ML, well-established patterns are discovered and found by algorithms. In contrast to the finding of well-established patterns, DM concentrates more on statistical measures and the finding of completely new structures and patterns in data sets. Especially data preparation and preprocessing are kind of preliminary steps from the area of DM, which are needed before typical ML algorithms can be applied to data sets [WFH11; HPK11]. Finally, several areas of ML and DM are essential in this work, where the best matching and suitable approaches and algorithms must be selected and combined in an adjusted and consecutively way.

After introducing the main areas of ML, the individual steps of the approach, which was introduced in Fig. 1.1 from Section 1.2, are classified as well, to comprehend which steps need to be taken considering the approach presented in Section 1.2. Once the whole pipeline of the approach is well-engineered, it is easier to manage completely new data sets stemming from a new source, which needs to be analyzed.

At the beginning of the process, capturing the data sets and their targeted preparation is essential, considering the preprocessing. This step is fundamental because a reputable analysis is only possible with the knowledge of the features and unique characteristics of the given data sets. The most important steps of preparation and preprocessing include feature extraction, i.e., choosing the most valuable features; data cleaning, i.e., dropping corrupt or incomplete data points; pruning, i.e., excluding specific data points that are not relevant; standardization and normalization to prepare the data set for the following steps by putting features in the same bounds and reduce the skewness of outliers. These steps are quite important areas of DM [WFH11; HPK11].

Moreover, learning the structure of data sets and their features to assign suitable labels manually is also a significant step. This and preprocessing steps are often iterative until satisfying results regarding fine-grained and well-structured user roles are reached. This step of the approach presented in this thesis is dedicated to the area of DM, while clustering, followed after the preprocessing, is an algorithm from the area of USL. Moreover, manual labeling is a part of the strategies from AL and SSL.

Furthermore, after learning the structure iteratively, the benefits of the classification hierarchy and the cluster analysis with a human review is the reduction of human intervention for the following identification in the classification step, which is a part of SL. The ability to adapt the strategy and model on different sizes of samples provides a scalable mean-based probabilistic allocation process, which allows a combination of them to gain more stability of single users and user roles, a representative assertion if only a few samples are adduced and a possibility to reach a good coverage even though data sets are pretty massive. Sampling strategies are well-known statistical approaches in several areas of managing big data, such as political and social polls and surveys [LZ20].

2.6 Unsupervised Learning: Clustering

As mentioned in Section 2.5, clustering is an USL ML technique to partition extensive data sets into clusters of common characteristics. As already introduced in Section 1.2, clustering plays a very significant role in this work, as partitioning similar data has significant benefits in terms of analyzing extensive and vast data sets regarding fine-grained structures. However, it takes work to phrase a formal definition. So [Lan+11] illustrated several phrases, like "*a cluster is a set of entities which are alike, and entities from different clusters are not alike*" or "*a cluster is an aggregate of points in the test space such that the distance between any two points in the cluster is less than the distance between any point in the cluster and any point not in it*". In the history of clustering, several approaches qualify for different shapes of data sets. The basic techniques will now be introduced and considered regarding their benefits and drawbacks for specific data sets. In contrast, approaches used in this work will be introduced later in Section 4.

2.6.1 Clustering Approaches

There are two traditional approaches of clustering techniques: *partitional*, also called *center-based*, and *hierarchical* clustering. While partitional clustering needs a specific predefined number of desired clusters as input to partition the clusters, hierarchical

clustering encapsulates data points with a sequence of nested partitions from singleton clusters until a cluster including all data points is reached. Besides those two traditional clustering approaches, there are many further approaches developed in the last 20 years, such as *model-based*, known as *distribution-based* approaches, *density-based* clustering techniques, and much more [WX08; GMW07].

2.6.1.1 Hierarchical Clustering

Hierarchical clustering can be divided into so-called *agglomerative* clustering or *divisive* clustering techniques. While *agglomerative* clustering describes the bottom-up technique from singleton clusters to a single cluster including all data points, *divisive clustering* is defined as splitting up a single cluster including all data points until each data point remains a single cluster. In both cases, the hierarchy is created by using a proximity matrix. Considering hierarchical agglomerative clustering, which is commonly used, the proximity matrix is created by calculating distances between each data point to each other using several measures. Furthermore, many fusion strategies exist, so-called linkages, which are needed to merge those two clusters in each step with a minimum distance. The algorithms create a clustering hierarchy called a dendrogram, where the clustering progress can be comprehended. At the top of the dendrogram, the cluster comprises all data points, while at the bottom, each cluster is represented by a single data point. Starting at the bottom, one can find out which two clusters are proximal to each other, as well as the distance between the two clusters. While *hierarchical divisive* clustering needs to consider 2^{N-1} possible subsets in each cluster for N data points, *hierarchical agglomerative* clustering has only to compare each data point to each other, which is less computationally intensive. Compared to other clustering techniques, the complexity of $O(n^2)$ is relatively high and an apparent drawback, especially in large data sets, which need a lot of time and resources. Besides classical *hierarchical clustering*, there exist *BIRCH* and *CURE*, which also are a kind of hierarchical clustering techniques that deal better with massive data sets but need a lot more tuning in terms of hyper parameters, which makes it quite complex to find a suitable configuration. Also, many final clusters have to be specified in advance, like in partitional clustering, which needs several passes and more expert knowledge to find the most suitable configuration [WX08; GMW07]. The algorithms, which are essential for this work and used distance measures, will be introduced later in Chapter 4.

Fig. 2.3 shows the dendrogram of a hierarchical agglomerative clustering. As this is a bottom-up approach of a quite huge data set, the dendrogram is pruned at the level of 30 clusters due to lucidity reasons. In the dendrogram, one can trace the progress of the cluster merges until all clusters are merged into one cluster.

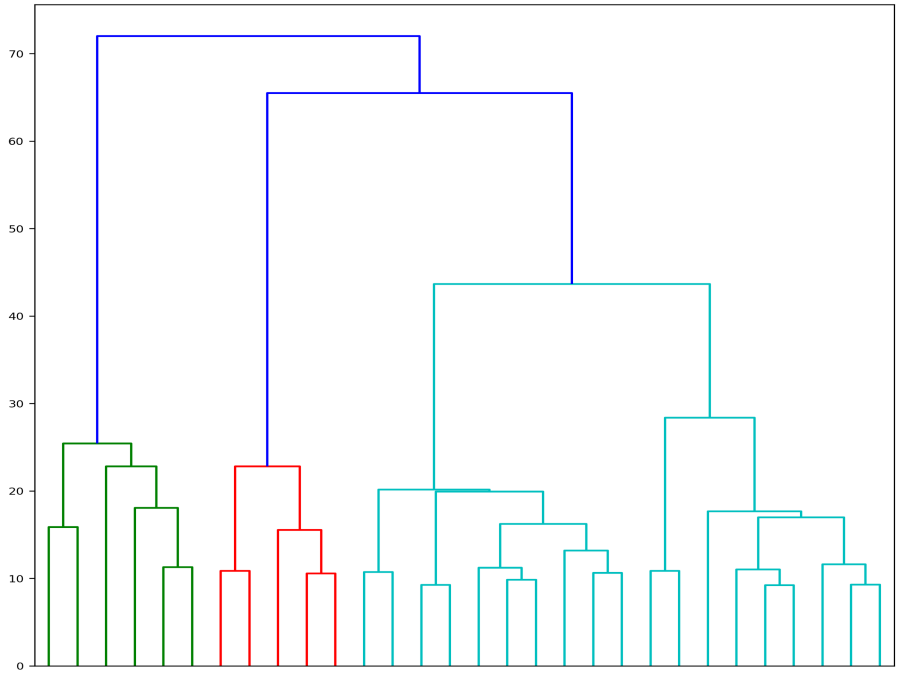


Figure 2.3: Dendrogram of hierarchical agglomerative clustering.

2.6.1.2 Partitional Clustering

After introducing hierarchical clustering techniques, the benefits and drawbacks of *partitional clusterings*, such as k-means clustering, will be discussed. *Partitional clustering* approaches are compared to hierarchical clustering and easier to manage, especially in massive data sets as, e.g., *k-means* has a complexity of $O(n \cdot k \cdot d)$ for n d-dimensional objects, which should be clustered in k clusters. Partitional clustering approaches converge against a (local) minimum by minimizing the variance of the means of each cluster. In *k-means*, each data point represented as a vector is assigned to randomly k predefined centroids. In contrast, some approaches do not concentrate on the means of the clusters but on representative data points from each cluster (*k-medoids*), also known as *Partitioning Around Medoids* (PAM), which has a complexity of $O(n^3 + k(n-k)^2 \cdot i)$ for n data objects, which should be clustered in k clusters within i iterations. Another benefit is that several distance measures can be used depending on the data set's specification. However, it is also a trial and error-influenced way to

find the best specification. Moreover, the fact that the desired number of clusters has to be chosen before the clustering process starts is a drawback, as the clustering has to be started again using another k until a good and satisfying result is found. Another drawback of *partitional clustering* is that clusters are built based on inertia, also called the within-cluster sum-of-squares criterion. If clusters do not have a compressed convex or isotropic structure but also have some outliers or have elongated or other manifold structures or irregular shapes, *partitional clustering* pushes its limits. Moreover, they do not work effectively in high dimensional data sets and are contingent on the initial choice of the k cluster centroids. In this work, *k-means* clustering is the base for an approach where normalization and standardization are prevented from saving a lot of human and machine resources. The basic *k-means* clustering is adapted as no traditional distance measure like the Euclidean distance is used, but a novel treatment to weight each feature dimension in an identical way, which will be introduced later in Chapter 3. Since *k-means* clustering works best for clusters with a structure, the use cases presented later are preference-based skyline data sets, which are more structured than raw data sets as only data objects are considered, a user favors in terms of a recommendation scenario. Other data sets, which are completely unstructured, are difficult to handle using k-means clustering [WX08; GMW07; ES00].

k-means Clustering For the purposes of the novel Borda Social Choice Clustering approach from Chapter 3, the methodology of the basic *k-means* clustering as well as the extension *k-means++* is exploited. Thus, both approaches will be recapped in this section. *k-means* clustering is presented as defined in [Jai10] as follows.

Definition 14 (k-means Clustering)
Given a set X consisting of n d-dimensional objects x and k user-desired clusters c_i, *k-means* works as follows:

1) Find an initial partition for the cluster centroids by choosing a random d-dimensional object of X for each of the k centroids.

2) Calculate for each object the distances to all centroids by using a distance measure, e.g., Euclidean distance, and subsequently allocate each object to the closest centroid.

3) Recalculate each centroid by averaging the contained objects.

4) Proceed with Step 2) until two succeeding clusterings are stable, which means that all clusters from the last iteration contain the equal set of objects as in the current iteration.

Compared to the classic *k-means* clustering, in *k-means++*, the initial partition is not chosen arbitrarily by random, but by a randomized *seeding technique*. For finding a more accurate clustering, the best centroids for the initial partition should be found. Step 1) of the k-means algorithm is replaced as follows, cp. [AV07]:

Definition 15 (k-means++ Clustering)

1a) Arbitrary choose an object of the set X as first cluster centroid c_0.

1b) For each further cluster centroid $c_i \mid i \in \{1, ..., k-1\}$ choose $x \in X$ with a probability of

$$p(x) = \frac{dist(x, c_i)^2}{\sum_{x \in X} dist(x, c_i)^2} \tag{2.14}$$

where $dist(x, c_i)^2$ is the shortest squared Euclidean distance from a point x to the already chosen closest centroid c_i.

1c) Proceed with Step 2) of the k-means clustering algorithm.

2.6.1.3 Distribution-based Clustering

Distribution-based clustering focuses on probability models, where a probability to each cluster is assigned for each data point. In contrast to centroid-based approaches, the data points to cluster are not described by a centroid or a representative but by a *probability distribution*, in most cases, the *Gaussian distribution*. For each cluster k, this distribution is described by the centroids and a covariance matrix for all data points in the cluster. A well-known approach is the *Expectation-Maximization* (EM) algorithm by Dempster et al. [DLR77], which consists of 2 steps being iterated. The first is the *Expectation* step, where probabilities of the feature vector of each data object to each cluster are estimated for an initial *Gaussian distribution* for k clusters. The cluster model is recalculated in the *Maximization* step by maximizing the parameters given as a probability distribution. This distribution leads to an optimized model until the model no longer improves. As *EM* delivers probabilities for each data point to each cluster, clusters can be built by data points with the highest probability. It is also possible that data points lie between 2 or more clusters; thus, there is no absolute majority for one cluster. Those data points can be considered outliers. A drawback of the *EM* approach is that, similar to centroid-based models, the number of desired distributions, i.e., the number of desired clusters, has to be chosen before the clustering starts. Several approaches, such as k-means, k-means++, or completely randomized chosen centroids, can be considered. The complexity of $O(n \cdot m \cdot i)$ for n objects, which should be clustered into n set of distributions in i iterations, is comparable to k-means clustering. Moreover, EM converges to a (local) minimum, too, and depends on the

initial allocation of the k clusters. Moreover, *EM* does not create clusters of similar size but very compact clusters depending on the probabilities of the data points. Thus, *EM* has similar problems to centroid-based approaches but can handle noise [WX08; GMW07; ES00]. *Distribution-based* clusterings were initially used for the approach, presented in Fig. 1.1 as it improves the primary k-means clustering and can handle noise, which is prevalent in massive data sets stemming from social media.

2.6.1.4 Density-based Clustering

Density-based clustering approaches can capture dense areas in the data set and distinguish them from areas with a lower *density*, i.e., data points outside clusters, such as *noise* data points. Crucial for this approach is a *threshold* differing between a *dense area* within a cluster for each data point and a noise area or area with no data points. The algorithm groups data points with a lower density of adjacent data points than a predefined threshold. The key to this approach is so-called *core points*, those with at least a predefined number of other points in their neighborhood. This *density-reachability* describes the data points directly reachable from another data point within a predefined distance, and the *density-connectivity* describes transitive connectivity between data points. Data points that do not fulfill the *core points* or *density reachable* aspects remain as *noise*, which is not clustered. Thus, *density-based* clustering approaches can cluster data sets with several structures in a d-dimensional space. One of the most established density-based approaches is *Density-Based Spatial Clustering of Applications with Noise (DBSCAN)*, introduced by [Est+96]. Compared to centroid-based and distribution-based approaches, *DBSCAN* can determine a suitable number of clusters without input for the number of clusters and has a good complexity of $O(n \log n)$. Furthermore, the tuning of the *hyper parameters* considering the number of minimum points in a neighborhood and the predefined maximum distance to other points are very significant and time-consuming, as the cluster can quickly get too big when setting both parameters to high or only less and relatively small clusters emerge when setting both values too low. A further density-based clustering approach is *Ordering Points to Identify the Clustering Structure (OPTICS)* [Ank+99], distinguishing from *DBSCAN* by differing noise points and the periphery. An optimization compared to *DBSCAN* is the possibility to define the radius of the neighborhood more precisely by an upper bound. Moreover, candidate points are ordered due to their core distance, enabling a kind of clustering hierarchy for each cluster. *Density-based* clustering techniques have drawbacks in data sets with a hierarchical structure within a cluster, i.e., the inner core of a cluster has a higher density than the outer core or if there are various structures with differing densities within a data set. This aspect often leads to the problem that smaller clusters cannot be clustered, which remains as noise [WX08; GMW07; ES00; BS13]. The possibility of

assigning a data point not wholly to a single cluster contradicts the classical definition of clustering. Nevertheless, a *density-based* clustering approach is auspicious for this work, where vast sets of multidimensional data sets are analyzed, and thus, overlapping clusters are inevitable. Especially in terms of explainability, OPTICS may be another worthwhile possibility, as the clustering hierarchy is given.

2.6.2 Cluster Analysis

After introducing the most common clustering techniques, it is also essential to discuss the results of clusterings. In this thesis, the granularity of structures plays an essential role. Thus, several traditional *cluster evaluation metrics* will be presented and evaluated on several use cases due to their suitability for detecting fine-grained structures. While in centroid-based clustering, the fixed number of desired clusters has to be chosen beforehand, in hierarchical clustering, analysts must decide where to cut off the clustering in the dendrogram to examine the specific clusters closely. Thus, *cluster analysis* is a rather important area, as it guides the analyst, and time can be saved. In clustering algorithms, the question of *quality* is a well-known problem, as there is no ground truth knowledge. Cluster analysis metrics give a clue on clusterings, which can be sufficient, as they primarily are based on statistical calculations to describe the separation of clusters from others or the density of clusters. Nevertheless, no perfect metric describes a specific clustering perfectly, as clustering depends on the requirements of analysts and given scenarios. Thus, it is difficult to compare the results of different approaches and to determine the quality of clustering methods [KP17]. To handle and analyze these clustering techniques, several well-known *measurements* are presented. While all these metrics are statistic-based and work with traditional distance functions, such as the Euclidean distance, the analysis in use cases such as the Borda Social Choice Voting Rule in Chapter 3 but also using no distance-based metrics in hierarchical clustering, e.g., Ward's linkage, the traditional distance metrics could reach their limits in terms of validity.

2.6.2.1 Silhouette Coefficient

The *Silhouette coefficient* is a fundamental and well-known internal cluster analysis metric, which is described by comparing the *tightness* and *separation* of clusters all over the whole clustering result by considering a single plot. So especially for partitioning clustering techniques that do not rely on a hierarchy, finding a starting point for cluster analysis is challenging. The *Silhouette coefficient* gives a good overview of clustering, as it can show if data points lie within a cluster or in between other clusters.

The *Silhouette coefficient* for a *sample* or the whole data set is defined as in [Rou87]:

Definition 16 (Silhouette Coefficient)

$$s(i) = \frac{b(i) - a(i)}{\max(ai, b(i)} \tag{2.15}$$

where $a(i)$ is the mean distance between a sample and all other points in the same cluster A and $b(i)$ is the mean distance between a sample and all other points in the next nearest cluster B.

The scores of $s(i)$ are lying in between $[-1, 1]$, as a value close to 1 describes the case that the within dissimilarity of $a(i)$ is much smaller than the smallest within dissimilarity $b(i)$, i.e., the clusters are well clustered and solid and are well separated from the next nearest cluster. If $s(i)$ has a value close to 0, $a(i)$ and $b(i)$ are almost equal, i.e., i lies in between the two clusters A and B. If $s(i)$ is close to -1, $a(i)$ has a much larger value than $b(i)$, which means that object i is much closer to cluster B than cluster A and thus object i got misclassified.

To sum up the *Silhouette coefficient*, it is a very straightforward technique to describe structures of a cluster, as all kinds of clusterings, from well-separated dense clusterings with of score of 1 over overlapping clusters, which have a score of 0 and utterly incorrect cluster, which have a score of -1 are described [ES00]. To specify the values [KR90] defined values for the *Silhouette coefficient* as follows. A silhouette value greater than 0.7 is defined as clusters with a strong structure, between 0.5 and 0.7 as clusters with an adequate structure, and between 0.25 and 0.5 as clusters with a weak structure. As previously defined, clusters that deliver a value lower than 0.25 remain as clusters with no structures. The *Silhouette* is a valuable metric if data sets with well-separated clusters are considered, but quite challenging to interpret if data sets with many diverse and manifold features have to be analyzed. Furthermore, the calculation of the whole data set is quite complex, as many computations are needed. Thus, the Silhouette coefficient is often calculated on a sample.

2.6.2.2 Davies-Bouldin Index

As it is only possible to analyze clusters adequately by considering one single cluster analysis metric, other metrics with different specifications deliver more manifoldness. The *Davies-Bouldin Index* is compared to the Silhouette coefficient; the advantage is that the computation is not that complex and is based on quantities and features.

The *Davies-Bouldin Index* is defined as the *average similarity* of each *cluster* and its *closest cluster* [DB79]:

Definition 17 (Davies-Bouldin Index)

$$R_{ij} = \frac{S_i + S_j}{M_{ij}} \tag{2.16}$$

where S_i and S_j are the cluster diameter, i.e., the average distance between each point and the cluster centroid of cluster i and j and M_{ij} is the distance between the centroids of the clusters i and j.

Thus, the Davies-Bouldin index DB for N clusters is defined as:

$$DB = \frac{1}{N} \sum_{i=1}^{N} \max R_{ij} \mid i \neq j \tag{2.17}$$

The Davies-Bouldin Index delivers values in $[0, \infty)$ whereas smaller values close to zero are interpreted as well separated and compact clusters.

To summarize the *Davies-Bouldin Index*, it is easier to interpret the clustering than using Silhouette. A drawback of Davies-Bouldin is that the value does nothing about overlapping clusters or completely unusable clusters, which have a Silhouette value around or below 0.

2.6.2.3 Calinski-Harabasz Index

Another well-known internal cluster analysis metric is the *Calinski-Harabasz Index*, the *Variance Ratio Criterion*. Compared to the other two metrics presented in this subsection, the focus is on the ratio of the sum of between-cluster dispersion and within-cluster dispersion considering all clusters. The *Calinski-Harabasz Index* s for a data set E of Size N, which should be clustered into k clusters is defined as in [CH74]:

Definition 18 (Calinski-Harabasz Index)

$$s = \frac{tr(B_k)}{tr(W_K)} \times \frac{n_E - k}{k - 1} \tag{2.18}$$

where $tr(B_K)$ is the trace of the between group dispersion matrix and $tr(W_k)$ the trace of the within cluster dispersion matrix.

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T \tag{2.19}$$

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T \tag{2.20}$$

where C_q is the set of points in cluster q, c_q the centroid of cluster q, c_E the centroid of data set E and n_q the number of points in cluster q.

Calinski-Harabasz delivers values in $[0, \infty)$, whereas higher values deliver well-separated and compact clusters.

2.7 Supervised Learning: Classification

Classification is a very significant area in SL. While clustering delivers groups of data points without knowledge of specific classes, classification's primary goal is to assign each of the given input data objects to a class. There are two critical steps to pave the way for successful classification. First, each classifier needs a *knowledge base*, so-called *training data*, to train the classifiers. As a first step, the manually labeled training data functions as a ground truth to improve the prediction of the classifiers until they work stable using the *ground truth* data both as input for the training data and the data to classify. If stable training data is given, it is used to classify completely new data sets in the second step, as they are assigned by their attributes to a class using the trained classifier [ES00]. As mentioned in the contributions in Section 1.3 and the allocation to areas of ML in Section 1.4, detecting fine-grained user roles plays a significant role in this work. The basis for identifying user roles is clustering, where users with similar features are structured. These clusters are now used as input for the classification step to label each of them probabilistically using a representative of the clusters, such as the mean, median, or other metrics, which will be discussed later in Section 4.6. This *probabilistic labeling* is part of the foundation for the novel *Multi-Sampling and Combination Strategy* for analyzing and detecting fine-grained structures in several use cases, such as detecting user roles in Twitter and Telegram data sets.

2.7.1 Popular Classification Approaches

In this section, the most common and well-known types of classifiers are presented and discussed in terms of their usefulness for the approach presented in this thesis. There are many types of classifiers, which all have advantages and drawbacks in different use cases. One widespread approach is *Bayes-Classifiers*, which rely on classifying conditional probabilities. For this approach, hypotheses need to be defined, whose probabilities are optimized together with a priori probabilities and the given data to reach a successful stable classification. This approach is used chiefly for classifying

text and text snippets and needs a reliable estimation of the a-priori probabilities. For this work, Bayes-Classifiers are not considered, as they are predestinated for text analysis. In contrast, this approach concentrates on classification using a manifold set of numerical features describing a user role by a vector [ES00]. Besides Bayes-Classifiers, there are many further classifying techniques, such as *Nearest Neighbor Classifiers*, *Support-Vector-Machines*, and *Decision Trees*, which will be introduced briefly in this section, considering their suitability to this approach. The specification considering hyper parameters will be discussed later in Section 4. As classifiers must also be carefully evaluated, several important quality metrics will be introduced.

2.7.1.1 Nearest Neighbor Classifier

Compared to kernel approaches, a straightforward classification model considering the *Nearest Neighbor Classifier* is KNN. Local *density estimation* by fixing the parameter of *k-nearest neighbors* is used by finding an appropriate value for the volume of a region V. To estimate the *density* around a *data point*, the *radius* of this *sphere* around the data point continuously grows until k data points are within this *sphere*.

Definition 19 (k-Nearest-Neighbor Density Estimation)
Bayes' Theorem is used to calculate the *posterior probability* of class membership. For each class C_k and a data point x, the posterior probability is defined by

$$p(x \mid C_k) = \frac{p(x \mid C_k)p(C_k)}{p(x)} = \frac{K_k}{K} \tag{2.21}$$

where the *probability* for a data point x to a class C_k is given by

$$P(x \mid C_k) = \frac{K_k}{N_k V} \tag{2.22}$$

the *unconditional density* is given by

$$p(x) = \frac{K}{NV} \tag{2.23}$$

and the *class priors* are given by

$$p(C_k) = \frac{N_k}{N} \tag{2.24}$$

To minimize the *probability* of misclassifying data points, a data point x is assigned to a class having the highest possible *posterior probability* by dividing the number of points K_k from a class C_k within the *sphere* V through all the *number of points* K in this *sphere*.

Thus, K is an essential parameter for the *smoothness*. The smaller the value of K, the *noisier* the *density model* becomes, as higher values lead to a *smoother distribution* and fewer *distinct boundaries*. A significant advantage of KNN is that only one parameter is needed for calculating the posterior probability and, thus, is a straightforward model. KNN is limited to data sets that have a low number of dimensions. Thus, the choice of features is crucial for the success of this model. Another drawback is that the whole training data set must be stored, which can be expensive for extensive training data. Using additional computation by constructing tree-based search structures leads to a more efficient search of nearest neighbors, especially in extensive training data [Bis06].

2.7.1.2 Support Vector Machines

SVM are *kernel-based models*, which rely only on *sparse solutions*, i.e., predictions of new input data depend only on a kernel function evaluated on a subset of training data. Since the determination of parameters of SVM focuses on an *optimization* problem, a benefit of this approach is that each local solution also delivers a global *optimum*. They work best in use cases considering a 2-class problem using linear models, as a linear separation between two classes is possible in a finite number of steps, finding the solution with the smallest generalization error. The *margin* is the smallest distance between the decision boundary and the samples. The so-called *support vectors* are maximized. If class distributions are overlapping, a precise separation is not possible. Thus, the concept of SVM can misclassify some training points, i.e., data points can be on the wrong side of the margin, by penalizing them using negative values depending on the distance to the *margin*. Thus, it is easily possible to manage overlapping classes, as exceedingly manifold data sets with many features often result in overlapping classes if training data is not fully well-engineered. As SVM works only in a two-class problem case, two solutions for a *k-class problem* are possible. *One-Versus-The-Rest* is an approach in which, for each class, two sets of training data are needed. One set is the training data for this class as a positive class; the other set consists of the training data of all other remaining classes, which is the negative class. Thus, a SVM must be trained for each class. A drawback of this solution is if unbalanced training data is given, i.e., classes with quite a small amount of data points compared to the others. The other solution is called *One-Versus-One*, where for each of the k classes $\frac{k(k-1)}{2}$ classifiers have to be created in terms of a *Multi-Class-SVM*. This approach leads to higher complexity in terms of time for creating and executing the classifiers on new data sets. A solution to reduce computation time is to organize the classifiers in a directed acyclic graph, where only k-1 pairwise classifiers are needed for the execution of new data points [Bis06].

2.7.1.3 Decision Trees

As nearest neighbor classifiers do not deliver knowledge considering specific classes, *Decision Tree*-based classifiers do so, as they use decision trees, which can be traced easily in terms of knowledge discovery. A decision tree is a tree with attributes represented by the tree's inner nodes, by classes represented by the leaves of a tree, and edges, which represent a test on the parent nodes attribute. A binary tree is used in most approaches based on *Decision Trees*, but some implementations consider more than two edges at a parent node. Each *decision tree* is built using training data to classify new data by paving the way from the root to a leaf given by the decisions at each node considering the attributes or features. Thus, each new data object to classify ends up in a specific leaf node, which represents a class [ES00].

Gradient Boosted Decision Trees (GBM) The idea of Gradient Boosted Decision Trees (GBM) combines several decision trees into an ensemble. A *decision tree* is a sequential model where attributes or values are compared hierarchically against other attributes or threshold values. Starting from the root, one path ends in a leaf, leading to a specific class. Using *boosting* has the advantage that single *decision trees*, which do not consistently deliver stable and significant results, are combined into more powerful and significant models. The boosting technique creates several base models sequentially and improves each model based on the model created before optimizing a cost function with the most negative gradient direction. For this work, the implementation of *XGBoost*[8] is used, as this approach enables a sufficient classification even for classes with a low number of training data. Moreover, *XGBoost* copes with overfitting problems even in small training data sets. Compared to single decision trees, it is hardly possible to trace the classification process, as many decision trees are combined [Bis06; Kot13; Fri01; CG16].

Extremely Randomized Trees (ET) Compared to GBM, not a cost function is used to combine decision trees to an ensemble, but randomized decision trees. While samples from the training data set create ensembles in random forests, the Extremely Randomized Trees (ET) approach delimits from *random forests* by splitting the nodes of the *decision tree*. *Random forests* use thoroughly randomly chosen features, a subset, for the splits of the decision tree, as ET specify this process by considering the whole learning sample, as well as choosing randomly generated thresholds for each considered feature. In contrast, the best are chosen for the splitting in the decision tree. Furthermore, for the prediction of the ensemble, an aggregation over all decision trees using a majority vote is used to gain a more stable model. Also, exploiting the whole

[8] https://www.xgboost.ai

training samples delivers a more stable classification than essential random forests, as the variance is reduced, but the bias is slightly increased. Even though ET tends to overfit, especially for classes with a small training data set, the approach classifies generalized classes as more reliable as the variance is lower than in other techniques. Both ET and GBM have their advantages in reducing variance and providing a better generalization for smaller classes as building ensembles rely on a lot of combined decision trees and thus are more stable compared to other classification approaches [Bis06; GEW06; Kot13; Bre01].

2.7.2 Quality Evaluation

This work deals, among other things, with building *training data* for detecting fine-grained structures. To evaluate and validate *classification models* as well as different configurations considering *hyper parameters* built upon data sets, including fine-grained structures, several well-known evaluation metrics are crucial for the performance of successfully classifying user roles. A pool of manually labeled *training data* for each class is needed for the evaluation, which serves as a ground truth. Moreover, each classifier delivers *probabilities* for each cluster to each class, whereas the class with the highest probability is adduced for the quality evaluation. Table 2.1 shows the structure of the confusion matrix where the actual labels are on the y-axis and the predicted labels are on the x-axis as defined in [SL09; Tha18]. The values on the *diagonal* are the number of *correctly classified* data points, e.g., $tp_{i,j}$ are the number of data points of class i which are *correctly classified as class* j, the so-called *true positives*, whereas $err_{k,l} \mid k \neq l$ is the number of data points of class k which got classified as class l, which are *relevant* for determining *misclassified classes*. *False negatives* fn_l is the sum of all data points that got misclassified in a class l $fn_l = \sum_{k=0}^{j} err_{k,l} \mid k \neq l$, whereas *false positives* fp_k describe all data points with an actual label k, which got classified in another class l $fp_k = \sum_{l=0}^{i} err_{k,l} \mid k \neq l$.

		Predicted Label					
		0	1	2	3	...	j
True Label	0	$tp_{0,0}$	$err_{0,1}$	$err_{0,2}$	$err_{0,3}$	...	$err_{0,j}$
	1	$err_{1,0}$	$tp_{1,1}$	$err_{1,2}$	$err_{1,3}$	...	$err_{1,j}$
	2	$err_{2,0}$	$err_{2,1}$	$tp_{2,2}$	$err_{2,3}$	...	$err_{2,j}$
	3	$err_{3,0}$	$err_{3,1}$	$err_{3,2}$	$tp_{3,2}$	...	$err_{3,j}$
	...	...	...	...	...	...	...
	i	$err_{i,0}$	$err_{i,1}$	$err_{i,2}$	$err_{i,3}$	...	$tp_{i,j}$

Table 2.1: Confusion matrix for quality evaluation.

Furthermore, in [SL09; Tha18], important evaluation metrics are defined for the set of given classes C, where *Precision* P_i of a class i describes the amount of *correctly classified* objects of class i dependent on *all classified objects classes*, which have the actual label i and *Recall* R_i is defined as the amount of *correctly classified* objects of class i dependent on *all objects which got classified* in class i. *F1* is a measure that combines both *Precision* and *Recall* in order to give a *comprehensive evaluation metric for classification*. The amount of considered data objects is irrelevant for these evaluation metrics and can influence the results positively and negatively when considering only a small amount of data sets.

Definition 20 (Precision)

$$P_i = \frac{tp_i}{tp_i + fp_i}, i \in C \tag{2.25}$$

Definition 21 (Recall)

$$R_i = \frac{tp_i}{tp_i + fn_i}, i \in C \tag{2.26}$$

Definition 22 (F1)

$$F1_i = \frac{2 \cdot P_i * R_i}{P_i + R_i}, i \in C \tag{2.27}$$

The following *macro evaluation metrics* are appropriate to provide a general value for *all metrics across all considered classes*, with *equal weighting* for each class. These metrics require good *support* in each class, as classes with a small amount of support would be soaked up by those with higher support.

Definition 23 (Precision Macro)

$$P_{macro} = \frac{1}{|C|} \cdot \sum_{i \in C} (\frac{tp_i}{tp_i + fp_i}) = \frac{1}{|C|} \cdot \sum_{i \in C} P_i \tag{2.28}$$

Definition 24 (Recall Macro/ Balanced Accuracy)

$$R_{macro} = \frac{1}{|C|} \cdot \sum_{i \in C} (\frac{tp_i}{tp_i + fn_i}) = \frac{1}{|C|} \cdot \sum_{i \in C} R_i \tag{2.29}$$

Definition 25 (F1 Macro)

$$F1_{macro} = \frac{1}{|C|} \cdot \sum_{i \in C} (\frac{2 \cdot P_i * R_i}{P_i + R_i}) = \frac{1}{|C|} \cdot \sum_{i \in C} F1_i \tag{2.30}$$

Moreover, *weighted evaluation metrics* can be beneficial, especially if some classes have *lower support*, which may be given in the use cases of this work, as there are always user roles with lower quantities in their training data sets. The following evaluation metrics aid in getting more stable results in creating training data for the classification.

Definition 26 (Precision Weighted)

$$P_{weighted} = \frac{1}{\sum_{j \in C} |C_j|} \cdot \sum_{i \in C} |C_i| \cdot \frac{tp_i}{tp_i + fp_i} = \frac{1}{\sum_{j \in C} |C_j|} \cdot \sum_{i \in C} |C_i| \cdot P_i \tag{2.31}$$

Definition 27 (Recall Weighted/ Accuracy)

$$R_{weighted} = \frac{1}{\sum_{j \in C} |C_j|} \cdot \sum_{i \in C} |C_i| \cdot \frac{tp_i}{tp_i + fn_i} = \frac{1}{\sum_{j \in C} |C_j|} \cdot \sum_{i \in C} |C_i| \cdot R_i \tag{2.32}$$

Definition 28 (F1 Weighted)

$$F1_{weighted} = \frac{1}{\sum_{j \in C} |C_j|} \cdot \sum_{i \in C} |C_i| \cdot \frac{2 \cdot P_i * R_i}{P_i + R_i} = \frac{1}{\sum_{j \in C} |C_j|} \cdot \sum_{i \in C} |C_i| \cdot F1_i \tag{2.33}$$

2.8 Dimensionality Reduction

To simplify the creation of representative training data for multidimensional data sets, *Dimensionality Reduction* is a valuable strategy as it enables easy data visualization in a reduced space. Two strategies discussed in the work of [Qah+15; Man08; HMT10; SKT14; BNJ03; HBB10] will be introduced in this section, as they both provide helpful aspects for this work.

Starting with PCA, the given data points are projected into a *low-dimensional space* by *maximizing their variance*. The strategy behind the PCA is *incremental*, where the new dimensions in the space, called principal components, are created by a *linear combination* of the *original dimensions* and are *orthogonal* to all of the other components. As the transformation progresses, each *principal component* explains less *variance* than the previous one due to the *orthogonality* of the components. The explained variance factor and the influence of the original dimensions are linked to each principal component's *linear combination*. In most cases, only a few principal components are needed to explain the variance of the whole data set.

Another method for projecting the original data distribution into a linear subspace is LDA. Unlike PCA, LDA does not involve *exploration* but requires prior *classification* of data points. The main objective of this projection is to improve the *differentiation* between various data points based on their class centroids while reducing the *scatter*

within each class. One must be aware that while utilizing LDA for mapping a dataset with k classes in a d-dimensional vector space, it is essential to note that the maximum number of resulting dimensions is $(d-1)$. The organization of dimensions is based on their *relevance* in distinguishing categories and is marked with descriptive labels for *clarity* and *consistency*.

Consequently, PCA and LDA are distinct techniques for transforming data into a new vector space, each with its own methodology. The effectiveness of these methods is context-dependent, and thus, a case-by-case determination is necessary to identify the most suitable approach for achieving the desired results. One major drawback of LDA compared to PCA is that features cannot be reproduced, as the information on feature quotas in the reduced components is missing. PCA is a valuable tool for cluster analysis in *multidimensional spaces*, as with only a low effort, clusters can be visualized and proved if the composition is satisfying. Especially for building *training data*, where each class representative is chosen, dimensionality reduction is beneficial, as the *significance* is better than using internal cluster evaluation metrics, as the latter strategy has problems when the number of data points is relatively small. Moreover, a *visual depiction* helps, especially in the building process of training data sets, as outliers can easily be spotted and removed.

Another kind of dimension reduction is introduced in an entirely different use case in Chapter 7, where whole graphs (nodes, edges) representing information cascades are reduced, with *Graph Embeddings* being a valuable strategy to cluster them. In contrast to the *dimensionality reduction* techniques presented in this Section, those techniques are only used in building training data, while *Graph Embeddings* and *Summarization* are preprocessing steps that clustering is even possible. *Dimensionality reduction* can also be used as a preprocessing step before clustering in a KD based approach. However, the effort would only be limited, and the *explainability* of features would be lost. Thus, PCA and LDA will only remain for proving training data [Pau+11]. For the purposes of this work, the implementation of scikit-learn will be used.

Part II

Main Part

Chapter 3

Normalization Avoidance by Exploiting the Borda Voting Rule for Clustering

Don't stop believin'
Hold on to that feelin'
Streetlights, people

Journey - *Don't Stop Believin'*

This chapter explains how the Borda Social-Choice voting rule can be exploited as a distance function for clustering *Pareto-optimal* data sets without requiring a time-consuming normalization and standardization step. The chapter covers two approaches' methodology, synthetic experiments, and quality evaluation. Most of the work presented in this chapter was previously published by the author in peer-reviewed papers [KEK17; KE19] a technical report [KE17], a book-chapter [EKR18] and a demo-poster [KRE19].

3.1 Motivation & Contributions

The approaches presented in this chapter follow only a few steps of a typical Knowledge Discovery (KD) process, which can be seen in Fig. 1.1 of Section 1.2. However, before the approaches are introduced, a general motivation considering *Pareto-frontiers* and the suitability of clustering such data will be discussed, and the research questions will be introduced.

3.1.1 Motivation

Pareto-optimal data is striking in several use cases dealing with recommendations or advertisements, as data sets are tailored to users' requirements. Several steps of a KD approach are needed to process data sets to those user's requirements. Comparing the approaches presented in this chapter to a traditional KD approach, most *preprocessing* steps are no longer needed as much human effort is reduced to save time. Of course, corrupt data has to be eliminated, but there is no need for any normalization and standardization of the data. Since most use cases follow *Pareto-optimal* data sets, sampling is undoubtedly unnecessary, as *Pareto-optimal* data sets are often narrowed down to a fraction of the original data set. The clustering and cluster analysis are the most essential KD related steps. Both approaches presented in this chapter appropriate specific characteristics of *Pareto-optimal* data sets and use them together in a k-means-based clustering approach. Considering cluster analysis, the approaches follow traditional internal cluster analysis techniques. Indeed, a kind of classification could be helpful for the approaches but was not considered. Pareto-dominant data sets often only contain a few data objects, and the number of valuable clusters delivered by both approaches is often relatively small. The overhead of creating training data and a ground truth exceeds the benefit of classifying only a few data objects.

As introduced in Section 1.3 and 1.4, automation and reducing human involvement plays a very significant role in Machine Learning (ML), as it brings benefits both in research as well as in application-specific use cases in e-commerce such as recommendations as well as micro-targeting. Popular streaming services like Netflix, Prime Video, Disney+, or music-streaming services like Spotify or Deezer are acquainted instances that must cope with continuously growing data. So it is challenging to keep track of the results presented to the users, as unintended results overwhelm the users [RRS15].

A ubiquitous approach to minimize unwanted results is to exploit preferences of the users to affect only some best matching suggestions using Pareto-frontiers, also called *Skylines*, which are defined in [BKS01] as "*those points which are not dominated by any other point. A point dominates another point if it is as good or better in all dimensions and better in at least one dimension*". However, the more features are considered, the

more the result set grows, as the set of *Pareto-optima* is built upon each data object that is not dominated by another object considering each feature. To keep track of the best matching results, clustering those objects afterward is a promising approach to grouping similar objects, which are delimited from other objects, to gain a better overview of *Pareto-optimal* objects. So it is possible to present only a few of those best matching representatives to the user instead of the whole clusters [Jai10; SM14]. Consider a recommender system addressing people with similar tastes in movies and series in the following use case.

Example 1
Assume user Bob wants to watch a movie. He prefers movies with a possible low *running time* and a recent *release year* at the same time. The movies presented to Bob are so-called *Pareto-optima* on a *Pareto-frontier* or *Skyline*, which can be seen in Fig. 3.1 as the blue points. In contrast, the skyline points dominate the red points, e.g., P_4 has a more recent release year than P_{14} and a lower running time simultaneously. So P_4 dominates P_{14} in both dimensions at the same time w.r.t. Bobs preferences, while P_3, e.g., is dominated by P_4 only w.r.t. the running time. As mentioned, clustering can be a promising approach to compress and express this set with a smaller, appropriate set of representatives. Another approach is to mask out noise points such as users like P_8 with a more recent release year but a very high running time.

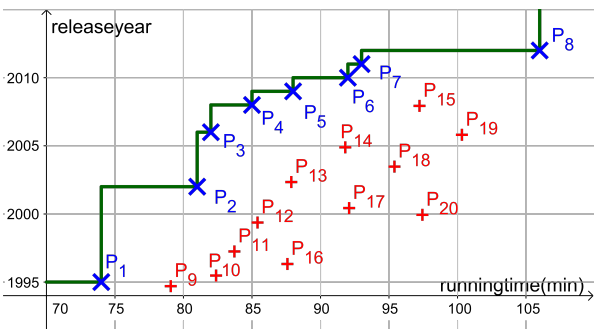


Figure 3.1: *Pareto-frontier* of users with preferably lowest running time and preferably recent release-year.

Addressing this use case, clustering like k-means is a very appropriate approach to automatically group similar objects and separate them from other groups with similar objects to gain a better overview of the data. As there are two dimensions with different domain ranges, achieving a beneficial outcome without great efforts, such as normalization and standardization from the area of preprocessing techniques, is

impossible because the *dimensions* should be set into the *same bounds* so that they are comparable using traditional distance measures. Considering the Example 1, setting this range of minutes for the running time is circumstantial concerning only a smaller range of release years. Thus, using the basic k-means clustering algorithm along with the well-known Euclidean distance is insufficient, as user features and their range can differ from use case to use case. To substantiate the aspect of clustering on a *Pareto-frontier*, the *Best-Matching-Only* objects from Fig. 3.1 could be clustered as can be seen in Fig. 3.2.

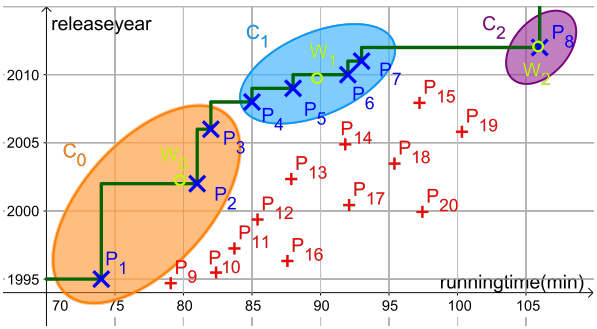


Figure 3.2: *Pareto-frontier* of users with a possible clustering.

Since preprocessing steps like normalization and standardization need much human effort, as experts have to check each feature carefully, the first approach presented in this chapter exploits the *dominance criterion* of *Pareto-frontiers* for cluster allocation and thus does not need preprocessing as each feature dimension is ranked separately. When focusing not only on two dimensions by selecting two features but, e.g., four or even more features, the result set is also growing, as the following example shows.

Example 2
Bob wants to watch a movie. He favors timeless *old-school movies of the late 70s, 80s, and early 90s*, prefers *action-, adventure-movies and dramas*. Since it is later in the evening, he only wants to watch movies with a runtime *between 90 and 130 minutes*. Furthermore, Bob prefers ambitious movies, so the *user rating should be higher than 7* on a score from 0 to 10. The result of such a *preference query* (cp. [KEW11]) on a movie data set, e.g., the Internet Movie Database[9] (IMDb) could produce a large, unclear result. In Example 1, the query would return 30 movies, cp. Table 3.1.

[9]https://www.imdb.com/

Table 3.1: Sample result of Bob's 4-dim. preference query.

ID	movie	rating	runtime	year	genres
0	Star Wars Episode V	8.8	127	1980	Action, Adventure, Sci-Fi
1	Star Wars	8.8	125	1977	Action, Sci-Fi
2	Raiders of the Lost Ark	8.7	115	1981	Action, Adventure
7	Reservoir Dogs	8.4	99	1992	Crime, Drama, Thriller
8	Blade Runner	8.3	117	1982	Drama, Sci-Fi, Thriller
13	The Terminator	8.1	107	1984	Action, Sci-Fi
22	Back to the Future Part II	7.7	108	1989	Adventure, Comedy, Sci-Fi
23	Indiana Jones II	7.6	118	1984	Action, Adventure, Fantasy
27	Die Hard 2	7.1	124	1990	Action, Thriller, Crime
...	...	...	...	...	...

The use case from Example 2 is more complex to solve with a *Pareto-dominant* clustering approach because more dimensions that result from the given features lead to a more complex decision in the *allocation process* as for a growing number of features, the possibility that no data object would dominate all other objects in all dimensions is growing as well. Also, it is possible to tune the allocation process of the clustering by *weighting* one or more favored dimensions in the case of two or more *Pareto-optima*, but this does not solve the problem in each use case, as the approach stretches to its limit when considering use cases with many features, such as those from Example 2, which requires a more precise and effective decision criterion for the allocation of objects to clusters. Also, considering the second or third-closest centroids in each dimension in the case of a *Pareto-optima* does not sufficiently improve the allocation process of the clustering and leads to a more complex approach.

While clustering *Pareto-frontiers* is not new [Hua+11; TB15; ESA12; MMB09; Kan+02; WWP88; ZZX08; JNH07], in this chapter, two approaches are presented, which can deal with unnormalized and -standardized data from Skylines automatically. The first approach, introduced in this chapter, uses the *Pareto-dominance* criterion. Compared to k-means clustering along with traditional distance measures like the Euclidean distance, this approach creates more precise and stable clusters in terms of quality in a sufficient runtime considering better-than relationships, unlike mapped Euclidean distances, which have to be adjusted inconveniently for each use case. The second approach, which exploits the *Borda Social Choice Voting Rule*, is a progression of the Pareto-dominance approach as it can cope better and independently in higher-dimensional spaces. The central aspect of this approach is considering each dimension separately to circumvent the normalization and standardization process and thus increase the degree of automation and reduce human involvement, as each feature dimension has to be sifted through.

Yet, clustering approaches like k-means provide a suitable solution for these large and confusing sets, which are encapsulated and presented clearly to the user. However, considering individual domain ranges of each dimension, traditional distance measures used commonly in k-means, e.g., the Euclidean distance, stretch to their limits, as already mentioned. Consider Example 2 again, the range of the dimension *rating* in Bob's query yields a range of almost 2 in the data set of Table 3.1, while the range of the movies' *runtime* is 28 minutes between the movies with the shortest and longest runtime (ID 0 vs. 7). Similar challenges are noticeable for the remaining dimensions *release year* and *genres*. The challenge in this use case is to set these quite diverse domain ranges into an equal relation to each other, as, e.g., the *release year* and *running time* would majorly influence the allocation in the clustering process. Reaching a more useful clustering, adjusting the feature ranges before the clustering process is inevitable (cp. [VSM15; MU13]). This adjustment might be challenging due to various and versatile user preferences. Normalization and standardization are possible, but the main question is whether to use the whole feature range of each feature dimension considering each data object or to normalize only within the local preference-based best-matching objects data set.

As k-means clustering focuses on minimizing the distances between each data object to each cluster centroid, considering each distance in each dimension separately does not deteriorate the complexity, as k-means calculates distances between each data object and each cluster. The second approach presented in this chapter splits the distance calculation to facilitate a sorting and weighting process in each dimension between each data point to each cluster by exploiting the *Borda Social Choice Voting Rule* (cp.[RVW11]) as a kind of voting process. This approach ensures that each dimension is considered equally. In contrast, the *Pareto-dominance*-based approach only adduces the closest centroids in each dimension, which minimizes the opportunity of more than one suitable cluster for each data object. Moreover, using the *Borda Voting rule* yields more balanced and smoother results than traditional distance measures, such as the Euclidean distance, as each feature dimension is considered independent of the size of their domain ranges. One more advantage of this approach is avoiding any preprocessing step, such as normalization and standardization, as this is, in most cases, a very time-consuming process. Only the selection of suitable features in terms of correlation and the data-cleaning process has to be considered.

3.1.2 Contributions

After introducing the main idea of the approaches presented in this chapter, the definition of clustering on a *Pareto-frontier* will be introduced and substantiated with the main contributions.

- When focusing on *Pareto-frontiers*, a specified data set consists of attributes not dominated by other objects in any feature dimension and, thus, are equal to each other in terms of importance. This *Pareto-Surface* is an excellent opportunity to reduce massive data sets to user-adjusted use cases instead of classical sampling. This data set is used as input for both clustering approaches. Considering a data object, at least one feature dimension always dominates the other objects and is simultaneously dominated by other objects in other feature dimensions. This aspect is getting increasingly complex for a growing number of dimensions, and thus, a subsequent clustering is realized on this set to group similar objects. Thus, clustering is a mighty aid to reducing the number of candidates in high dimensional use cases as the best case objects, which have dominating features in common, are clustered together. This sequence of techniques helps to identify potentially relevant data objects in a manageable and comprehensible way. It limits human effort, as experts can save time in preprocessing steps.

- The *Pareto-dominant* and the *Borda Social Choice*-based approaches share contributions, which this section will summarize and specify. When talking about typical Data Science and KD pipelines, an essential step is preprocessing. However, data cleaning is inevitable, as both approaches will not work with corrupt data. Nevertheless, a very time-consuming step is standardizing and normalizing the given data. A significant benefit of both approaches is the avoidance of normalization and standardization, as each feature dimension will be handled equally in the cluster allocation process. The more features are given, the more experts need to analyze them regarding skewness and variance, which can consume a lot of human knowledge and time. To reduce the overhead of normalization and standardization where values are globally set into the same bounds, a ranking-based cluster allocation technique is presented, where each dimension is considered and ranked separately according to the given cluster centroids. Thus, experts do not need to care about preprocessing and can invest their time in more important topics such as cluster analysis. Consequently, the degree of automation is also given, as no additional specifications compared to the basic k-means clustering are needed.

- When talking about *Pareto-frontiers* and *Best-Matches-Only* sets, especially in higher dimensional spaces, such as hotel booking, car purchasing, or other scenarios in terms of e-commerce or recommendations in several streaming-on-demand platforms, flooding effects in terms of the results are ubiquitous. Both experts and users can hardly handle an unsorted and large result set, as too much information can lead to indifferences. Both clustering approaches handle such data sets lucidly and appealingly, as users can compare similar results to each other. Additionally, representatives of each cluster group can

be identified to present only a few results to the users. Especially the use case of recommender systems is especially appealing, as recommendations suppose a reduced set of objects, which should match the users' desires perfectly. The traditional approach of recommendations and *Pareto-frontiers* aim for a lucid data set and thus coincide in this aspect as candidates are reduced to a minimum, which is also associated with reducing human involvement.

- *Pareto-frontiers* form an approach that can also be compared to fine-grained micro-targeting-based approaches, as both can handle different inputs with various diverse and manifold feature dimensions. Both approaches are a kind of personalization, but only preference-based *Pareto-frontiers* deliver the best-matching results, while most micro-targeting approaches present results based on crawled data and thus also deliver unwanted results, which mostly do not match perfectly.

3.2 Methodology

Before introducing the methodology of the Pareto-dominance approach, which works best for lower dimensional data sets, and the *Borda-Social Choice* approach as a progression from two-dimensional to n-dimensional feature spaces, further background knowledge needs to be introduced.

3.2.1 Further Background Knowledge on Preferences

First, a brief introduction to preference theory is given, as preferences can represent desires for specific features in an order. This order is described with numerical values, a significant part of clustering, as distances can be calculated easily. Furthermore, the *Pareto-frontier* is one of the central application fields in this chapter, as it produces *Best-Matching-Only* data sets, which are the starting point for all scenarios in this chapter. Moreover, *similarity measures*, already introduced in Section 2.3, will be discussed briefly as they are crucial for analyzing the novel approaches, as comparisons to the basic k-means clustering are needed to substantiate the approach's benefits. As k-means clustering works with several distance measures, several of them were introduced in Section 2.3. As a significant step in the proposed approaches, k-means clustering, an iterative partitional clustering (cp. Section 2.6.1.2), is adapted. Furthermore, k-means++, a version of k-means where the initial partition is adjusted with seeding, was also reviewed, as it improves the results in the clustering process.

As the approaches presented in this chapter all focus on exploiting wishes, an order has to be modeled. A suitable approach is to use *preferences*, which can map wishes

in a numerical range. A preference can be described as "*I like y more than x*," which leads to a formal definition of $x <_P y$. In theory, a preference $P = (A, <_P)$ is defined as a strict partial order on the domain of A, $dom(A)$. To construct preferences, there are several *base preference constructors*, as well as two *complex preference constructors* published in [Kie02; KEW11]. The most helpful preference constructors are presented in the following paragraphs.

Base Preferences The so-called *base preferences*, seen in Fig. 3.3, represent *wishes* in a specific *domain* and are described as features. These constructors, introduced in [Kie02; KEW11], are the base to build *Pareto-frontiers* intuitively. There are several constructors that model numerical preferences for managing features like the running time of a movie or year. Nevertheless, *categorical constructors* for managing sets of, e.g., genres or favored actors, can also be handled. Moreover, some further constructors also can model *geospatial preferences*. Restricting the attention to *LOWEST/HIGHEST* as input preferences, *Pareto* preference queries coincide with the traditional *Skyline* queries introduced in [BKS01]. To handle categorical domains, such as data sets, there are also preference constructors, e.g., *LAYERED*, where sets of values are defined according to their most, second, etc., preferred values.

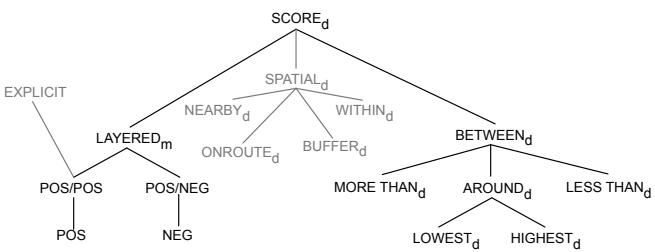


Figure 3.3: Taxonomy of base preferences.

Pareto Preference For the approach presented in this chapter, the most important preference is the well-known *Pareto* preference, which models *equal importance* where base preferences are used to build *Pareto* preferences intuitively. A Pareto preference $P := P_1 \otimes P_2 = (A_1 \times A_2, <_P)$ with preferences $P_i = (A_i, <_{P_i})$ and tuples $x = (x_1, x_2), y = (y_1, y_2) \in dom(A_1) \times dom(A_2)$ is defined as follows:

$$\begin{aligned}(x_1, x_2) <_P (y_1, y_2) \Leftrightarrow (x_1 <_{P_1} y_1 \wedge (x_2 <_{P_2} y_2 \vee x_2 = y_2)) \vee \\ (x_2 <_{P_2} y_2 \wedge (x_1 <_{P_1} y_1 \vee x_1 = y_1))\end{aligned} \tag{3.1}$$

Discussion w.r.t Sets Each domain relies on numerical values and sets of movies or genres, so they must be processed before Preferences can be applied. The following example shows the drawbacks of traditional distance measures like the Euclidean distance in data sets with diverse feature domain ranges.

Example 3
Consider Example 2. Assume a clustering with $k = 3$ clusters is favored, and the movies with the IDs (1), (7), and (23) are chosen as initial centroids. The movie with ID (27) should be allocated to one of the clusters using k-means with the Euclidean distance on the attributes *rating*, *running time*, *release year*, and *genres*.

The *Jaccard* coefficient from Section 2.3 is used to determine the distance for categorical attributes like *genre*, e.g., $J(\text{genres}_{ID=1}, \text{genres}_{ID=27}) = \frac{1}{4}$ for the movie (1) and (27). Thus, the distance between the movie (1) and (27) is then given by

$$dist(1, 27) = \sqrt{(8.8 - 7.1)^2 + (125 - 124)^2 + (1990 - 1977)^2 + (1 - 0.25)^2} = 13.2$$

and shows that the (large) domain range of the year significantly influences the distance calculation. Finally, movie (27) would be allocated to the cluster with centroid (23) because of the lowest distance of only $dist(23, 27) = 8.5$.

3.2.2 Pareto Dominance Clustering

After introducing further basic knowledge, the *Pareto-dominance* framework is described in detail for two-dimensional use cases. While a *Pareto* preference determines the importance of preferences, the *Pareto-dominance* in the proposed approach is used to allocate an object to the possibly best cluster, which is not dominated by other clusters w.r.t. the distances of the individual objects, by using the Euclidean distance for one-dimensional distances. Furthermore, the *Pareto-dominance* can additionally be used to find a representative of each cluster on the Pareto-frontier, which has the closest distance to the calculated centroid of the cluster. This representative can be used as a new centroid for the next iteration.

3.2.2.1 Cluster Allocation

Consider the *Pareto-frontier* from Fig. 3.4, presenting users w.r.t. a possibly high music-matching score and a possibly close distance. The goal is to get three promising clusters C_1, C_2 and C_3. For initialization, the points P_2, P_5 and P_8 are chosen as cluster centroids, marked as violet diamonds. Continuing, for each point $P_1, ..., P_{10}$ the particular distances of both the x- and the y-dimension to the cluster centroids are calculated, which can be seen in Table 3.2. Moreover, the y-dimension, representing

the music-matching score, is chosen as more significant than the x-dimension, being considered, if *Pareto-optimal* cluster centroids appear in the allocation process. This so-called *one-dimensional clustering* realizes that a decision is made for each object at the cluster allocation. Fig. 3.4 shows a clustering after the first iteration.

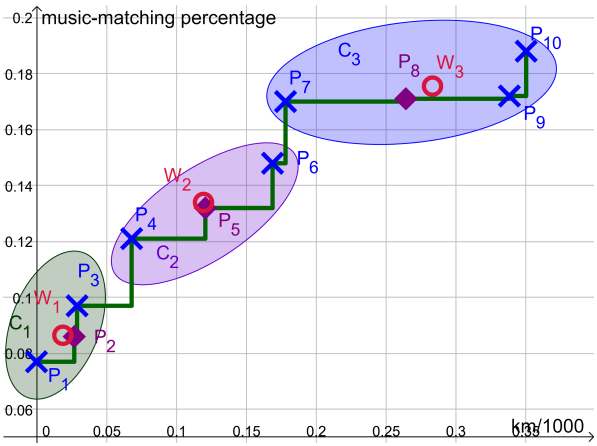


Figure 3.4: Clustering of users with the *Pareto-dominance* after the first iteration.

Table 3.2: Distances of each user to each cluster centroid. $C_1 := P_2, C_2 := P_5, C_3 := P_8$, x-dim.: distance, y-dim.: music matching score.

	$\mathbf{d_{x_{C_1}}}$	$\mathbf{d_{y_{C_1}}}$	$\mathbf{d_{x_{C_2}}}$	$\mathbf{d_{y_{C_2}}}$	$\mathbf{d_{x_{C_3}}}$	$\mathbf{d_{y_{C_3}}}$
$\mathbf{P_1}$	**27.44**	**0.01**	120.72	0.06	264.60	0.09
$\mathbf{P_2}$	**0.00**	**0.00**	93.27	0.05	237.16	0.09
$\mathbf{P_3}$	**1.66**	**0.01**	91.61	0.04	235.50	0.07
$\mathbf{P_4}$	**41.27**	0.04	52.00	**0.01**	195.89	0.05
$\mathbf{P_5}$	93.27	0.05	**0.00**	**0.00**	143.89	0.04
$\mathbf{P_6}$	141.27	0.06	**48.00**	**0.02**	95.89	0.02
$\mathbf{P_7}$	150.88	0.09	**57.61**	0.04	86.28	**0.00**
$\mathbf{P_8}$	237.16	0.09	143.89	0.04	**0.00**	**0.00**
$\mathbf{P_9}$	311.32	0.09	218.05	0.04	**74.16**	**0.00**
$\mathbf{P_{10}}$	323.08	0.10	229.81	0.06	**85.92**	**0.02**

The fundamental rules will be explained in more detail after introducing the general approach with a brief example.

- Users P_1 and P_3 are assigned to cluster C_1 as the centroid of C_1 has a closer distance to the other two centroids regarding the distances in both dimensions.

- P_4 has *2 Pareto-optima*, because of the closer distance to C_1 regarding the x-dimension and to C_2 regarding the y-dimension. Hence C_1 and C_2 are *Pareto-optimal* w.r.t. to the x- and y-dimension. Now the one-dimensional clustering tips the balance to C_2.

- P_6 and P_8 are allocated to Cluster C_2, because of the closer distances to the centroid of C_2 in both dimensions.

- P_7 is closer to C_2 concerning the x-dimension but has a smaller distance to the centroid of C_3 w.r.t. the y-dimension. This ensures that P_7 is allocated to C_3.

- P_8 and P_{10} are allocated to cluster C_3 because of the existence of only one Pareto-dominant cluster centroid, namely C_3.

To explain the proposed approach more in detail, the attention is drawn to Fig. 3.5, which shows a snippet of the *Pareto-frontier* of Fig. 3.4. While P_4 is assigned to C_1 regarding the smaller Euclidean distance of 41.27, unlike a distance of 52.00 to C_2, the versatility of the Pareto-dominance approach is shown. The user can influence the clustering by choosing one dimension as the more important at the appearance of Pareto-optima. Choosing the x-dimension as more important, P_4 will be assigned to C_2. Thus, each data point will be allocated to one and only one cluster to avoid overlapping and imprecise clusters. This use case shows that a Pareto-dominant clustering combined with a one-dimensional clustering at the appearance of *Pareto-optima* tends to a kind of hierarchical clustering because users with similar scoring values w.r.t. the music matching score are clustered together, unlike in the primary k-means clustering. In particular, cluster C_1 and C_2 contain users with similar music-matching scores, where the range between the two boundary points is petite, unlike the k-means clustering approach. So if P_7 is allocated to C_2 and P_4 to C_1, the users contained in the clusters are not as similar as in the proposed approach.

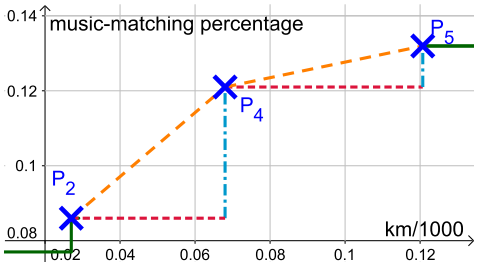


Figure 3.5: Comparison of *Pareto-dominance* and Euclidean distance. $C_1 := P_2, C_2 := P_4$.

3.2.2.2 Cluster Centroids

As mentioned in Section 2.6.1.2, there are two main possibilities to determine the cluster centroids. The primary k-means clustering calculates the centroids by averaging all data points. For this implementation, an approach similar to k-Medoids, also mentioned in Section 2.6.1.2 was also considered but did not significantly affect the results. After allocating each data point to a cluster, the cluster centroids are recalculated considering the contained users. For each cluster, the x-dimension and y-dimension are averaged for all values, which can be seen in Fig. 3.4 as W_1, W_2 and W_3.

For each cluster-centroid W_1, W_2 and W_3, the closest Pareto-dominant user in each cluster is selected as the new cluster-centroid for the next iteration. The particular distances regarding the two dimensions are calculated to find these users, as shown in Table 3.3. Fig. 3.4 shows the calculated centroids before assigning the new centroids.

- P_2 is the new cluster centroid of C_1, because of the closest distance of each x- and y-dimension to W_1.
- For cluster C_2 P_5 is allocated as a new centroid because of the closer distances in both dimensions, too.
- P_8 and P_9 both are Pareto-optima for the allocation of the cluster centroid of C_3 because P_9 is closer to W_3 regarding the y-dimension and P_8 regarding the x-dimension. The one-dimensional clustering determined by Bob tips the balance to P_8 as a new cluster centroid.

Table 3.3: Particular distances of recalculated cluster-centroids W_1, W_2, W_3 to the points in each cluster.

	$\mathbf{d_{x_{W_1}}}$	$\mathbf{d_{y_{W_1}}}$	$\mathbf{d_{x_{W_2}}}$	$\mathbf{d_{y_{W_2}}}$	$\mathbf{d_{x_{W_3}}}$	$\mathbf{d_{y_{W_3}}}$
$\mathbf{P_1}$	18.848	0.010	—	—	—	—
$\mathbf{P_2}$	**8.595**	**0.001**	—	—	—	—
$\mathbf{P_3}$	10.253	0.010	—	—	—	—
$\mathbf{P_4}$	—	—	50.667	0.012	—	—
$\mathbf{P_5}$	—	—	**1.331**	**0.002**	—	—
$\mathbf{P_6}$	—	—	49.331	0.014	—	—
$\mathbf{P_7}$	—	—	—	—	104.730	0.004
$\mathbf{P_8}$	—	—	—	—	**18.451**	0.004
$\mathbf{P_9}$	—	—	—	—	55.709	**0.003**
$\mathbf{P_{10}}$	—	—	—	—	67.4726	0.0120

3.2.2.3 Complexity

The implementation of the proposed Pareto-dominant clustering approach was realized as a Java program and reaches a complexity of $\mathcal{O}(n \cdot c \cdot d)$ where n is the number of d-dimensional points that should be clustered in c clusters. In each iteration, i for every point n, the distances to each cluster c are calculated in $\mathcal{O}(c)$, and the best centroid is chosen after the distance calculation. Compared to the basic k-means clustering and k-Medoids clustering, a better or at least equal complexity is reached.

3.2.2.4 Discussion

This section introduced a promising solution for a two-dimensional use case. The benchmarks showed that the proposed approach is competitive to the basic k-means clustering along with the Euclidean distance. While the *Pareto-dominance* as a decision criterion for k-means clustering works well in this use case, in higher dimensions, Pareto-dominance is stretching to its limits because of a higher probability of occurring Pareto-optima for a growing number of dimensions. As use cases exploiting user preferences do not only focus on two feature dimensions, this approach is scarcely considerable, as the proposed decision criterion in *Pareto-optimal* cases in the allocation process only considers a predefined feature dimension. Thus, a distortion of the clustering can occur. Thus, a more precise decision criterion has to be found for allocating each point to one and only one cluster. The following section presents a more versatile approach to handle this challenge more precisely. This approach works with an ordering process and assigns votes according to this order in each dimension separately. Thus, no predefined feature dimension is needed in the case of *Pareto-optimal* dimensions.

3.2.3 Borda Social Choice Clustering

This section presents the novel *Borda social choice clustering* approach. Social choice deals with aggregating individual preferences for managing social assessments and ruling. The *Borda* social choice voting rule is omnipresent in political or other elections, e.g., the Eurovision Song Contest. The Social Choice was founded in the 18th century and was first published by Jean-Charles de Borda and Marquis de Condorcet [Sen99].

As mentioned in [Deb92], the *Borda Social Choice Voting Rule* is a very appealing approach to considering each dimension in a multi-dimensional scenario equally. This rule can be used for the allocation of objects to one and only one cluster and, therefore, allows more influence on smaller domain ranges. The *Borda Social Choice Voting Rule* is a promising method for the proposed approach because every candidate receives equal weighted votes from each voter.

Given k candidates C_i, and d voters V_j, where each *voter* votes for each *candidate*. Each voter has to allocate the *voting* $v_{jm} \in \{0, ..., k-1\}$, $m = 1, ..., k$, where all v_{jm} are pairwise distinct. After all, voters assigned their votes, the votes for each candidate are *summed up* as it can be seen in Eq. 3.2, while the *Borda winner* is determined as depicted in Eq. 3.3.

$$\text{bordaSum}_{C_i} = \sum_{l=1}^{d} v_{li} \tag{3.2}$$

$$\text{bordaWinner} = \max\{\text{bordaSum}_{C_i} \mid i = 1, ..., k\} \tag{3.3}$$

Suppose this rule is applied to the proposed clustering framework. In that case, the *candidates* correspond to the available *clusters* and the *voters* to the *dimensions of the d-dimensional object, which should be allocated to a cluster*. Then, for each dimension, votes are assigned for the distances between the object and the centroids of the clusters. While the closest distance receives a maximum vote of k-1, the second closest gets a vote of k-2, etc., and the most enormous distance obtains a vote of 0, where k is the number of desired clusters. After the voting, Eq. 3.2 determines the sum of all votes for each cluster, and subsequently, Eq. 3.3 identifies the winner. Therefore, *dimensions* that would not be equally considered because of a smaller or larger domain range, e.g., using a distance measure like Euclidean, get *equal weighted votes* like the other dimensions and significantly influence the clustering process. To clarify the principle of the *Borda Social Choice Voting Rule*, Example 4 shows the allocation of an object to other representatives of clusters.

Example 4
Reconsider Example 2. Table 3.4 shows the Borda social choice cluster allocation for the movie (27). The *centroids* of the initial clusters C_1, C_2, C_3 are the movies with the IDs (1), (7), and (23). The *distances* between movie (27) and the centroids are calculated for each dimension. The *Borda* votes are depicted in parentheses, e.g., the dimension *rating* is closest to C_3 and therefore gets a vote of $k - 1 = 2$. The second closest centroid is C_2 with vote 1, and C_1 gets the vote 0. Finally, C_2 with movie (7) as the initial centroid is determined as the Borda winner with a *Borda* sum of 5, cp. Eq. 3.2 and Eq. 3.3. Compared to the Euclidean distance, more concise results for the cluster allocation are obtained due to ranking the values in each dimension according to their closeness. Note that a Jaccard coefficient of 1.0 is the best value for the *genre*.

Table 3.4: Cluster allocation for movie (27) with Borda voting rule.

	Movie (27): Die Hard		
Dimension	**C_1**	**C_2**	**C_3**
rating	1.70 (0)	1.30 (1)	**0.50 (2)**
running time	**1.00 (2)**	25.00 (0)	6.00 (1)
release year	13.00 (0)	**2.00 (2)**	6.00 (1)
genre	0.25 (1)	**0.50 (2)**	0.20 (0)
$\sum$	3	**5**	4

3.2.3.1 Borda Clustering Algorithm

The classic *k-means* algorithm from Section 2.6.1.2 is modified to realize a clustering with the *Borda Social Choice Voting Rule* as a decision criterion for the cluster allocation. For this, Step 2) of k-means is replaced by the *Borda rule*, where the distances of each object to the available clusters are calculated and allocated afterward. This allocation is described in Function 1, which finally returns the centroids id the object should be allocated to.

For managing the *Borda Social Choice Voting Rule*, an object array *votes[]* is used to save the centroid IDs and *Borda values*. As further information for each object $x = (x_1, ..., x_d) \in X$, an identifier id_{last} of the allocated centroid from the previous iteration is necessary.

In Line 2, the array *votes[]* is set to the *bordaSum* values from Equation 3.2. In detail, in each dimension, the distances between the considered object x and each cluster centroid of C are calculated, saved with the centroids id in an object-based data structure, and appended to an object array. Once all distances in the current dimension

Function 1 Determine Borda Winner

Input: d-dim. object $x = (x_1, ..., x_d)$, centroids C, cluster-id last iteration id_{last}.
Output: id of the closest cluster for object $x = (x_1, ..., x_d)$.

```
1: function getBordaWinner(x, C, id_last)
2:     votes[] ←calculateBordaSum(x, C) //determine & sum up votes.
3:     id= analyzeBordaWinners(votes[], id_last) //analyze all Borda winners.
4:     return id
5: end function
```

are calculated, this object array is sorted ascending according to the distances to assign the Borda votes from 0 to k-1. After the sorting, the votes for each cluster are determined and summed up in the array *votes[]* overall dimensions.

Subsequently, the *Borda winner(s)* with the highest score in the array *votes[]* (Line 3) are found, and afterward, the corresponding cluster id is returned in Line 4. If there is more than one *Borda winner*, the winner is randomly chosen. After the object x got allocated to the centroid with the identifier id, the clustering continues with Step 3) of k-means. Note that this approach will be called *Borda* for further purposes. There are some improvements w.r.t. the convergence, which will be discussed in the next section.

3.2.3.2 Convergence

When discussing clustering, *convergence* is a major topic. In [Mac67; Jai10], it was shown that k-means could only converge to a local optimum (with some probability to a global optimum when clusters are well separated). The proposed algorithm is based on k-means and only uses another kind of "*distance measure*". Therefore, the convergence proof is similar to the one of the k-means.

Proof of convergence. There is only a finite number of ways to partition n data points into k clusters [Mac67; Jai10]. A new clustering based *only* on the old clustering is produced for each iteration of the proposed algorithm. In addition, it holds that

1) If the old clustering is the same as the new, the next clustering will be the same again. Thus, a kind of fixed point is reached.

2) If the new clustering differs from the old one, the newer one has a lower cost (due to better overall voting).

Since the algorithm iterates a function whose domain is a finite set, the iteration must eventually enter a cycle. The cycle cannot have a length greater than 1 because, otherwise, by 2), one would have some clustering, which has a lower cost than itself, which is impossible. □

Therefore, k-means using the *Borda Social Choice Voting Rule* converges in a finite number of iterations to a local solution but does not permit eliminating the compelling possibility that a point oscillates indefinitely between two clusters. Indeed, after some preliminary tests, especially for higher dimensions and a higher number of clusters, some problems regarding the convergence of the approach were noticed. To solve this problem, a decision criterion for the cluster allocation was added if there is more than one *Borda winner*.

In detail: For each iteration, the IDs of the cluster objects the object got allocated to are saved. Assume there is more than one *Borda winner* in the next iteration, the allocation to the centroid from the last iteration id_{last} (Line 3 of Function 1) is consulted. If so, the object goes to the same cluster as in the last iteration. As the benchmarks show, this solution ensures the clusters become stable in a few iterations.

Another problem concerns the initial partition, which could result in empty clusters. If the first centroid was randomly chosen, the probability that a pretty similar object to the first centroid would be chosen is minuscule but possible. Especially if there are, e.g., different movies with almost the exact specifications, the possibility is given that these movies are chosen as cluster centroids. Then, the order of the cluster centroids decides that the first of the clusters will be occupied with objects, while the following cluster with a similar centroid will stay empty. *k-means++* minimizes these problems and cares that the runtime and the number of iterations will decrease. This comprehensive approach considering convergence and empty clusters is called *Borda++*.

3.2.3.3 Complexity

The algorithm's complexity is given by $\mathcal{O}(ndk \cdot k \log(k) + k)$ where each n d-dimensional object should be clustered in k clusters. The algorithm calculates the distances of the dimension d for each object for each cluster k in $\mathcal{O}(ndk)$. Depending on the *sorting algorithm*, the distances are sorted, e.g., by *Quicksort* in $k \cdot \log(k)$ [Hoa61]. Finally, the *Borda winners* are determined in $\mathcal{O}(k)$. Hence a *complexity* of $\mathcal{O}(nd \cdot k^2 \cdot \log(k))$ is reached. Compared to the basic k-means clustering, this approach reaches a different *complexity*. In Section 3.3, the proposed approach will be evaluated against the basic k-means clustering considering *runtime* and the number of needed *iterations* to show the benefits even though the complexity does not.

3.3 Synthetic Experiments

Experiments on several data sets are performed to validate the benefits of the approaches as they show competitiveness. Benchmarking is an effective procedure to compare approaches against each other. In this section, the benchmark settings are briefly described. Afterward, results of experiments regarding *runtime*, number of *iterations*, and quality of the clustering approaches compared to the basic k-means approach with traditional distance measures both for the *Pareto-dominance* clustering approach and the *Borda Social Choice* clustering approach are presented.

3.3.1 Benchmark Settings

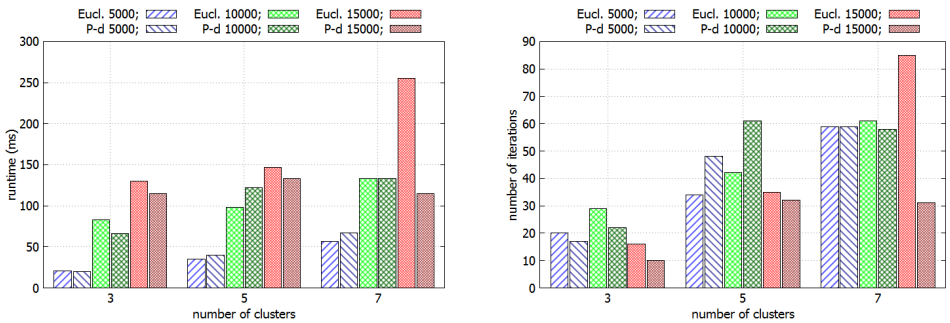
Both approaches of the experiments were performed on an Intel Xeon machine with 2.53 GHz and 44 GB RAM. To compare the runtimes, several synthetic data sets of *independent* and *anti-correlated multi-dimensional Pareto-optimal* points were created using a data generator, as described in [BKS01], as they form a unique use case in terms of *Pareto-optimal* data sets. Clusterings were performed in test rows with 1000 repeats to gain averaged reliable data. Furthermore, the number of *dimensions*, the number of *data objects per set*, and the *number of desired clusters* were also varied. The main aim of the benchmarks was to evaluate both runtime and number of iterations of the proposed approaches until a stable clustering is reached considering the following clustering techniques.

3.3.2 Benchmarks Pareto-dominance

First, the benchmarks of the *Pareto-dominance* implementation are introduced and discussed. The proposed *Pareto-dominance* clustering approach was evaluated against the basic k-means clustering algorithm with Euclidean distance in terms of *runtime* and needed *iterations* until *convergence*.

Runtime The benchmarks of the Java implementation in Fig. 3.6a show that the approach using the *Pareto-dominance* is mostly similar regarding the *runtime* compared to the basic k-means approach using the Euclidean distance. The average clustering *runtime* is growing for both approaches for constant clusters and a growing number of points. Whereas for constant numbers of points and growing clusterings, there are some aberrations at $k = 7$ for 15000 data objects for both approaches. Significantly, if points at the border of the cluster switch between two clusters, the runtime grows. Overall, the clustering approach using the *Pareto-dominance* is nearly as efficient as using the Euclidean distance.

Iterations The number of *iterations* w.r.t., the number of desired clusters, and the number of points can be seen in Fig. 3.6b. For both frameworks, the number of *iterations* is similar. For a growing number of desired clusters, the number of *iterations* is growing for both approaches, except for the sets of $k = 5, 7$ with 15000 points using the *Pareto-dominance*. Contrary to the expectations for this experiment, the number of points in the sets is independent of the number of necessary iterations to achieve a stable clustering.



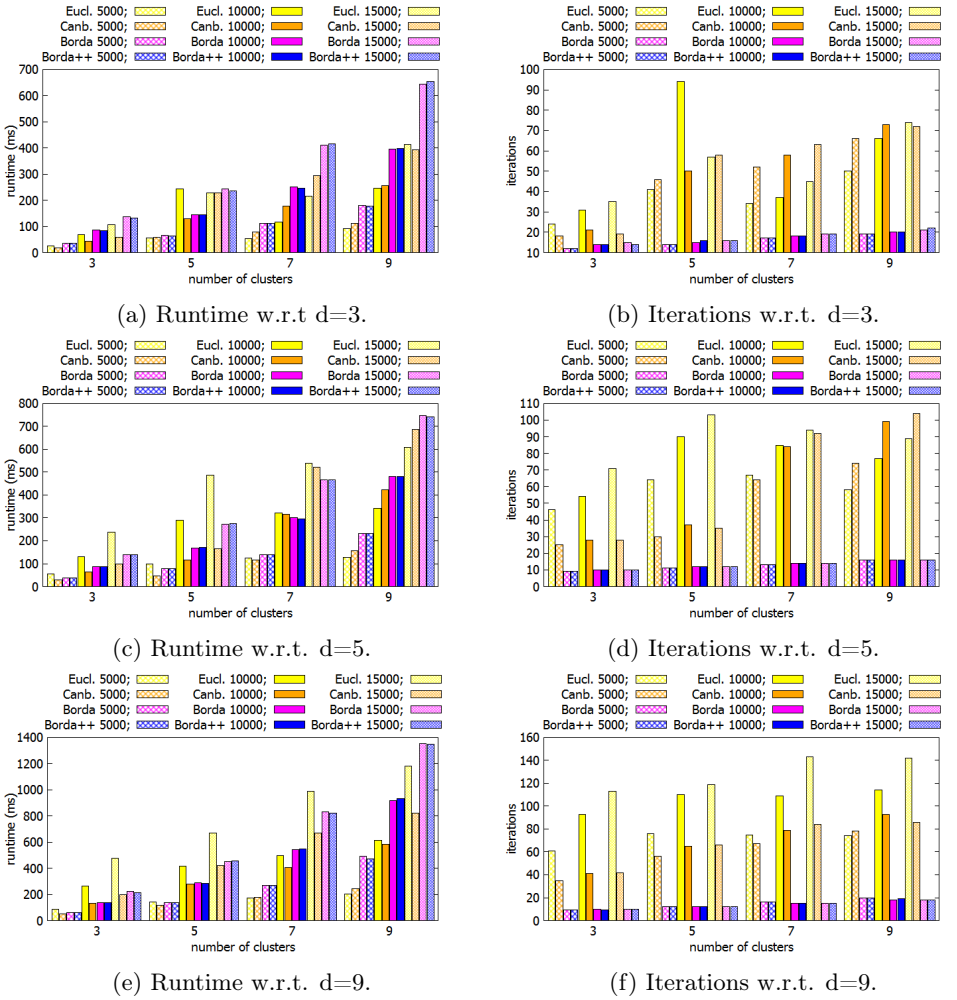
(a) Runtime. (b) Iterations.

Figure 3.6: Evaluation of Euclidean distance(Eucl.) vs. *Pareto-dominance*(P-d)

3.3.3 Benchmarks Borda-Clustering

Since Euclidean is the most common distance for k-means clustering, the main goal to show is that the proposed approach terminates at least as fast as k-means and needs the same or fewer iterations until termination. Furthermore, to consider also a distance that inhibits a kind of normalization, Canberra distance was evaluated to gain reference values, which should be dominated by them of the *Borda approach* w.r.t. the *runtime* and the number of *iterations*. To receive a faster runtime and fewer iterations until stable clusterings, the utility of k-means++ for the Borda approach w.r.t. the runtime and number of iterations were also considered. The following settings for the experiments were considered:

- *Eucl.*: *k-means* with *Euclidean distance* for cluster allocation.
- *Canb.*: *k-means* with *Canberra distance* for cluster allocation.
- *Borda*: *k-means* with *Borda voting rule* for cluster allocation.
- *Borda++*: *k-means++* with *Borda voting rule* for cluster allocation.



(a) Runtime w.r.t d=3.

(b) Iterations w.r.t. d=3.

(c) Runtime w.r.t. d=5.

(d) Iterations w.r.t. d=5.

(e) Runtime w.r.t. d=9.

(f) Iterations w.r.t. d=9.

Figure 3.7: Evaluation of runtime and iterations for Borda Clustering.

Runtime In Fig. 3.7a the number of clusters ($k = 3, 5, 7, 9$) and the data size, i.e., 5000, 10000, and 15000 input objects were considered, for the clustering were varied. In this 3-dimensional case, increasing the number of clusters and the input data leads to an increasing runtime. The *Borda* approach works in equal time compared to *k-means with Euclidean* (Eucl.) and *Canberra* (Canb.) for small numbers of clusters. Because of higher complexity, the *Borda* approach is slower for 7 and 9 clusters, independent of

the number of input objects. The benefits of a faster runtime for *k-means++* are hardly recognizable in most cases considering *Borda++*. Fig. 3.7c presents the results on a 5-dimensional domain. For increasing numbers of dimensions, *Borda* reaches a better runtime compared to *Eucl.* except for a high number of clusters. In some cases, *Borda* terminates faster than *Canb.*, e.g., the test series with 7 clusters, but *Borda* mainly reaches an equal runtime in a 5-dimensional space. A similar behavior illustrates the test series for a 9-dimensional set of objects in Fig. 3.7e. While both *Borda* and *Borda++* terminate in similar time compared to *Canb.* for 3 and 5 clusters, they are a lot faster than *Eucl.*. The trends for growing runtimes w.r.t. the number of clusters and objects can also be noticed in 9-dimensional space. Further experiments have shown that the runtime increases with more objects and clusters in higher dimensions.

Iterations This section shows the number of iterations necessary to reach a stable clustering, cp. Fig. 3.7b are considered. The experiments indicate that for an increasing number of clusters and objects (cp. Fig. 3.7d and 3.7f), the number of iterations until termination increases, too. However, *Borda* and *Borda++* reach a stable clustering in clearly less iterations than *Eucl.* and *Canb.* In higher dimensions ($d > 9$), the number of iterations is increasing slightly for a growing number of objects and desired clusters. Thus, the seeding performed with the k-means++ algorithm has only a tiny effect on the number of iterations for *Borda++* compared to Borda. In summary, the *Borda* approach has a runtime similar to the classic k-means algorithm but only needs a fractional part of iterations until termination for all test series. Therefore, using traditional distance measures, *Borda* and *Borda++* can be considered competitive to k-means and k-means++.

3.3.4 Discussion & Comparison of the Results

The *Pareto-dominance* clustering and the *Borda Social Choice* clustering approach have a similar idea considering the cluster allocation, as both approaches do not need traditional distance measures in the allocation process. As the *Pareto-dominance* approach has its drawbacks regarding a growing number of feature dimensions, the *Borda Social Choice* clustering approach does not need to care about *Pareto-dominant* distances in the allocation process, as they are sorted and weighted afterward. Thus, a higher complexity is given for the *Borda Social Choice* clustering approach, and unsurprisingly, the runtime is also growing when comparing the runtimes of the *Pareto-dominance* approach, which was evaluated in a two-dimensional use case to the experiments of the 3-dimensional use case considering the *Borda Social Choice* clustering. A significant benefit is the number of needed iterations, which is only marginally growing for a growing number of dimensions, data objects, and desired

clusters. Compared to the *Pareto-dominance* clustering approach, the weighting and careful seeding using k-means++ in the Borda Social Choice clustering assort well together and deliver a more convenient approach.

3.4 Quality Experiments

In the context of clustering algorithms, the question of *quality* often arises. However, it is difficult to compare the results of different approaches and to determine the *quality* of clustering methods [Kni+12; KP17]. To evaluate *quality*, the clustering evaluation indicators *Silhouette* and the *Davies-Bouldin Index*, which were already introduced in Section 2.6.2, were considered to measure the *internal quality*, i.e., if a clustering has a *high intra-cluster similarity* and a *low inter-cluster similarity*.

3.4.1 Settings

To test the quality, a movie recommender system was developed [KRE19] based on the JMDb movie database, a Java-based alternative interface[10] of the IMDb. The prototype recommends clusters of movies based on the user's preferences and allows a comparison of the clustering techniques. This system was used to filter all movies w.r.t. the following preferences:

- Scenario (S1): *Action and comedy movies of the 2000s to the present day. Running time between 60 and 120 minutes. Rating between 6 and 10.*
- Scenario (S2): *Drama, thriller and crime movies during the 90s and 2000s. Rating between 8 and 10.*
- Scenario (S3): *Classic-movies of the 70s and 80s. Release year between 1975 and 1989. Running time between 90 and 150 minutes. Action-, adventure movies, and dramas as genres.*

The chosen scenarios were built on common user preferences considering movies of the last 40 years. To determine valuable values for the *Silhouette (Sil.)* and *Davies-Bouldin Index (DB)*, the experiments were evaluated on $k \in \{3, 5, 7, 9\}$ w.r.t. the results of the scenarios. To reach a significant mean value for the experiments, 1000 runs were performed for each scenario.

[10] http://www.jmdb.de/

3.4.2 Execution

Note that the *Borda* clustering approach is not generally metric but uses an *assignment function*. Therefore, no appropriate numerical distance measure for *Silhouette* and *Davies-Bouldin* could be found for the *Borda* clustering approach. Both the *Euclidean* and *Canberra* distances were considered in both indicators to evaluate the quality, even though this needs to be revised for *Borda* and leads to a bias. Tables 3.5, 3.6 and 3.7 show the results for the scenarios *(S1)*, *(S2)* and *(S3)*.

The leftmost column in all tables represents the algorithm used for clustering, i.e., *k-means++* with the *Canberra (Canb.)*, *Borda++*, or *Euclidean (Eucl.)* measure. For each algorithm, the *Silhouette* and *Davies-Bouldin* quality indicator, one time with the *Canberra* distance, the other time with the *Euclidean* distance, were computed.

Unsurprisingly, *Canb.* performs best with the *Canberra* distance and *Eucl.* provides the best internal quality using the *Euclidean* distance for all evaluated scenarios. This is because the clusters are computed using the corresponding distance measure, and therefore, the quality indicators also compute a high intra-cluster similarity.

For example, consider Table 3.5, where *Silhouette* leads to an internal quality of 0.484 (*Canberra* dist.) on the score of -1 to 1, and therefore $k = 9$ would be best in this case. Also, *Davies-Bouldin*, having a value of 0.839, is best for $k = 9$. This is not the case with *Borda* because neither the *Euclidean* nor the *Canberra* distance fits the *Borda* assignment function. Nevertheless, the observations show that the internal quality of *Borda* always gets values between *Canb.*. and *Eucl.* In addition, the Borda approach reaches more reasonable values than *Canb.* using the *Euclidean* distance or *Eucl.* with the *Canberra* distance. Thus, the approach is adequate for the *intra-cluster* and *inter-cluster similarity (Davies-Bouldin Index)* as well as the *coherence* of the clusters *(Silhouette)*.

Table 3.5: Quality measures using Sil. and DB Index for Scenario (S1).

		with Canberra dist.				**with Euclidean dist.**			
Alg.	**Indic.**	**k=3**	**k=5**	**k=7**	**k=9**	**k=3**	**k=5**	**k=7**	**k=9**
Canb.	**Sil.**	**0.430**	**0.470**	**0.479**	**0.484**	-0.028	-0.119	-0.150	-0.153
Canb.	**DB**	**1.080**	**1.029**	**0.896**	**0.839**	10.570	11.230	8.774	7.642
Borda++	**Sil.**	0.106	0.091	0.073	0.067	0.062	-0.003	-0.036	-0.050
Borda++	**DB**	2.171	2.163	2.279	2.295	2.627	2.784	2.804	2.815
Eucl.	**Sil.**	0.135	0.038	-0.024	-0.060	**0.457**	**0.405**	**0.411**	**0.421**
Eucl.	**DB**	3.850	4.552	5.115	5.878	**0.775**	**0.881**	**0.836**	**0.799**

Table 3.6: Quality measures using Sil. and DB Index for Scenario (S2).

		with Canberra dist.				with Euclidean dist.			
Alg.	**Indic.**	**k=3**	**k=5**	**k=7**	**k=9**	**k=3**	**k=5**	**k=7**	**k=9**
Canb.	**Sil.**	**0.591**	**0.605**	**0.650**	**0.679**	-0.050	-0.113	-0.171	-0.221
Canb.	**DB**	**0.602**	**0.567**	**0.531**	**0.501**	15.003	18.256	18.195	18.837
Borda++	**Sil.**	0.090	0.001	-0.040	-0.065	0.141	-0.010	-0.086	-0.103
Borda++	**DB**	2.552	2.892	3.194	3.241	2.394	2.908	3.174	3.155
Eucl.	**Sil.**	-0.026	-0.087	-0.192	-0.308	**0.592**	**0.562**	**0.547**	**0.531**
Eucl.	**DB**	7.070	13.879	15.768	16.541	**0.513**	**0.551**	**0.606**	**0.618**

Table 3.7: Quality measures using Sil. and DB Index for Scenario (S3).

		with Canberra dist.				with Euclidean dist.			
Alg.	**Indic.**	**k=3**	**k=5**	**k=7**	**k=9**	**k=3**	**k=5**	**k=7**	**k=9**
Canb.	**Sil.**	**0.507**	**0.541**	**0.591**	**0.612**	0.001	-0.024	-0.034	-0.020
Canb.	**DB**	**0.930**	**0.875**	**0.703**	**0.591**	7.493	5.353	4.370	3.928
Borda++	**Sil.**	0.132	0.135	0.123	0.133	0.127	0.117	0.116	0.131
Borda++	**DB**	1.748	1.730	2.040	2.012	2.022	1.944	2.007	1.881
Eucl.	**Sil.**	0.110	0.009	-0.072	-0.138	**0.436**	**0.430**	**0.453**	**0.465**
Eucl.	**DB**	4.486	4.811	5.315	5.414	**0.820**	**0.762**	**0.672**	**0.624**

3.5 Interpretation & Discussion of Results

After introducing and discussing the results of the proposed approaches considering synthetic and quality experiments, the lessons from both approaches are drawn. Afterward, some use cases for the *Borda Social Choice* clustering approach are introduced in more detail.

3.5.1 Interpretation of Results

As already discussed after the synthetic experiments, *Borda Social Choice* is the more convenient approach, as it can handle use cases that focus on a higher dimensional feature space superiorly. Considering the *internal clustering indicators*, the proposed approach confirms a high quality. That means that *Borda++* achieves a high intra-cluster similarity, even between Eucl. and Canb. Note that the distance calculations evaluate the quality indicators Silhouette and the Davies-Bouldin-Index using the *Euclidean distance* and the *Canberra distance*. Therefore, it is evident that Eucl.

gets a better quality by using the Euclidean distance and Canb. is better using the Canberra distance. However, the *Borda++*-based allocation lies between both quality measures and provides high internal quality.

Finally, *Borda++* is a competitive alternative for cluster allocation in centroid-based clustering algorithms like k-means due to similar *runtimes*, fewer *iterations*, and the benefit that no *normalization* is needed before the clustering. Additionally, comprehensive experiments considering the internal quality emphasize the advantages of the alternative clustering approach.

3.5.2 Use Cases

As mentioned, a *demo recommender* exploiting the *Borda Clustering* to avoid domain normalization, based on the IMDb, was introduced in [KRE19]. This recommender effectively caters to user preferences by utilizing preference-based queries to identify suitable movies based on key features such as running time, genres, actors, rating, and release year. Considering Example 2 again, preferences are given, which can be seen in Fig. 3.8. Clustering those results on the *Pareto-frontier* delivers comparable results between two clustering scenarios, which also get evaluated using quality metrics to find the most appropriate number of clusters. A possible result comparing the *Borda++* as well as standard k-means++ with Euclidean distance can be seen in Fig. 3.9[11].

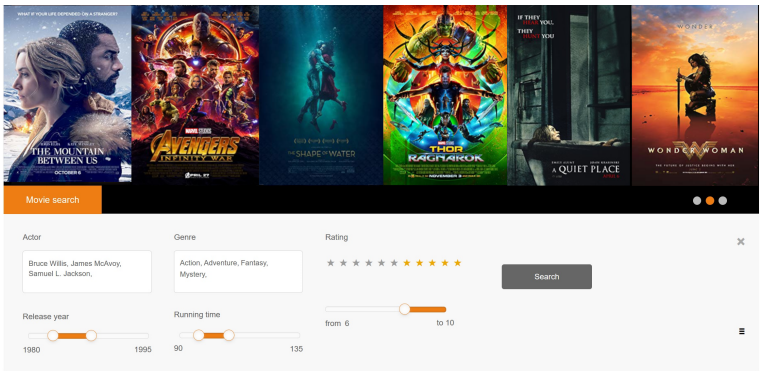


Figure 3.8: Preference-based movie selection in a recommender system.

[11]Images of the Demo Recommender contain contents of the OMDb API (http://www.omdbapi.com/), which are licensed under CC BY-NC 4.0. (https://creativecommons.org/licenses/by-nc/4.0/)

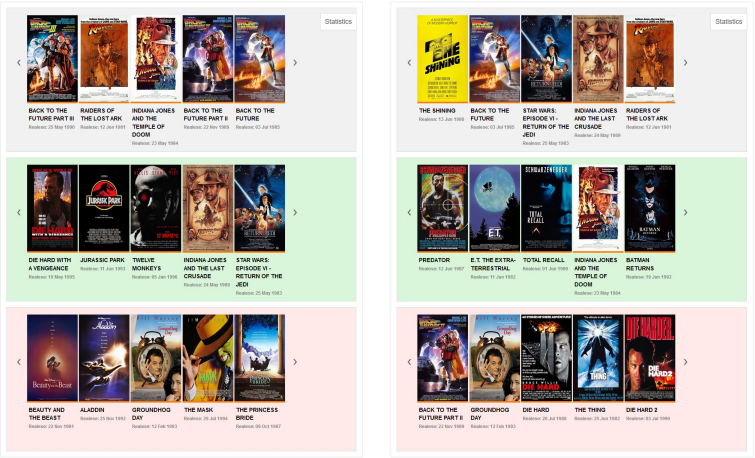


Figure 3.9: Compare mode of the cluster results in movie recommender.

Another scenario presented in [EKR18] is clustering *Pareto-optimal* objects in *data streams*. As stream data analysis is a prevalent topic both in research and the economy, social media services such as Twitter provide an API for analysis. As data streams often provide a vast data set when considering, e.g., sports events or political elections, but also other areas such as network monitoring, infrastructure manufacturing, or meteorological observations, preference-based stream processing provides many benefits when learning from data streams. The approach combines a novel preference-based query processing framework with the *Borda Clustering* presented in this chapter. This approach shows that the *Borda Clustering* approach can deal with various data sets from several data streams, such as Twitter, as both parts of the framework can work with manifold and varying data sets based on stream data.

3.6 Related Work

Since clustering is a prevalent topic in Data Mining (DM) and ML, there are several other approaches considering clustering and *Pareto-dominance*, which were published years before the proposed approaches. Those and the development in recent years will be recapped in this section and delimited to the approaches presented in this chapter.

A very early approach considering a Pareto-efficient clustering was published in [FB92] where multicriteria objects for clustering were consulted, where only a few, but at most, the probably best results appear. They focus on a modified relocation algorithm

and a modified agglomerative algorithm, finding a Pareto-dominant clustering that dominates all other clusterings. In [Hua+11], a k-means clustering-based technique was published, where a so-called SkyClustering method works within a Skyline-computation in SQL on a relational database to compress a large *Pareto-optimal* set of objects to explore the diversity of a Skyline. [TB15] presents a supervised alternative clusterings approach for handling Skylines by finding clusterings of good quality starting from given negative clusterings, which should be as different as possible and, simultaneously, a *Pareto-optimal* solution. Ebrahimi et al. [ESA12] provide a Pareto-dominant clustering approach using a semi-supervised clustering where user-defined constraints are considered. In contrast to this research, which focuses on integrating finding *Pareto-frontiers* while the clustering process, both the *Pareto-dominance* and the *Borda Social Choice* voting rule are included in the clustering approach, which uses a specific criterion to allocate an object to a specific cluster. Both approaches separate the process of building a *Pareto-frontier* and analyzing them, as the traceability is much better. Another approach focusing on supervised learning combines *Pareto-optimal* clusters as presented in [MMB09]. The authors use a fuzzy clustering-based approach to yield a *Pareto-frontier* for a given data set. This set is used as training data in a classification-based approach using Support Vector machines to allocate the remaining data objects that are not Pareto-optimal. In contrast to this work, the proposed approaches do not need supervision and focus on use cases where mostly only *Pareto-optimal* objects must be clustered. Regarding specific use cases such as Recommender Systems, it is also challenging to use Semi-Supervised Learning (SSL) or Supervised Learning (SL), as it is needed to provide tailored knowledge for each use case individually.

Much research is also considering extending the primary k-means clustering, as the authors of [Kan+02] extended k-means and published an implementation, which filters the data set with a kd-tree to ensure a better separation between clusters. Considering multi-dimensional data sets are a general research area, hence [WWP88] published an approach using hyperboxes for partitioning and forming clusters to reach fewer errors compared to the ordinary k-means clustering, especially in higher dimensions. Zhang et al. ([ZZX08]) published an approach that ensures the stability of k-means clustering by adding a heuristic for finding optimal centroids during the cluster allocation. In contrast, [JNH07] are using weighting for identifying subsets in k-means reaches better results. Both [KA09] and [Ukk11] deal with chains as input for clustering algorithms and, therefore, present solutions for the cluster allocation of chains using orders instead of trivial distances. All of those approaches focus on extending and adjusting k-means clustering, but none of those works considers focusing on avoiding the normalization and standardization process. In contrast to their work, the approaches presented in this chapter deal with an ordering process in the cluster allocation to improve quality and benchmarking compared to the traditional k-means clustering.

Also, weighting and selection of feature dimensions play a significant role in research, as large sets of *Pareto-dominant* objects are avoided, such as presented in [Cha+06] where only a few dimensions k are considered for the *Skyline* computation. This approach attaches weight to less but more critical dimensions of objects. In [PHL04], Subspace Clustering is considered to mask out dimensions in high dimensional data by a so-called feature selection, which reduces the dimensions by removing irrelevant and redundant ones, reaching overlapping clusterings in subspaces. Gong et al. published a collaborative filtering recommendation algorithm in [Wei+12], which considers clustering approaches for user and item clustering using similar ratings. Clustering personalized music recommendations by setting favored music as centroids for the clustering process is published in [Kim+07]. In contrast to their work, the approach presented in this chapter exploits *Borda Social Choice* to weight the distances to clusters in each dimension according to the distances for the cluster allocation for each object. Furthermore, especially in the *Borda Social Choice* approach, no reduction of feature dimensions or any filtering is needed, as the approach can cope easily with various and diverse feature domains.

The work of Virmani et al. [VSM15] is worth discussing when examining normalization. They proposed a k-means clustering approach, where a feature normalization is integrated before the clustering process starts. Weights are assigned to each attribute value to achieve a standardization. Also, Mohamad and Usman [MU13] discuss the effects of domain range standardization. They found that selecting a specific standardization procedure according to the data set is crucial to obtaining better-quality results. In contrast to their work, no standardization or normalization is needed in the proposed approaches before the clustering step. However, a kind of normalization is performed in each step of the cluster allocation. In all these cases, the question is how to find the proper weights and which standardization procedure should be applied. Both approaches in this chapter subdue the normalization and standardization problem by applying the *Pareto-dominance* and the *Borda Social Choice Voting Rule* to allocate objects to clusters.

A recent approach to finding *Pareto-optimal* sets using a dual clustering approach is presented in [Lin+21]. The authors propose a local density-based approach similar to DBSCAN to find neighborhoods of local *Pareto-optimal* sets. Afterward, the non-dominated objects are selected from the sets and clustered again using hierarchical agglomerative clustering to reach a global *Pareto-optimal* set. A similar approach is presented in [Li+22], where an offline and online objective reduction is proposed for objective optimization. A Gaussian Mixture Model-based clustering is used to divide a *Pareto-frontier* into several subsets, which are reduced with an offline and online objective reduction method to determine the significant objectives for each cluster. These very recent approaches show that clustering *Pareto-optimal* sets is still a relevant research topic. Several diverging clustering approaches show that *Pareto-frontiers*

can quickly be processed beyond k-means clustering prosperously. The success of the approaches is encouraging for adapting, especially the *Borda Social Choice Voting Rule*, to other clustering approaches.

3.7 Conclusion & Outlook

In this chapter, on the one hand, a novel *Pareto-dominance*-based clustering framework on *Pareto-frontiers* for two-dimensional use cases was introduced. This framework provides several reasons for using this to manage large, confusing sets of tuples with explicitly different domains. First, one can influence the clustering result by attaching a weight to a more critical dimension to cluster at least over one dimension at the appearance of Pareto-optima. Second, tuples can now be clustered over better-than relationships to avoid adjustments for utilization in different use cases. Third, benchmarks show that a Pareto-dominant clustering can be realized quickly. The quality of the proposed approach is satisfying because the stable clusters distinguish from them the basic k-means clustering but are still as similar as possible, especially regarding the affiliation of similar points w.r.t. the one-dimensional clustering. As discussed previously, the drawbacks of the *Pareto-dominance* approach occur mainly in higher dimensional feature spaces, as the allocation is stretching to its limits if *Pareto-dominant* distances considering the dimensions are occurring. One-dimensional clustering may help in a two- or three-dimensional space but is no solution in higher dimensions as the clustering results are influenced too much. Thus, the whole clustering process is sophisticated and only advisable for low-dimensional feature spaces.

On the other hand, a novel clustering framework exploiting the *Borda Social Choice Voting Rule* was introduced. Especially in high-dimensional applications, the proposed framework handles large and dizzying sets of objects. The users do not need to care about normalizing the domain ranges because *Borda Social Choice* for the cluster allocation consults each dimension equally by weighting the distances to the clusters. The experiments show that the approach terminates in comparative runtime to k-means clustering with traditional distance measures but needs fewer iterations until a stable clustering is reached. Furthermore, comprehensive quality experiments verify the benefit of the proposed approach in the context of a large and multi-dimensional environment, namely the IMDb movie recommender from [KRE19]. The *Borda Clustering* framework can manage various multi-dimensional preference-based use cases with diverse domains, e.g., movie search, hotel booking, or car purchasing.

However, the *Borda Social Choice* clustering approach has some drawbacks. Even though this approach does not need normalization, hyper-standardization is possible, as the distances in each dimension are sorted by their closeness to the centroids. Assuming there are a lot of very close objects and some objects that have more

considerable distances in one dimension, the effect of allocating weights may lead to standardized values independent of their closeness to the centroid.

Another worthwhile discussion considering centroids is using actual data objects as centroids for both approaches. This approach's benefit is that *Pareto-dominant* objects are always considered centroids, and finally, for each cluster, a representative is given after termination. This approach may work well, as k-Medoids is an established approach. However, the fewer data objects are given, the clusters may be shifted compared to the basic k-means clustering if the distances between the cluster centroid and the surrounding data objects are large. Both approaches in this chapter were evaluated similarly to k-Medoids but did not deliver significant improvements considering quality and benchmarking results. Nevertheless, using representatives as cluster centroids in each step may be a possible solution for larger data sets.

In summary, both approaches presented in this chapter are promising because of the satisfying results of the experiments. Since *Pareto-dominance* works well for clustering objects in smaller dimensions, *Borda Social Choice* is more convenient for use cases with at least three dimensions. However, the synthetic experiments show that both approaches are competitive to basic k-means clustering w.r.t. running time and number of needed iterations until termination. However, more is needed to prove the quality of large data sets. As k-means clustering tends to find local minima, it hardly handles noise, and thus, centroids can be shifted a lot; both proposed approaches may have problems in massive data sets.

For future work, minimizing the empty cluster problem by choosing a better initial partition, e.g., populating the centroids initially with the most preferred movies of the users, could be a very appealing approach. Furthermore, since the *Borda* clustering approach provides concise results for the Borda winners at the cluster assignment, investigating weighting dimensions by user preferences is worthwhile. In addition, the integration of the *Borda Social Choice Voting Rule* into other clustering techniques like X-Means, EM-Clustering, Hierarchical agglomerative, or a density-based clustering algorithm like DBSCAN to identify the behavior of *Borda* are encouraging projects, as k-means is not always the best clustering solution.

Chapter 4

Structure Discovery of Fine-Grained User Roles in Social Media

Und ich gebe dir einen Namen
Damit du weißt, dass du wirklich existierst
Und wir bauen uns eine neue Welt
Bewohnt von Geschöpfen
Die du Monster nennst

Callejon - *Mary Shelley*

This chapter introduces the novel Knowledge Discovery (KD) based approach step by step, which was introduced previously in Section 1.2 and substantiates it with definitions, metrics, and algorithms introduced in Chapter 2. Before presenting several Sampling strategies, Clustering techniques, and Cluster Analysis metrics, a recap including motivation and contributions of the methodology is given. Further steps, such as manual class labeling, and classification, are introduced before the probabilistic combination of the novel *Multi-Sampling and Combination Strategy* is proposed. These steps of the general KD approach were already published in peer-reviewed papers of the author [KF21; KF23].

4.1 Motivation & Contributions

Based on the general contributions from Section 1.3 and the assignment of them to areas of machine learning in Section 1.4, they will be brought into line with the general approach, which was already introduced in Section 1.2. Also, the proposed approach will be recapped and presented in more detail, focusing on the crucial steps, substantiating with essential background knowledge from Section 2. Moreover, robust algorithms and techniques will additionally be introduced in this section to augment the general background. In particular, the discovery and explainability of fine-grained structures, such as fine-grained user roles, will be performed with the approach introduced in the following chapter. In Chapters 5 and 6, the suitability of the approach's application will be investigated and evaluated on two diverse use cases, i.e., Twitter data sets as a representative of traditional social media services and a Telegram data set representing a more contrary use case of an instant messaging service. Moreover, parts of this approach will also be applied to an entirely different use case in Chapter 7 where clustered shapes of information cascades will be evaluated.

Pointing to the first contribution from Section 1.3, dealing with providing a framework for fine-grained structural analysis for data sets from social media, it is essential to learn and understand the structure of user roles and groups. Remembering the definition of a user role from Section 2.1.2 and the related work detecting them from Section 2.1.2, the focus is often only on finding coarse-grained structures and roles. The main goal of the approach presented in this chapter is dealing with a more fine-grained structural analysis, as the following example motivates the benefits.

Example 5
Assuming a Twitter data set consisting of several user messages such as *tweets*, *retweets*, and *comments* considering a sports event such as the *Olympic Games 2012* over several weeks. Some users are interested in the whole event and tend to comment on each competition. Moreover, users who interact with others but are only interested in a few sports are recognizable when looking through the data set. So many various features characterize users and their behavior, which leads to grouping similar users. Distinctions are observable when looking into a coarse-grained group of users, such as less active users. Some users only consume other users' content while creating their own content is hardly recognizable, but also users who restrain from sharing and forwarding other users' content. In contrast, the amount of creating their own content is less, too. These observations create an impression of the existence of fine-grained user roles as a kind of refinement of coarse-grained structures from related work.

Finding fine-grained user roles is a very challenging problem, as multiple Machine Learning (ML) approaches need to be combined appropriately from the KD approach presented in Fig. 1.1. Starting with clustering providing an unlabeled classification

tree for fine-grained structural analysis, finding the sweet spots of clusterings in cluster analysis is an established area in Unsupervised Learning (USL). The approach presented in this thesis concentrates on a novel *Multi-Sampling and Combination Strategy* to ensure a kind of probabilistic hierarchical clustering. It provides an effect size-based cluster analysis, finding fine-grained structures by identifying significant feature deviations in a depth-first search. This approach finally leads to finding stable groups of similar users who distinguish well from other groups and can also be found in other data sets.

Starting with the first contribution of providing a framework for fine-grained structural analyses, adapting ML techniques with less effort is crucial to saving resources and human interventions from a stable and elaborated framework. Recognizing patterns and structures considering user groups and roles is easily adaptable and transferable over various known and unknown data sets stemming from the same source. Moreover, topical deviations, as well as time variations, can be handled with little human effort. Not only the application of single steps such as preprocessing, sampling, clustering, and cluster analysis, as well as building and applying a classifier to (entirely) new data sets is valuable in terms of reducing human effort and expertise but also the portability of the comprehensive approach, plays a significant part in this work.

Human intervention is minimized by considering the second contribution of adapting the proposed approach to a variety of new unknown data sets. Clustering only needs little human intervention. In contrast, feature engineering and cluster analysis need more attention, as features w.r.t. their selection in Feature Engineering and the significance of features in cluster analysis are iterative steps with some tuneables. The need for human intervention is also cut short in the Classification step. However, building entirely new training data and a ground truth needs time. Regardless, once a pool of training data and ground truth is available, clustered samples of new data sets stemming from the same source can benefit from carefully built training data. Also, temporally and topically distinct data sets can be classified, as training data can be reused or combined with other training data sets. Moreover, existing training data sets can be used to build an entirely new training data set for those kinds of data sets that deviate significantly from the existing training data sets as part of a validation process. In the best case, only little amendments must be made, such as enriching existing training data sets.

Finally, this approach ensures that fine-grained user roles can be detected in aspects of certainty and stability independent of the size of the data sets. Small and massive data sets can be sampled and combined easily after clustering, cluster analysis, and probabilistic classification.

4.2 Methodology

In Section 1.2, a brief and compact overview of the methodology of this chapter was already given to ensure a comprehensible motivation of the proposed approach in terms of a classic KD process. In this section, the main steps of the approach will be introduced more precisely, introducing and emphasizing both the steps and their interaction to reveal the novelty and benefits as well as answering the questions from the contributions in Section 1.3. Considering Fig. 1.1 again, the approach consists of 5 main steps, namely *Feature Engineering*, *Sampling*, *Clustering*, and *Cluster Analysis*, as well as *Classification* and *Combination*, which will be compared to a more high-level overview of the model in Fig. 4.1.

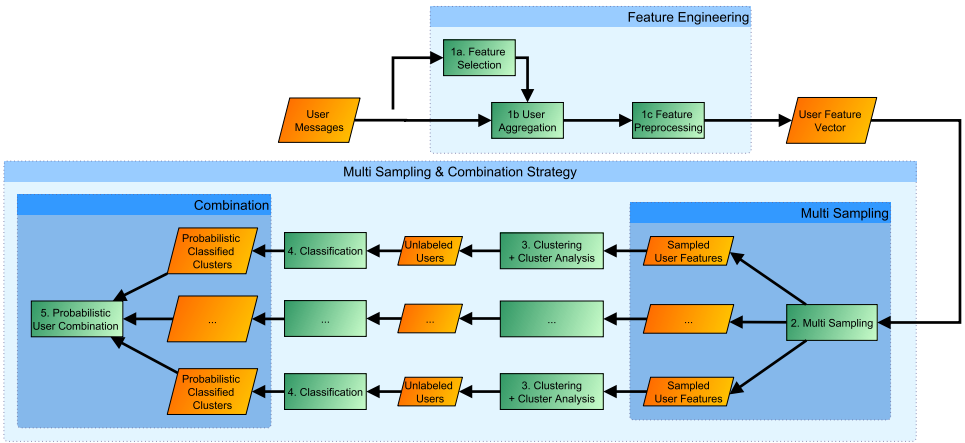


Figure 4.1: Detailed flowchart of the KD approach.

Starting with *user messages* as a *Raw Data Set*, that was recorded from a social media service, e.g., Twitter, in the first step *Feature Engineering* of a KD process (cp. Fig. 1.1) relevant features to capture various properties of users are determined and processed. In this approach, *Feature Engineering* is capturing both the *Feature Selection*, as well as *Feature Preprocessing*, in Fig. 4.1. However, before features can be processed, the raw user messages need to be *aggregated* for each user over the whole time captured by the data set to build a *feature vector*, representing several user-based features as well as *aggregated features* from the *raw user messages*, as the basis for further analysis in the KD approach. So, unique feature characteristics, such as total messages or number of replies, can be derived from the *raw messages* for each user.

The main focus is on features that rely on widely available data and are easy to compute even for large-scale data w.r.t. complexity and runtime. Moreover, this work concentrates only on individual user messages and not complex structures between users, avoiding the need for an entire social graph. Feature Selection is only essential if entirely new and unexplored data sets stemming from social media services such as Twitter or messaging groups from instant messaging services such as Telegram are considered. Beyond that, this step can be skipped if data sets are already explored, and knowledge of suitable features is given. Features should be selected according to their manifoldness, as features that are too similar tend to correlate too much. Narrowing the number of chosen features is often an iterative step, as the impact of the correlation coefficient presented in Section 2.2.1 sometimes can only be evaluated after the clustering step. Thus, providing a fully automatic approach is challenging, especially for distinct social media data sets, as the features have to be carefully chosen and examined each time. In turn, only minor adaptions are needed for data sets from the same social media.

After choosing the relevant features, they need to be *preprocessed* to suit the requirements of the following clustering and classification methods, including but not necessarily limited to *outlier removal*, *corrupt data*, and *normalization*, as well as *standardization*. Pointing to Chapter 2 again, there are several effective techniques in the process of *preprocessing* (cp. Section 2.4), such as data *standardization* aiding to balancing the asymmetric distribution of a data set, caused e.g., by outliers, while data normalization helps to adjust the bounds of all features equally and to reach a more symmetric distribution around the mean value, without affecting standing out patterns too much. As most clustering techniques work with traditional distance measures, *preprocessing* is an inevitable step to reaching good clusterings because of the presence of different deviations and skewness in each data set. In Summary, *Feature Engineering* is one of the most essential steps in the proposed approach, as it significantly affects the following steps.

The challenge of achieving scalability, dependability, and explainability for fine-grained roles is confidently addressed by the proposed *Multi-Sampling and Combination Strategy*. While sampling is not a traditional step in KD, it has become more and more essential in typical KD approaches, such as in Fig. 1.1 in the last ten years as the size of data sets became larger. The aspects of *Sampling* were also introduced and discussed in [FPSS96] and more consolidated in [ZAL14], as sampling, approximation, and parallel processing are inevitable for handling massive data sets. Pointing again to this thesis works' aspects, even large data sets can be processed precisely and flexibly using sampling, as costly clustering methods such as hierarchical agglomerative clustering are performed on each sample separately. By gradually expanding the coverage of representative samples with a controllable overlap, the analysis can be turned from an overall discovery of the general role structure in the data set to a complete

assignment of all users to roles. Moreover, tuning this strategy is possible by evaluating several sampling strategies against each other. Finally, the novel *Multi-Sampling and Combination Strategy* underpins the detection of fine-grained user roles.

The most significant benefit of the proposed approach is facilitating a hierarchical *Clustering* with fixed cluster assignments (cp. Section 2.6.1.1) creating sets of explainable candidates for user roles in the hierarchy aiding a probabilistic user-role assignment when combining several clustered and classified samples. In contrast to typical KD steps, the following cluster analysis deals with finding a possibly best clustering by exploiting an entirely new strategy. While typical cluster analysis deals with analyzing internal structures of clusters, this new approach can cope better with finding well-separated clusters by comparing feature variances in a depth-first-search, compared to traditional internal cluster analysis methods from Section 2.6.2.

In this approach, the *Cluster Analysis* is followed, correspondingly in traditional KD pipelines, by the *Classification* step (cp. Section 2.7), providing user-role probabilities for each cluster. While in the beginning of building a classifier, manual class labeling is vital, in the further steps, Active Learning (AL) and Semi-Supervised Learning (SSL) are worthwhile strategies to build and enrich training data sets with only less human supervision. The benefits of an automatic classification prevail, as trained classifiers can be used for all data sets stemming from the same source the classifier got trained for. For testing purposes, a ground truth for clustered labels has to be found manually, which deals with creating training data and evaluating the training data against the ground truth.

After classifying the clusterings, the competing labels of individual users are *combined* to produce a *Probabilistic Role Assignment*. This strategy advances a clear recognition of the core users of clusters, which got the same role assignment in all samples they are occurring, and users who receive different labels in the samples they got covered, leading to unstable user roles. Since some users do not get covered by the novel *Multi-Sampling and Combination Strategy* or occur only once when combining the samples, several sampling strategies can solve these problems as the coverage is maximized, which will be introduced later in this chapter.

This *Multi-Sampling and Combination Strategy* is a very substantive approach, enriching the hierarchical agglomerative clustering with aspects of probabilities, as several clustered and classified samples are combined by averaging the probabilistic classification vector per user in the last step *Probabilistic User Combination*. Thus, the benefits of traceability from hierarchical clustering and probabilities from Density-based clusterings are combined and present in this approach, as uncertainties of user role allocation can be captured on the fringe between groups.

The *Probabilistic User Combination* is the last step of the novel *Multi-Sampling and Combination Strategy*, laying the foundation for analyzing user roles within single

data sets and the evolution of user roles over time. The significant benefit of reaching fine-grained user roles in this combination strategy arises from the combination of reasonably chosen samples, the explainability of hierarchically clustered and thoroughly analyzed groups of user roles, and the probabilistic classification of the clusters. The further sections of this chapter deal with the specification of the individual steps of the proposed approach, while the analysis of specific data sets will be performed in Chapters 5, 6, and 7 for the specific use cases.

4.3 Sampling

As already mentioned in Section 4.2, sampling plays an essential role in this approach as part of the novel *Multi-Sampling and Combination Strategy* and thus is a very central contribution to this work, as massive data sets can easily be handled. Nevertheless, sampling does not come along without issues, as complete coverage and a suitable overlap must be found. Thus, different sampling strategies, which will be presented in the following section, were evaluated and compared on different data sets. Furthermore, *representativity* plays an essential role in all sampling strategies, as samples need to meet the expectations of the whole data sets considering features and a close deviation of representatives from similar users [D'E14]. The *representativity* of a sample can be proved with several *statistical measures* such as the *pooled Cohen's d* (cp. Section. 2.2.2), which comprises both means and the standard deviation in the calculation.

In particular, the *Multi-Sampling and Combination Strategy* aids in investigating both the change of distribution of user roles for an increasing number of samples and the certainty and stability of user roles, covering users more than once, generally aiding a full coverage of the data sets. In addition, the effects of oversampling and weak probability support will also be investigated to prove the suitability of several sampling strategies. The main goal of sampling is to obtain more structured samples and thus reach a better coverage and a sufficient overlap of users, ensuring more stability, as data points can be selected multiply. The strategies presented in the following section will be characterized briefly by balancing the pros and cons w.r.t. their costs, representativity, selection, and influence of drawn objects, and straightforward feasibility.

Most well-known sampling strategies are based on a random seed, which is the input and delivers completely random or pseudo-random generated subsets of a data set. The following section presents several strategies, such as wholly randomized techniques and approaches that partition the input before samples are built. The advantages and drawbacks of the strategies can be seen in Table 4.1 and will be presented and discussed more in detail in the following sections, w.r.t. costs for implementing and running the strategies but also aspects of representativity and verifiability.

Table 4.1: Overview on sampling strategies.

Advantages	Drawbacks
Random Sampling	
Easy to implement	Only first random is selected
Low costs	No influence on drawn data objects
Representative samples	
Linear Sample Expansion	
Low costs	Hardly related work available
Easy to implement	No library implementations
Simple to draw	Increased sample sizes
Easy to verify	
Systematic Random Sampling	
Moderate costs	Only first random is selected
High internal/external validity	Missable important characteristics
Simple to draw	
Easy to verify	
Stratified Random Sampling	
All groups represented	Need of accurate stratum proportion information
Assumptions about strata characteristics possible	Possible indivisibility of strata into proportional sizes
Better representativity than Systematic	Divisibility of population
Easy to verify	Higher preparation costs of strata lists
Quota Sampling	
Quicker & easier to conduct than Stratified	Harder calculation of sampling error
Easy exploration of distinctions in subgroups	Possible sampling bias
More variations compared to strata sampling	Misrepresent. of populations for non-suitable groups

4.3.1 Random Sampling

For testing purposes in an early stage of the development of the approach, *random sampling* is a fast strategy to get a *representative* subset of the original data set. This strategy is suitable as the effect size (cp. 2.2.2) shows only minimal effects for all given features between subsets and data sets for random sampling. For this approach, the implementation of Python is used, which draws upon [Dow]. Random sampling is especially suitable if a *representative assertion* of a data set is needed, as there is less effort in creating some samples. A minor drawback of the random sampling strategy is that the analyst does not influence how often users are represented when combining the samples again after classification and thus can hardly influence the coverage of the whole data set. Moreover, it is also quite challenging to reach complete coverage using random sampling. *Random sampling* is deterministic, using the output of the first random as input for the following number to draw. A seed initiates the process, generating a pseudo-random result list.

4.3.2 Linear Sample Expansion

This sampling strategy is not a basic sampling strategy from the literature but a specifically developed strategy to increase the coverage of all data points to 100 %. As predominantly random sampling leads to many overlapping data points when merging a set of randomly sampled data points, this strategy is a suitable starting point if the coverage decreases when adding new samples. Thus, a saturation of distinct data points arises. This strategy (cp. Fig. 4.2) can be defined in three steps as follows:

Definition 29 (Linear Sample Expansion)

1.) For a given dataset D consisting of d_i data points, a set of samples S, consisting of s_i samples, is created using, e.g., the random sampling strategy.

2.) Create a distinct set of data points to find which data points from the whole data set are already covered and which are not.

3.) Distribute the remaining data points evenly over all samples.

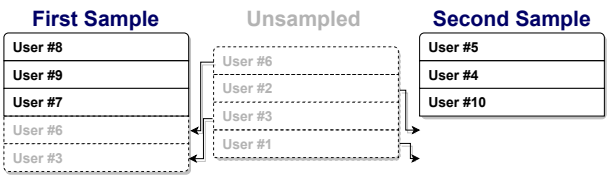


Figure 4.2: Methodology of Linear Sample Expansion from [Mac23].

Linear Sample Expansion has low costs because of its linear complexity, is easy to implement, simple to draw, and easy to verify in terms of representativity. On the other hand, this strategy is difficult to compare to approaches as it is specifically developed, is not included in libraries, and significantly increases the sample size. The latter aspect increases runtime and memory in clustering, as hierarchical clustering approaches are more complex. Suppose this sampling strategy is used based on a suitable amount of samples stemming from the random sampling strategy. In that case, the new samples can easily be verified regarding representativeness, as the samples do not change too much [Ach+13]. Like the Random Sampling strategy, this sampling strategy fulfilled a valuable degree of representativity as the pooled Cohen's d showed fewer effects over all features in the tested data sets. Thus, this strategy delivered a satisfying quality but needs at least a medium degree of drawn samples to reach an adequate overlap and a good coverage.

4.3.3 Systematic Random Sampling

While the first two strategies are based on kinds of random sampling and thus have a high degree of *representativity*, *Systematic Random Sampling* introduced in [Dan11; Mah+20; Ber20] only has a *random seed*, which is used to shuffle the data set as can be seen in Fig. 4.3. This shuffled data set is now used to draw the data points. Before the sampling process can start, one has to choose how many samples are desired because the whole data set is partitioned equally so that each sample has the same amount of data points. Considering n samples, each $n-th$ element is allocated to the same sample starting from the beginning. Finally, the whole data set is partitioned so that each user is allocated exactly once to a sample. If multiple coverages of data points are needed, this process can be repeated with another seed, leading to a different shuffling of the whole data set. Partitioning the data sets is relatively straightforward and reaches higher coverages than the original. However, there are also two anomalies, as there are only a limited number of other samples after drawing a sample without shuffling the data set. The second anomaly arises because each data object is assigned to precisely one sample, creating an extreme test case.

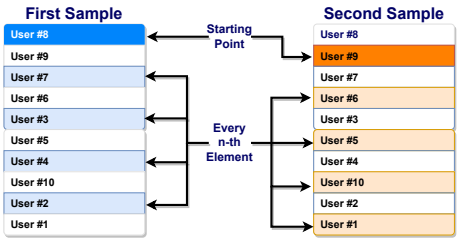


Figure 4.3: Methodology of Systematic Random Sampling from [Mac23].

The benefit of this strategy is that full coverage can be reached easily as the complexity of this sampling strategy is straightforward in terms of memory and runtime, and only one run is needed for partitioning. Moreover, this strategy has a high internal and external validity; samples are easy to draw and verify regarding representativity. A drawback of this strategy is the missing randomness, as only the first data point is chosen by a random seed. Thus, the representativity may not be given as in the strategies before.

4.3.4 Stratified Random Sampling

Compared to basic and systematic random sampling, a more algorithmic-driven sampling approach is *Stratified Random Sampling*, visualized in Fig. 4.4 and presented first in the work of [Dan11; Ber20], which aims at grouping the whole data set into n groups, each for one of the n *samples*, the so-called *strata*. From each *stratum*, randomly chosen data points are allocated to the same sample each time. One of the main goals of improving sampling strategies is the increase of covered users; this strategy can cope better with this problem as users will be removed from the strata once they get allocated to a sample. Moreover, the essential *Stratified Random Sampling* will be adjusted to create more samples; the strata should stay dynamic. So, after creating a sample, the strata will be updated after each drawn sample.

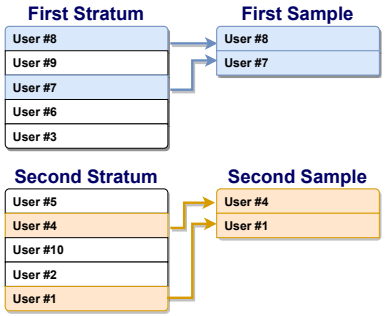


Figure 4.4: Methodology of Stratified Random Sampling from [Mac23]..

Stratified Random Sampling is not just a partitioning of the whole data set, as randomness is given because the objects allocated from the stratum to the samples are drawn with the same probability within each stratum. Compared to systematic random sampling, on the one hand, the probability of reaching 100 percent coverage is inferior but possible when choosing suitable strata. On the other hand, *Stratified Random Sampling* is not as extreme as Quota Sampling, which will be introduced next. The advantages of this approach are that more representative samples are drawn, this strategy is easy to verify considering representativity, and there is knowledge about the characteristics of the strata. Drawbacks of this strategy include the need for accurate information to reach balanced samples, the possibility that the original data set is easily divisible, and the higher costs for the groups' calculations.

4.3.5 Quota Sampling

A progression of the Stratified Random Sampling is *Quota Sampling* (cp. Fig. 4.5), where two categories as strata are created, from which the samples are drawn. In contrast to *Stratified Random Sampling*, a sample is created by *merging candidates* from both *strata*, a stratum with already *allocated* candidates and a stratum with candidates who have *not already been allocated* to a sample, which helps to create a coverage of 100 percent. While the categories can easily be adjusted after a sample is drawn, the number of objects per sample from each of the two groups is predefined and helps reach complete coverage. Thus, finding a suitable percentage of the two strata is essential to prevent extreme cases such as an almost non-random sampling or results that simulate an utterly random sampling.

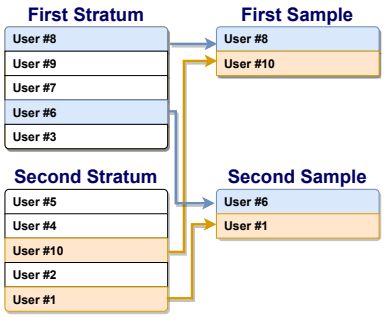


Figure 4.5: Methodology of Quota Sampling from [Mac23].

As mentioned, *Quota* has much in common with *Stratified Random sampling*. However, it defines itself with the possibility to draw from 2 different strata, which enables a variable strategy depending on sample sizes and specifications of data sets. Moreover, quota sampling has lower complexity, and subgroups can easily be explored due to distinctions. Drawbacks of *Quota* results because of their non-probability property, as possible sampling errors, are incalculable since they can only be detected for probability-based sampling strategies. Verifying samples created with the *Quota* strategy is thus possible by evaluating the portions of the whole data set due to their representativity with metrics or with additional statistics, which will be introduced later when finding suitable percentages for the specific use cases. Moreover, sampling biases and misrepresentations can also occur. Regardless, it can easily be avoided by adjusting the percentage for the different groups, which makes quota a worthwhile strategy due to their flexibility [Mos52; YB14].

4.3.6 Linear Cluster Expansion

Since some of the strategies from this section can hardly reach complete coverage as combined samples reach saturation at a specific point, where only a few new objects are covered, the *Linear Cluster Expansion* is a strategy to cover the non-allocated objects. This is not a sampling strategy and has nothing in common with the *Linear Sample Expansion*. Nevertheless, it is a helpful supplement as it facilitates that, combined with the previously introduced strategies, *coverage* of 100 percent can be reached easily. So, it is possible to detect the *saturation* points of each sampling strategy after combination and then start the *Linear Cluster Expansion*, where non-allocated users are assigned to user roles. For this assignment, the user roles are built upon the users from the classified clusters due to their majority. No clustering and classification is needed for each non-allocated user, as only the distances to representatives of each user role are calculated. This strategy has a lower *complexity* and, thus, a lower runtime, as the allocation process is *linear*.

It can be compared to a k-means clustering with only one iteration, as the distances between each non-allocated object and the representative of a set of users, which belong to a user role, act as an allocation criterion. Strategies that work with representatives for clustering, such as k-means clustering, as well as other clustering techniques forcing a point-assignment strategy like CURE, which exploits aspects of hierarchical and point-based clustering or BFR, an extended version of k-means were anticipated but tend to distort the results of the hierarchical clustering and make the aspects of traceability void. This strategy is novel and thus cannot be found in related work or literature; therefore, several hypotheses will be tested later in Section 5.5.3.

4.4 Clustering & Cluster Analysis

The following section presents a detailed explanation of *Clustering* and *Cluster Analysis*, which are the next steps of the proposed approach shown in Fig. 4.1. As previously mentioned in Section 2.6, clustering is a widely used ML technique that partitions data into distinct and well-separated clusters. The input for the clustering step are the normalized and standardized samples from the *Feature Engineering* step, with well-established and unquestionably explainable features. Even though a feature reduction using dimension-reduction techniques such as PCA and LDA would reduce the distance calculation between pairs of data objects and thus would also reduce the running time of the clustering as stated in [Pau+11], the influence will only be slightly remarkable by reducing from 12 to 4 or 5 dimensions. Furthermore, the explainability aspects in the following cluster analysis will be futile, as significant feature deviations cannot be identified clearly to outstanding features due to information loss after dimension

reduction, leading to inadequate clusterings. This explainability loss also proceeds in the following steps of building the classifiers. Thus, this work focuses solely on pure features selected and identified conscientiously in the Feature Engineering step.

Several clustering approaches, which rely on techniques presented in Section 2.6, were considered and tested with the data sets to identify the structure and (sub)-groups among the user data, such as approaches based on *partitioning* (e.g., k-means[12]), *density* (like *DBSCAN*[13], *OPTICS*[14]) and *probability distribution* (e.g., *EM*[15]). While centroid-based clustering approaches, such as k-means and variations, did not perform well on multi-dimensional data sets due to the lack of structure, and thus, clusters are built in regular shapes as the number of desired clusters has to be chosen in advance (cp. Section 2.6.1.2), distribution-based methods can handle noise more easily. However, they still face similar challenges with these data sets as discussed in Section 2.6.1.3. Moreover, partitional clustering techniques require the number of desired clusters as input in advance and tend to build clusters in regular shapes.

Density-based approaches such as *DBSCAN* and their specification *OPTICS* (cp. Section 2.6.1.4) delivered halfway beneficial results as they can distinguish noise from core points and handle a clustering without the need for a predefined number of clusters. However, hierarchical structures within clusters and different densities within a data set, leading to different sizes of clusters, cannot be handled thoroughly as in traditional hierarchical clustering techniques. *Hierarchical clustering*[16] turned out to be most suitable, as it can capture complex, irregular shapes without requiring a fixed number of clusters. Unfortunately, hierarchical clustering does not provide probabilities like distribution-based approaches. However, the proposed approach based on a *Multi-Sampling and Combination Strategy* delivers a kind of probability, as several data objects are considered more than once and may have a different classification in different samples. Moreover, the hierarchy is served as an unlabeled classification tree, the so-called dendrogram, on which feature differences explain the variations between user roles.

4.4.1 Hierarchical Agglomerative Clustering

This work evaluated several hierarchical-based clustering approaches on the same data sets, such as divisive and agglomerative clustering, along with several linkage criteria. Nevertheless, only the *Hierarchical Agglomerative Clustering* delivered relevant and comprehensive results.

[12] https://scikit-learn.org/stable/modules/clustering.html#k-means
[13] https://scikit-learn.org/stable/modules/clustering.html#dbscan
[14] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.OPTICS.html
[15] https://scikit-learn.org/stable/modules/mixture.html
[16] https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html

The *Hierarchical Agglomerative Clustering* is defined as follows [WX08]:

Definition 30 (Hierarchical Agglomerative Clustering)

1) Start with N singleton clusters and calculate the proximity matrix for N clusters.
2) Search the minimal distance $D(C_i, C_j) = \min_{l \leq m, l \leq N, m \neq l} D(C_m, C_l)$ in the proximity matrix using a distance function and combine C_i and C_j to C_{ij}.
3) Update the proximity matrix be computing the distances between C_{ij} and the other clusters.
4) Repeat steps 2 and 3 until only one cluster is remaining.

As mentioned in Definition 30, several cluster distance functions, the so-called *linkages*, can be used to perform Hierarchical Agglomerative Clustering. For this work, the most common functions were evaluated on data sets to find the most suitable. There are graph methods like *single*, *complete*, or *average linkage* and geometric methods like *centroid*, *median*, or *Ward's* linkage. *Single* linkage is defined as the smallest possible distance between 2 clusters, i.e., the distance between 2 points of each cluster, which delivers the smallest distance. This distance is also called the nearest neighbor. A drawback of this method is the chaining effect, where data points are clustered in an elongated way, which can cause clusters with blurry features because of noise in chains. Another distance measure is the *complete* linkage, which uses the farthest possible distance between two clusters in contrast to the single linkage. This technique works well in most cases, especially if data points are well separated from each other, and delivers, in most cases, small clusters. *(Weighted) group average* linkage is defined as the average distance of all data point combinations between the two clusters (divided) through the number of data points. All these linkage functions work with several distance methods, e.g., the well-known Euclidean distance.

As mentioned, there are also *geometric* methods like the *centroid* or *median* distance, where the *shortest possible distances* of the *mean* or *median* of each cluster are adduced. A more specific case is *Ward's* method, also known as the *minimum variance method*. This approach minimizes the within-class sum of squared errors between clusters, which is generally similar to k-means. In contrast to k-means clustering, where distances between all objects and cluster centroids are minimized in each iteration, leading to convex clusters, Ward's minimum variance is considered at each step when two clusters are merged, considering only two subsets of all data objects. This aspect does not prevent the generation of convex clusters. However, it diminishes the feasibility as core clusters sharing the slightest *variance* are generated first and merged with other clusters having a more significant *variance* in the last steps of the dendrogram. In most cases, the most common distance function is also the *Euclidean distance* [WX08;

GMW07]. Since all of the considered data sets are multi-dimensional, geometric linkage methods, in particular, *Ward's* worked best. The *Error Sum of Squares (ESS)* of a set of data points C is defined as in [GMW07]:

Definition 31 (Error Sum of Squares)

$$ESS(C) = \sum_{x \in C} (x - \mu(C))(x - \mu(C))^T \tag{4.1}$$

where $\mu(C)$ is defined as the mean of set C:

$$\mu(C) = \frac{1}{\mid C \mid} \sum_{x \in C} x \tag{4.2}$$

If there are k groups $C_1, C_2, \cdot, C_k$ given in one level of the clustering, the information loss is defined as the sum of ESS:

$$ESS = \sum_{i=1}^{k} ESS(C_i) \tag{4.3}$$

At each step of the *hierarchical clustering*, the union of each pair of groups is determined. Finally, the pair is chosen, whose fusion results in the minimum increase of information compared to the unfusioned clusters.

Hierarchical Agglomerative Clustering requires careful consideration of the datasets' characteristics. Depending on their characteristics, only certain *linkage* methods and *distance measures* are effective for specific datasets. In most cases, the analyst must trial and error until one or more suitable approaches are found. *Ward's* linkage was chosen for the proposed approach, as it works with a minimum increase of information. In other words, while traversing through the dendrogram of clusterings, the aspects of traceability and explainability came out best for *Ward's* linkage, as feature changes can be read off from *dendrograms* easily.

Clustering methods that follow a hierarchical approach have a few issues. Firstly, they can be computationally expensive in terms of CPU and memory usage, even for moderately large data sets, due to their $O(n^2)$ scaling. Secondly, most popular clustering approaches only support "hard" clustering, meaning a data point can only be assigned to one group. However, users may have multiple roles to varying degrees in reality, making a *soft* and *probabilistic* assignment more accurate and meaningful.

Addressing both of these issues, the *sampling/ensemble-based* approach introduced in Section 4.3 is an excellent opportunity, as reducing the sample size allows one to quickly discover the structure while drastically reducing the cost compared to clustering the whole data set. A linear cost increase is noticeable by incrementally

drawing more samples while allowing a parallel execution that provides a faithful data representation. With overlapping clustering results from several samples for the same user, a majority role or the probability for specific roles can be chosen. Likewise, the stability of the role recognition can be recognized. The number of samples becomes a tuneable, trading off the effort of computation and labeling with the coverage of users and the amount of support for the roles.

4.4.2 Cluster Analysis

A key question when identifying user groups and, thus, roles by clustering is the *actual number of such groups*. While hierarchical clustering avoids the issue of having to provide a *fixed number* of clusters beforehand as, e.g., k-means or EM require, the *classification tree* produced by the clustering, often represented as a *dendrogram*, allows for an extensive range of cluster numbers between 1 and the number of input values, as the cluster candidates are aggregated along the hierarchy.

Traditionally, this issue is tackled by computing internal quality metrics such as *Davies-Bouldin*, *Silhouette*, and *Calinski Harabasz*, introduced in Section 2.6.2. All of those metrics determine a valid point where to look inside a clustering, as they deliver for each number of clusters a value in this metric space. Those metrics did not work well, as *Ward's* linkage is based on the Error Sum of Squares, while those traditional metrics are built upon traditional distance metrics such as the Euclidean distance. Thus, the linkage of the dendrogram was also considered more in detail, as the pairwise fusion of clusters depends on distances. An up-and-coming method to find visual abstractions when considering the boxplots of the clusters is the elbow method, which relies on the acceleration of distances. Following the approach of [Zam16], which relies on the distances of the dendrogram as a metric and refining the elbow with the acceleration of these global and local distances, more valuable insights were given. This approach yielded valuable but only partially satisfactory results as generalized, coarse-grained main groups could be determined reliably, which is a starting point for more detailed analysis.

The *Elbow*, based on the *acceleration*, is defined as:

Definition 32 (Elbow)

$$\hat{k}_E = 2 + \arg\max\{d_{i-2} - 2d_{i-1} + d_i\} \mid i \in [2, N] \tag{4.4}$$

where d is a set of distances in the dendrogram starting at 0 until N-1.

The *Elbow* function delivers clusters where distances to other clusters are rather significant, i.e., they are well separated and, on the other side, also compressed. Fig. 4.6 shows the *Elbow* function (blue line) as well as the *acceleration* (orange line) of

the dendrogram in Fig. 2.3. One can see that the *acceleration* has its highest value at 4, which means that at least 4 clusters are found that have a high distance to the fusion cluster in the next merge and thus are well separated from each other.

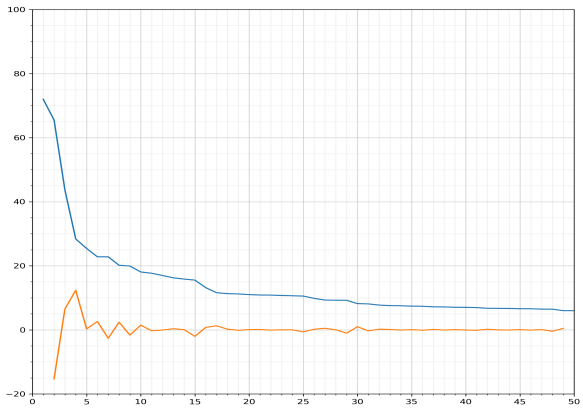


Figure 4.6: Elbow function and acceleration for dendrogram in Fig. 2.3.

This generic approach was augmented with a domain-specific methodology based on the insight that user roles often can be refined by apparent differences in specific features, not just on general, global metrics. Intuitively, comparing specific characteristics of boxplots such as the means, median, and quartiles in the manual labeling process led to determining features whose differences explain the characteristics of subgroups. To formally express and discover these differences, statistical measures were considered, as they can cope best with geometric linkages such as the considered *Ward's* linkage, which relies on variance changes. In particular, effect sizes such as *(pooled) Cohen's d* [Saw09], which was already introduced in Section 2.2.2, were used to capture significant feature deviations. Cohen's d is the difference between the means of two sets divided through the standard deviation. In contrast, the pooled standard deviation [Coh88] allows dealing with cluster candidates of different sizes, so smaller clusters with significant features can be detected reliably. Otherwise, smaller clusters representing prominent user roles tend to get absorbed by more significant clusters. Furthermore, pooling is less sensitive to feature drift.

The refinement process is modeled using a *depth-first* search covering the subtrees in the dendrogram forming the generalized roles, which can be seen in Definition 33. At a search step, the process compares in a pairwise fashion the measures for each feature of the current cluster to those of its two direct descendants, which are the refinement

candidates. This search continues as long as there are significant effect changes, leading to a possible *cutoff* for *refinement* in this particular path finding salient features. The most challenging aspect was finding a suitable *significance criterion* for the cutoff based on finding at least a predefined number of occurring effects considering *effect size* from Section 2.2.2. Both a general cutoff at the *deepest point*, where significant changes could be observed, and a distinct *non-uniform cutoff* after the *last significant change* in each step of the *depth-first search* were considered. The most suitable outcome for this strategy was reached using the *non-uniform cutoff*, as a *general cutoff* would lead to a refinement in each cluster and thus lead to blurry clusters. In contrast, the *non-uniform cutoff* reached more distinct clusters, which were also confirmed by the following classifications. Moreover, the number and the degree of considered effects for features remain tuneable, as different data sets deliver different effects and have to be adjusted.

Definition 33 (Depth-first search based effect size analysis)

1) Define *significance* by the number of *desired effects*, e.g., at least two medium or one large effect.

2) Start a recursive *depth-first search* at the root of the *dendrogram*.

3) As long as there is no *significant change* in a row of two effect size comparisons, i.e., whether there was significance or not, calculate effect size *pairwise* between the *parent node* and the *two offspring nodes* for each feature and determine the significance of the whole feature vector.

4) If a child node falls below a predefined *threshold*, the child node is returned as a final cluster.

5) Return the clusters, w.r.t their *significance changes*.

Considering how clusterings are used in the overall approach, no perfect fit for the cluster number is necessary. Instead, a slight overestimation and, thus, a specification of the number of clusters is manageable, avoiding an early cutoff that would lose possible user groups. The spurious groups will be merged during the manual class labeling or by the trained classifier, as shown in the following section. A *graphic tool* was implemented for evaluation purposes, depicting the *effect size calculations* for each pairwise feature vector in the dendrogram and the boxplots, which will be introduced in Section 5.4.2 for the Twitter use case and also utilized in Section 6.4.3 for the Telegram use case.

4.5 Manual Class Labeling

A central point in this work's approach is the *manual class labeling* of clusters, especially at the beginning of analyzing entirely new data sets. Though not traditional w.r.t. a classical KD approach, manual class labeling is significant before classification is even possible, as it catalyzes the building and verification of *training data*. As an interface between cluster analysis and classification, manual labeling also addresses *tuneables* for a further intermediate step: Cluster analysis.

The first use case is realized right after the clustering process in the early stage of analyzing an *entirely new data set*, where creating an entirely new *ground truth* is unavoidable. As understanding the structure of a dendrogram and searching for fine-grained structures starting at the top of the dendrogram is a significant contribution to this thesis, manual labeling helps to find *anchors*. By exploiting the *effect sized-based depth-first search* for cluster analysis, introduced in the previous section, these *anchor* user roles help to adjust *tuneables* for the stopping criterion. Thus, *manual class labeling* is vital to identify fine-grained user roles within the cluster hierarchy. The *graphic tool* from Fig. 5.9, mentioned in the section before, can display feature changes within the hierarchy and visualize them as boxplots to identify user roles by their deviations from parent and sibling clusters. So, a manual correction is essential to ascertain the tool's functioning.

Moreover, a second use case for *manual class labeling* arises in creating and validating *training data* as a part of an AL and SSL process, which was introduced in Section 2.5. This use case is essential when analyzing data sets from already known sources. Thus, suitable *training data* regarding *topical* and *close-in time-related* data sets were (partially) created before. In this case, the manual labeling process is only needed as a kind of catalyst to fill the *training data* set initially as part of the SSL and AL-driven building process, which was introduced in Fig. 2.2 from Section 2.5. With the aid of the training data, the model suggests *queried data* to the *human supervisor*, who can *accept* or *decline suggestions* to incrementally *enrich training data*. In terms of this thesis, further metrics such as the *pooled Cohen's d* to evaluate effects for features to the suggested class training data were considered. The first essential part of creating *training data* deals again with a manual analysis by labeling *boxplots* of clusters within a *dendrogram*. Also, deviations to the features of the entire data set and to siblings and parent clusters help find suitable user role labels from literature for each cluster manually, similar to the first use case of tuning the tool. The second part focuses on *dimensionality reduction* strategies such as *PCA* or *LDA* (cp. Section 2.8). Relevant user features are highlighted by a composition of principal components, leading to a *diminishing number of dimensions*. These insights help to validate training data at irregular intervals, as candidates for training data can be manually observed by

visualizing the cluster means in a 3D space. PCA and LDA are worthwhile strategies to identify *feature drifts* and *changes* in user roles within single *training data sets* and across data sets.

The iterative process of manually labeling classes for all mentioned use cases involves proposed *stopping heuristics* that effectively narrow down the clusters to a manageable 15-30 candidates. The analyst is guided through the dendrogram from the top, while other heuristics, such as *feature distributions* and *deviations* from the *boxplots* and the *effect sizes*, provide support throughout the manual class labeling process. Early identification of generalized user roles serves as a foundation, while distinct fine-grained user roles act as an anchor. The refinement process concludes when no further clusters can be identified distinctly, and the proper clusters are determined through the coarsening or combining clusters.

To summarize the process of manual class labeling, user roles are first identified and mapped to candidates from the literature. In further investigating user roles, the strategy also helps find entirely new user roles, which lie in between two or three established user roles. *Manual labeling* is a very tedious process as it is hardly scalable and suffers from reproducibility issues, as human assessments tend to be of a subjective nature. *Expert knowledge* is rare but unavoidable as this process initially needs a mapping from user roles from the literature to clusters, which is part of all Supervised Learning (SL) approaches such as AL or SSL. The effort of *manually labeling* clusters is rewarded with a high quality of well-described user roles, which can be used as *ground truth* and *training data* for building *classifiers*. *Manual labeling* is mostly only essential at the beginning of creating training data and a domain-specific ground truth or adjusting the *effect size-based depth-first search* cluster analysis tool and thus can be skipped once training data and ground truth are created. Thus, manual labeling economizes human intervention and time in further steps of the approach, even though it has a high human effort in all cases, it is vital as it is the starting point for a successful classification. In this work, two use cases concentrating on labeling user roles in social networks are present: Twitter users in Section 5.4.3 and Telegram users in Section 6.4.3, while in Section 7.4 a manual labeling of entirely different classes in a completely new use case is applied.

4.6 Classification

After the manual labeling process of several cluster-means, an eminent step for creating training data for *classification*, the sampled and clustered data represented as an *unlabeled hierarchical classification tree* deals as an input for the *Classification*. Even though classifying pure unclustered users would cut the clustering process short, this approach will not work, as the classifiers would produce blurry output for most

users, as many users must be represented in the training data. Moreover, choosing *representatives* of each cluster is way more suitable, which will work more suitably for training classifiers and produce better results as only whole clusters will be classified. Thus, representatives such as the *cluster mean* representing an average feature vector of all users within the cluster tended to be most suitable as input for the *Classification*.

The main goal of the proposed approach is to allocate a *multi-class label* representing user role probabilities as a probability vector to each user as part of a cluster. This step will be repeated for each sample, capturing many users for satisfying coverage and gathering users multiple times for more stability in the *Multi-Sampling and Combination Strategy*, which was introduced in Section 1.2 and will be presented explicitly in Section 4.7. As users may be present in more than one sample, each user can receive a set of probability vectors to all current user roles, which will be averaged in the combination step and prepare the foundation for the *probabilistic allocation* of users to user roles. Moreover, as the Classification step is one of the most time-consuming steps, w.r.t. *expert knowledge* and thus demands a lot of *human intervention* and effort, the focus is also on transferring knowledge from previous analysis of other data sets in terms of topical or time semblance as stated in section 1.2. Thus, not for each data set, a *distinct training data set* for the Classification step must be created, as the focus is on reusing training data for several topical or time-related data sets, cutting short the time for creating training data.

Nevertheless, when talking about classification, several preparations have to be made before the users can be classified. Independent from the different classification techniques introduced in Section 2.7, both ground truth and training data must be specified. The whole Classification process has things in common with the well-known procedures SSL and AL, which were already introduced in Section 2.5 and 4.5, as training data must be created. The AL process is a rather substantial part of building a classifier, as it can cut short the whole process and thus reduce human intervention and effort.

Since different use cases were addressed in the questions in the introduction, a distinction between complementary scenarios has to be made w.r.t. the degree of exploration of data sets, as they require different quantities of *human involvement*. Considering entirely or partly new data sets without or with little training data need further *human involvement*, while well-analyzed data sets have a well-established training data set. This distinction is emphasized in the program flow chart in Fig. 4.7, which guides this section. While in the first use case considering the Twitter data sets (Chapter 5), terms of *explainability* through the whole process had priority, and thus the whole *training data building process* was nearly *manually* driven, for the Telegram use case the mentioned AL approach was utilized to *automatize* the building of the *training data* and cut short *human involvement* and *intervention* to reduce the *effort*. The steps involved in the preprocessing of the raw dataset, which include normalization

and standardization techniques, as well as sampling and clustering of data, remain the same for both scenarios. The only difference lies in the Classification process.

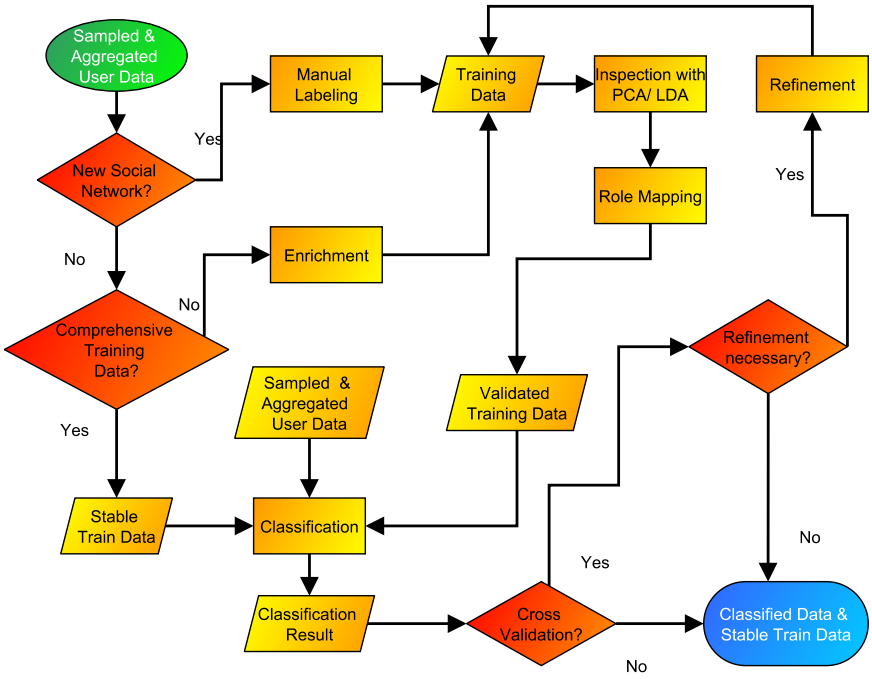


Figure 4.7: Flowchart of the classification process considering the scenarios.

1) If only data sets such as a new social network or not yet comprehensive training data are available, groups of similar users and their hierarchical relationship are discovered by the clustering and cluster analysis, thus providing candidates for user roles. The analyst will then assign role labels to these groups to manually build new *training data* or *enrich* already *available training data*. He/she is aided by *quality metrics*, *visualizations*, and *dimensionality reduction* like Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) to inspect the assigned labels. In turn, these manually provided labels form the input for a classifier that captures this knowledge and can be *cross-validated* on this data set.

2) If *sufficient training data* from the same social network with the same features are available for a *classifier*, this tedious labeling process can be cut short by providing

candidate labels for the clusters in a new data set. The existing *training data sets* and additional manual labels may be *cross-validated* to ensure the quality of the model. The analyst can evaluate these candidates within the new dataset or compare the roles across the datasets, which will be shown in a later analysis. Also, causes of mislabelling and methods to adapt them were explored.

In the beginning, with many unknown data sets, manual labeling is a very tedious and time-consuming process, as analysts have to pick out data objects with outstanding features. Nevertheless, if enough training data is available at a specific point, the Classification step can be cut short, as training data can be reused for similar data sets. That means, on the side of the spectrum, data sets that encompass a similar period of time as the training data set can be classified as the most reliable. On the other side of the spectrum, topically related data sets such as sports events also benefit from the transferability of training data on new data sets.

After classifying each cluster in each sample, and thus each given user, the *probability vectors* of distinct users across the samples can be combined into a *single probability vector* by averaging them, which will be discussed later in Section 4.7. The main reason to consider whole probability vectors was to reveal *stability* and the *certainty* of user roles. If only the user roles with the highest probabilities for each user were considered, uncertainties for users at the fringe of two or more user roles could not be captured, and valuable information would get lost, such as users with no *majority* for a user role. Moreover, variances in samples may lead to differing clusters, w.r.t the composition of clusters, which are all representative but may lead to slightly different user role allocation, as in samples, only relative user behavior can be observed. The facet of relative user behavior strengthens the need for a probabilistic classification, as accuracy for borderline users can be improved when users are grasped in several samples with differing compositions.

Pronounced issues in human labelings, such as the tedious process with limited scalability and reproducibility due to human subjectiveness, were handled by utilizing classifiers trained with several samples of one data set by composing the means of cluster feature vectors. This *training data* can determine user role labels on clusters from further data sets expressing an *n-class problem*.

After introducing the general classification process and the motivation for a *probabilistic classification*, the different classification techniques considered in the approach and already introduced in Section 2.7 are now more detailed. An appropriate classifier is chosen for each technique, helping to evaluate a broad range of different classification techniques against each other to find the most suitable for each use case. As part of the K-Nearest Neighbor (KNN) classifiers, the implementation of scikit-learn[17] is used

[17] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

together with the Ball Tree data structure, which is a fast indexing structure for finding the nearest neighbors efficiently. To also consider Support Vector Machines (SVM) classifiers, the implementation of scikit-learn[18] is used, which relies on the One-Versus-One approach. As part of the Decision Tree bases Classifiers, the implementation of XGBoost[19] is used for the Gradient Boosted Decision Trees (GBM) as well as the implementation of scikit-learn[20] for Extremely Randomized Trees (ET).

Each of them needs *training data* and *ground truth* for the *evaluation* process. At a specific point when the classifiers deliver results close to the ground truth, the training data has reached a stable point and thus can be utilized in the following classifiers. Also, the aspects of training the classifiers are very significant, as most of the classifiers have a lot of tuneable parameters. In Section 5.4.4, the initial process of building a classifier upon manually analyzed data sets from Section 5.4.3 for the Twitter data sets is described, while in Section 5.5.1 optimization steps using a grid search for finding possibly best parameter configurations for the classifiers are explained. In contrast to this initial application of classifiers, in Section 6.4.4, the knowledge gained from the insights of the Twitter data sets was applied to the Classification process of the Telegram data sets.

4.7 Multi-Sampling & Combination Strategy

This approach's central and novel aspect is the probabilistic combination of the clustered and classified users from the given samples as part of the *Multi-Sampling and Combination Strategy*. After classifying the clusters of each sample, each of the consisting users got a *probabilistic vector* consisting of probabilities for each present user role. Depending on the *sampling strategy*, users may occur more than once, which yields more stability and certainty of user roles, as an aggregation of the probability vectors of each user is affected by averaging the probabilities, as can be seen in Fig. 4.8. When analyzing the aggregated vectors, each user can get a hard label resulting from the role with the highest probability, depending on the use case and the following analysis. Since one main contribution was to guarantee a fine-grained structural and comprehensible analysis of user roles, this novel *Multi-Sampling and Combination Strategy* delivers a probabilistic allocation of user roles for each user. It ensures the certainty and stability of mapped user roles for each user.

[18] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
[19] https://www.xgboost.ai
[20] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html

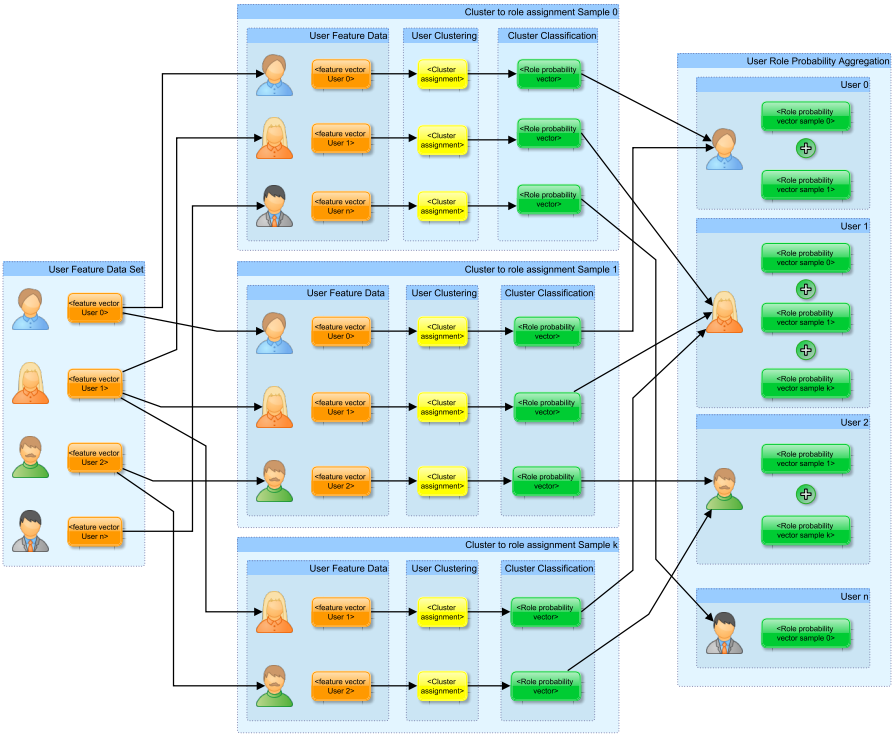


Figure 4.8: Process of Multi-Sampling & Combination Strategy.

This *probabilistic combination* of hierarchically clustered and classified users is a worthwhile alternative to distribution-based clustering (cp. Section 2.6.1.3), as each user in the data set receives a probability for each of the present user roles. While evaluating hierarchical clustering manually, only a single label to each cluster and the contained users can be mapped; the following classification enables a *probabilistic label* for each cluster as a probability vector comprising each user role. When combining distinct users from several classified clusters, with less effort, user roles can quickly be stabilized and assured by tuning the sampling strategy, the size of samples, and the number of samples, influencing the whole *Multi-Sampling and Combination Strategy*. Thus, the *stability* and *certainty* of user roles can be easily improved by clustering and classifying more samples while diminishing the sample size. This affects the stability and the certainty of user roles, but also the technical side, as reducing the sample size has a powerful impact on *runtime* and needed *memory* due to the higher

complexity of hierarchical clustering. In contrast, an enhancement of the number of samples has a linear increase. However, distribution-based clusterings are superior to hierarchical clusterings in terms of complexity, but hierarchical clusterings have the side effect of a comprehensible built hierarchy. Moreover, parallelization for hierarchical clustering of multiple samples can cut needed memory and run time short compared to distribution-based clustering. Covering most of the given user more than once in varying but representative compositions of the sampling strategy delivers different insights on the data sets, as a user may get a slightly deviant allocation, w.r.t variable picked users from the sampling strategies while clustering a whole data set only once delivers each time the same result. Finally, this approach is the starting point for further investigations, such as analysis of single or across data sets, which will be discussed later in Chapter 5.

4.8 Related Work

Clearly, identifying user roles has been one of the textbook examples of classifier algorithms, yet the application to social networks has been limited to particular aspects. Often, the studies focus on detecting specific roles or describing only a small number of coarse-grained classes. Considering the negative dynamics of many social networks, most researchers focus on identifying specific malicious users. Examples include the detection of bots [Chu+10] or spammers [Li+17], identification of aggressors in the context of cyber bullying [Cha+17; Kao+19] or, of particular interest recently, the discovery of instigators and spreaders of fake news [Shu+19; ECR20]. In contrast, the proposed approach's goal is to comprehensively assign all users to roles. Multi-role approaches such as Varol et al. [Var+14], Rocha et al. [Roc+11], and Lazaridou et al. [LNN16] limit themselves to identify a small number (often 3-5) of primary, coarse-grained groups, roughly corresponding the upper levels of the detection hierarchy in this approach. Du et al. [Du+16] provide a somewhat higher number of rules, which is still lower than the number of rules in this approach, but only gives generic descriptions. All of these previously mentioned methods are constrained to just detecting the structure by unsupervised learning such as clustering via k-means [LNN16], EM [Roc+11] or topic models [Du+16], leaving the analysis entirely to human experts. In terms of classification, Varol et al. [Var+14] entirely rely on such human expertise, using similarity matrices and handcrafted rules. In contrast, qualitative work like Tinati et al. [Tin+12] or Java et al. [Jav+07] provides a comprehensive overview of fine-grained roles and their semantics but considers only general rules on how to detect them. An alluring, complementary direction is the work on content communities/web forums, often exploring complex temporal models, e.g., [Fu19]. It should be noted that all of these works, with the exception of [Du+16] (Weibo, 12K users), [Kao+19]

(Instagram, 18K users), and [Fu19] (Stack Overflow) solely rely on Twitter due to the limited availability of data from other services. A recent work by Hacker et al. [HR21] comes closest to the approach presented in this thesis while tackling the more constrained problem of user role identification in Enterprise Social Networks. Like this work, it follows a process-based approach involving and aiding human analysts in discovering and interpreting user roles. It applies a broad set of user features and employs clustering to identify user group candidates. The authors recognize that their problem is less challenging due to the smaller scale and better observability, allowing for more expressive metrics and more well-defined and less context-dependent roles. Furthermore, a more extensive process by incorporating a classifier to perform knowledge transfer of user roles between data sets is provided, and a *Multi-Sampling and Combination Strategy* for probabilistic role assignment and better scalability is employed. While probabilistic clustering is well-established for centroid methods [DLR77] and recent work presents probabilistic density-based methods with constraints (Lasek et al. [LG19]), hierarchical clustering is not covered well regarding the probabilistic assignment.

4.9 Conclusion

The proposed approach presented more in detail in this section is based on a typical KD approach introduced in Section 1.2. The main steps examined in Section 4.2 such as *Feature Engineering*, *Sampling*, *Clustering* and *Cluster Analysis*, *Manual Class Labeling*, and the building process of the *Classifier*, were introduced explicitly in this section to lay the methodology knowledge for the initial application of the pipeline in the Twitter data sets' use case in Chapter 5 and the *transfer* to a new data sets such as the Telegram data set use case in Chapter 6. The central aspect of the proposed approach, the combination of samples to gain *stable* and *explicit* user roles, will be investigated by expedient experiments for both use cases, substantiating the *suitability* and *transferability* of the approach. Moreover, the central aspects of this approach, the *Clustering* and *Cluster Analysis*, will also be applied to an entirely new kind of data set in Chapter 7, approving the adaptability and suitability of both steps.

Chapter 5

Analyzing Fine-Grained User Roles in Twitter

Hope for the hopeless
A light in the darkness
Hope for the hopeless
You've got one life, one shot
Give it all you got

Parkway Drive - *Vice Grip*

While the general Knowledge Discovery (KD) approach was introduced in the previous chapter, this chapter addresses a use case dealing with the application and transfer of the KD approach to several Twitter data sets. Besides analyzing fine-grained structures after clustering, the creation process and transfer of training data are central aspects of this chapter. Moreover, the impacts of a plethora of sampling strategies and further tuneables are investigated on the stability, certainty, and coverage of user roles. The last aspect covered in this chapter deals with building a threshold-based transition model arising from the long-term analysis of user roles, addressing to economize time and human effort by simulating and predicting known and new data sets. Parts of this chapter were already published in peer-reviewed papers of the author [KF21; KF23].

5.1 Motivation & Contributions

The social media service Twitter is a well-known and favored platform for consuming news and exchanging attitudes on several daily life topics, e.g, political discussions, sports events, or tragic incidents. The users' structure reveals inspiring research aspects for analyzing fine-grained user roles by several aspects of user behavior such as posting *tweets*, and *retweets*, as well as influential *reactions* and *answers* on other users' *content*.

To accomplish this challenging analysis, the KD approach presented in Chapter 4 will first be applied to a single data set. The scope in this chapter addresses the whole KD pipeline from Chapter 4 applying all steps from a traditional KD pipeline as parts of the novel *Multi-Sampling and Combination Strategy* to find explainable fine-grained user roles. Once user roles are defined in one data set, the aim is to transfer the whole strategy to other related data sets stemming from the same data source to approve the suitability of a conceptual transfer and to find fine-grained user roles in an explainable way again. Twitter users develop unique communication *behaviors* that may evolve or shift over time, with or without a specific topical reason. Being a very significant topic in the present work, this approach takes another path. Addressing the evolution of users as well as user roles, a comprehensive analysis of a plethora of data sets considering shifts and drifts of user role quotas and distinct users is performed.

Analyzing related data sets over a longer time span enables the construction of whole role chains explaining migrations of users between data sets. The latter steps in this chapter are the foundation for a dynamic threshold-driven model-building approach, creating transition models for simulating and predicting user role changes from data set to data set. This process is applied and evaluated for two data set series, covering a period of ten years for each use case, aiming to investigate if this process can cut short the whole KD pipeline when analyzing additional data sets for the time series.

Finally, in this chapter, several research questions are discussed and substantiated with experiments and analyses:

- Is it possible to enable a fine-grained structural analysis framework to explore intuitively stable and precise fine-grained user roles?
- Can the framework be adapted to other related data sets stemming from the same source?
- Can the framework be adapted to entirely new data sets stemming from the same source?
- To which extent is user (role) movement beyond data sets w.r.t. feature drifts and shifts explorable?

- Can a model-building process cut short the analysis of entirely new data sets?

5.2 Background on Twitter

Before specific data sets are presented, and the KD pipeline introduced in Chapter 4 is applied to Twitter data sets, some more specific details about the social media service Twitter, enriching the general definition from Section 2.1.1, will be presented in this section.

Fig. 5.1 presents an introductory *message* called *Tweet* in the social media service Twitter. Each registered user can post tweets, which are shared with all users from the network. As mentioned previously in Section 2.1.1, Twitter was one of the first social media services to provide the possibility to *mention* users with the @ sign as well as including *hashtags* starting with the # sign enabling a *global search* of all tweets including the desired *hashtag*. Moreover, in Fig. 5.1, a snippet of *replies* is shown below the original tweet, enabling lively discussions on specific tweets within a thread.

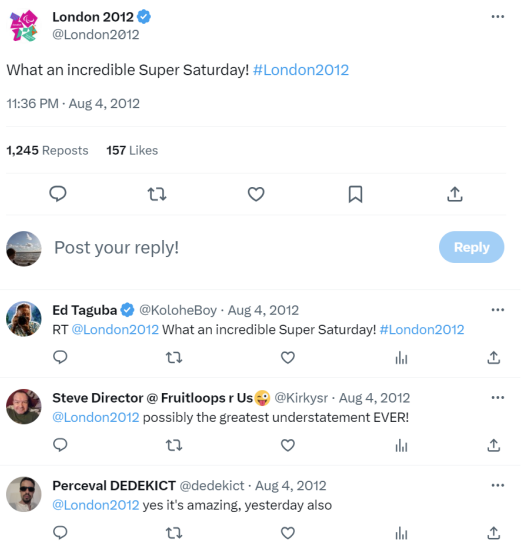


Figure 5.1: Sample Tweet - Twitter.

Fig. 5.2 shows two tweets from the *Olympics 2012* official account. The first *Tweet* had a total of 29 *replies*, while 205 *Retweets* were made. The second one represents such a

retweet, defined as forwarding another user's tweet. This opportunity is worthwhile, as information can be easily forwarded to followers to spread information and start discussions.

Figure 5.2: Sample Retweet - Twitter.

Moreover, Twitter has a unique *follower-followee* system, which is not based on a traditional bidirectional friendship system like Facebook. Each user can choose the accounts to follow, while the followed user can decide to re-follow or not, creating a unique use case, as users' popularity can be defined by their followers and followees.

5.3 Data Sets & Preparation

As already mentioned before in Section 2.1.1, data sets from Twitter are very suitable for the KD approach presented in Chapter 4, as Twitter is a very established social media service, present in day-to-day life and due to their availability over Twitters Streams API and Search API[21].

The availability of user activities such as *messages*, including creating new content (*Tweets*) and *responding* to and *forwarding* messages, made recording and extracting whole data sets possible but worthwhile. Also, the broad availability of further information considering basic user *profile features* and *reactions* to social network content substantiated the suitability of Twitter data sets for this approach. Thus, data sets from the last ten years until the beginning of 2023, where the API of Twitter was disabled for crawling new data sets, were considered.

Pointing now to the strategy of crawling data sets, it is helpful to capture specific topics, such as major sports events or other happenings of primary public interests, by filtering messages using commonly proposed *hashtags* for the specific events. While the long-term goal is to recognize user roles over miscellaneous data from various

[21] https://developer.twitter.com/en/docs/twitter-api

social media, this initial analysis concentrates on well-defined data sets that contain a substantial but manageable number of users.

Table 5.1: Overview on Twitter data sets.

Data Set	Messages	Users	Time Period	Category
Olympic Games 2012	13.68M	2.27M	Aug. 2012	sports event
Olympic Games 2014	14.58M	1.96M	Feb. 2014	sports event
Olympic Games 2016	38.05M	4.76M	Jul./Aug. 2016	sports event
Olympic Games 2020	119.02M	6.08M	Jul./Aug. 2021	sports event
Olympic Games 2022	43.76M	3.05M	Feb. 2022	sports event
FIFA World Cup 2014	109.00M	10.40M	Jun./Jul. 2014	sports event
2015 Paris Attacks	6.77M	0.74M	Nov. 2015	tragic incidence
NFL Super Bowl XLVII 2013	3.20M	0.64M	3. Feb. 2013	sports event
NFL Super Bowl LIV 2020	8.89M	0.89M	2. Feb. 2020	sports event
NFL Super Bowl LV 2021	10.36M	1.11M	7. Feb. 2021	sports event
NFL Super Bowl LVI 2022	12.51M	1.23M	13. Feb. 2022	sports event
2016 Berlin Truck Attack	0.66M	0.15M	19. Dec. 2016	tragic incidence

In order to transfer knowledge on user role detection, several classes of events were considered. Major sports events are repetitive and predictable, with numerous *messages* and *users* covering significant spans. Investigating both user roles in events with a shorter and extended period, the *Super Bowl*, which has 3-5 days around Super Bowl Sunday, and the *Olympic Winter Games*, with a period of 2-3 weeks, are suitable candidates. In contrast, events like the *Summer Olympics* and *Football* tournaments, which have a period of 4-6 weeks, complete the analysis, as the challenge of user role recognition in different types of events considering the period is a worthwhile attempt transferring the approach. Moreover, different types of sports provide an albeit limited thematic variance. These data sets are complemented by those of two *major disasters*, which also tend to have a strong yet very different topic focus and different interaction patterns. All in all, a plethora of specific events (cp. Table 5.1), such as sports events like the *Olympic Games*, the *FIFA Worldcup* or *American Football*, especially the *Super Bowl* but also tragic incidences such as the *Paris Attacks 2015* or the *Berlin Truck Attack 2016* were considered for this approach. Finally, the approach was applied to the Twitter *sample stream* instance to assess a data set without a strong topic focus.

5.4 Adapting the Methodology

After introducing the selected data sets and their specifications, in this section, the single steps of the primary approach, introduced in Fig. 1.1 of Section 1.2 and specified

in Fig. 4.1 of Section 4.2 will be performed initially on one of the Twitter data sets, which were introduced in Table 5.1 Section 5.3. The execution of the proposed pipeline will be explained based on the *Olympics 2012* data set, while particular amendments considering the data sets will be elucidated based on changes. The focus of this section is to initially find patterns and structures and improve standing-out characteristics to build a model that can label user roles (semi-) automatically.

5.4.1 Feature Engineering

The first aspect of the proposed pipeline is *Feature Engineering*, a significant part of the *Preprocessing* step, which includes feature selection and the aggregation of the messages considering the distinct users, which was introduced in more detail in Section 4.2. This section introduces all the meaningful steps needed before the novel *Multi-Sampling and Combination Strategy*.

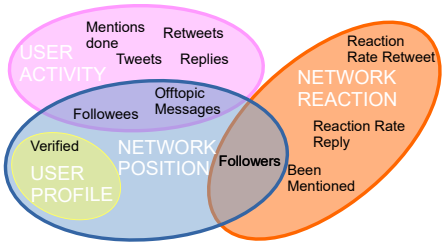


Figure 5.3: User feature classification - Twitter.

All records are delivered as JSON data with several different features. Some features can efficiently be utilized in their pure status, while others must be processed and chosen due to several aspects mentioned at the beginning of Section 4.2. Furthermore, features were chosen which are well established in the literature [Roc+11; LNN16; Cha+17]. The selected features can be seen in the Venn diagram in Fig. 5.3, which highlights the classes and instances of features, while a more detailed explanation can be seen in Table 5.2.

`Static user properties` expressing a (self-) description, such as the `verified` status of a user, are traditionally reserved for celebrities or influencers. The next class `user activity` is characterized by the number of original `tweets` of each user, which deal with the context of the event. Also, the amount of topically non-related tweets a user writes during the whole event is considered (`offtopic messages`) describing a user's behavior more precisely. Moreover, activities considering other users' tweets, such as

`replies` and `retweets` within the recorded topic, and mentions of other users give valuable insights into specific user behavior within the data set. Considering `network positions`, the number of `followers` and `followees` of a user at the time the data set is recorded are significant features and substantiate the potential to exert influence in the network. Also, the ability to trigger other users' `reactions` in the network is modeled by the `ratio` of tweets to `replies` and `retweets` and the frequency of `being mentioned` in messages, giving a worthwhile insight into the ability to influence other users and their standing within the social network.

Table 5.2: Overview on Twitter features.

Feature	**Description**
followers	The most recent number of users subscribed to users' content feed.
followees	The most recent number of users the user is subscribed to.
tweets	The number of newly created tweets of a user during the record of the data set. Describes the activeness of a user.
retweets	The number of topic-based retweets of a user. Describes the diffusion of information.
replies	The number of topic-based replies of a user. Describes the communicativeness of users.
offtopic messages	Number of all kinds of messages(tweets, retweets, replies) of a user during the recorded period unrelated to the given event.
verified	Status if a user is verified on Twitter. These users are real persons or organizations.
reaction rate reply	Percentage of how many users' tweets got at least one reply.
reaction rate retweet	Percentage of how many users' tweets got at least one retweet.
been mentioned	The number of times a user has been mentioned in other users' tweets.
mentions done	The number of times a user mentioned other users in a tweet.

Despite various additional features being evaluated and investigated from these classes, they delivered no additional value as they were correlated or had little discriminative power, such as a user having a URL. Also, complex network-based metrics like *centrality*, *spatio-temporal features* [Var+14] or c*ontent-analysis*-based features [Cha+17; Kao+19] were not considered as, e.g., graph data is only available up to a certain extent or has a higher complexity in terms of computation. In turn, the crawling strategy for data sets provides a particular topic focus, so there is no need for partial social graphs.

As feature selection is a very significant part of the Preprocessing step, it is also essential to validate features by metrics, such as their correlation to each other (cp. Section 2.2.1) but also ensure domain variance and skewness, which is inevitable for

later steps such as clustering because of the need of unified feature-spaces and fair feature comparison, w.r.t to feature-drifts across data sets.

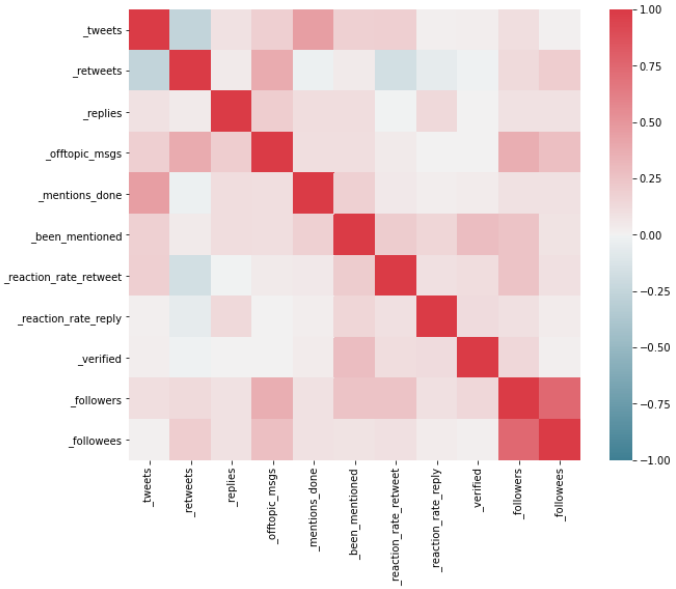


Figure 5.4: Correlation matrix for Twitter user features.

Investigating the *correlation* of the features described in the last paragraph, pairwise *features* in a symmetric *heat map* can be seen in Fig. 5.4 for the *Olympics 2012* data set. Other data sets were also investigated in terms of *correlation*. Regardless, as they all showed a similar correlation, the following investigations are presented based on the *Olympics 2012* data set, the initial data set for all analyses. The bar on the right-hand side visualizes if pairwise features have a high negative correlation (deep blue), no correlation (white), or a high positive correlation (deep red). As most feature pairs have no correlation or only a weak positive or negative correlation, the features `followers` and `followees` are the most correlated, as popular users show gains in either dimension. Nevertheless, changes in their ratio evolved a discriminative feature for specific groups, which led to considering both features. Likewise, some feature pairs with moderate positive correlation, e.g., `mentions done` and `tweets`, `offtopic messages` and `retweets` as well as `followers` and `offtopic messages` were also considered as they amend each other in hierarchical clustering.

Reducing the feature space is essential for clustering, but the data sets are also flooded

by many users, who occur only once in the recorded data set. To circumvent this effect, only active users with *at least two messages* are considered, preventing massive data sets due to complexity issues in the further steps. After reducing the number of features and the number of messages, the *aggregation* leading to feature vectors for each user can be performed.

Given the number of various features in social media exhibiting significant *skew* and value domain *variation*, the need for individual *standardization*(cp. Section 2.4) and *normalization*(cp. Section 2.4) of each data set is necessary, to reduce extreme outliers and set comparable bounds for all features within a data set. Before a suitable technique can be chosen, the raw features must be analyzed by exploiting characteristics such as *mean*, *median*, *skewness*, and *standard deviation* to estimate the suitability and effectivity of normalization and standardization techniques. More specifically, the skewness is reduced by using a *logarithmic transformation* that compresses extreme outliers, followed by a *Min-Max normalization* to bring the values into a range of 0 to 1. Miscellaneous other normalization methods discussed in [Osb10; FPT04] were also considered, e.g., inverse transformation, square root, cube root, box-cox, percentile, and rank transformation, but neither resulted in more balanced results. Moreover, standardization methods such as division by greatest value and z-score normalization analyzed in [ZC18; MC88] were also considered. The first technique guarantees a value range between 0 and 1, but the domain is not utilized entirely, while the latter technique does not provide a specific range. In addition, those techniques had issues with the following clustering techniques, resulting in blurry and not well-separated clusters that were difficult to analyze. Furthermore, each data set is standardized and normalized disconnected individually, independent from knowledge of future data sets, capturing the relative distribution differences and tracing feature drifts when comparing them in chronological order.

Table 5.3 as well as the boxplots in Fig. 5.5a and 5.5b show the properties of the *Olympics 2012* data set features before and after the normalization and standardization process. Like the correlation investigation, the other data sets delivered a similar outcome considering the feature statistics. Thus, the normalization and standardization are only presented based on the initial data set.

Most of the *original features* in Table 5.3 excluding `offtopic messages`, `reaction rate` for `retweets` and `replies` and the `verified` status contain strongly *right-skewed* data represented by either higher median values reaching the high 99th percentile and maximum (`followers`, `followees`) or a high amount of outliers beyond the upper whisker in the boxplots in Fig. 5.5a, depicting the raw data set with a log y-axis even to show extreme outliers. Besides the features of `followers` and `followees`, `offtopic messages` and `been mentioned` have similar effects considering the standard deviation, as a *symmetric deviation* around the mean values are not given. Some features do

not have a box, e.g., `reaction rate retweet`, `reaction rate reply`, or `replies`, but the *standard deviation* has lower values, showing a proper deviation around the means. Examining the *standard deviation* and *skewness* is essential as they represent a powerful indication of the necessity of *standardization* and *normalization* when combined, leading to almost balanced *skewness*, standard deviation, and median values in equal bounds, smoothing out outliers without affecting the characterization of features too much (cp. Fig. 5.5b). Furthermore, preprocessing helps to unify feature spaces, gaining comparability across data sets revealing feature drifts, discussed more in detail in Section 5.6 when comparing the individual data sets against each other.

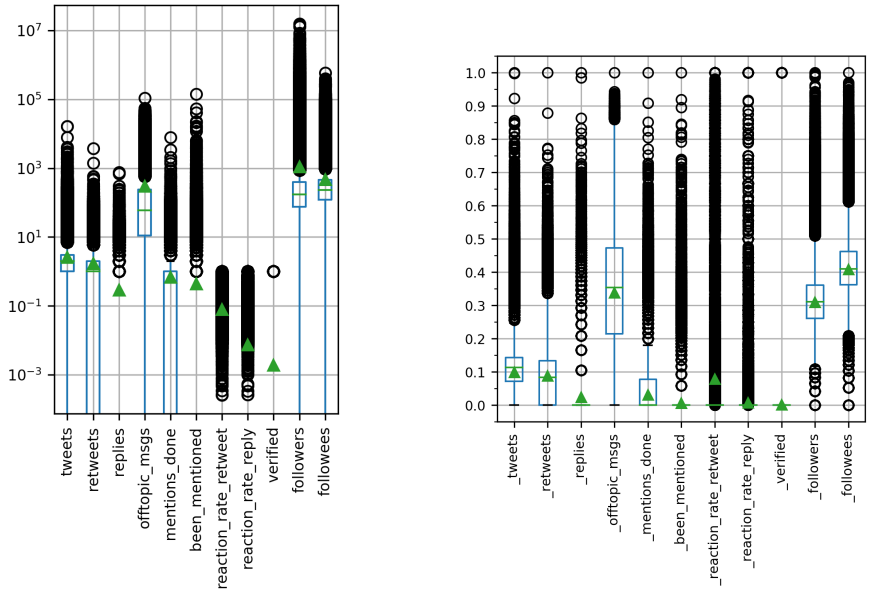
Table 5.3: Original feature statistics Olympics 2012 - Twitter.

Features	**Median**	**99%**	**Max**	**Skew**	**StD**
tweets	2	19.00	16621	543.97	20.72
retweets	1	13.00	3780	331.82	4.26
replies	0	3.00	759	204.19	1.23
offtopic messages	59	3790	107484	13.28	879.53
mentions done	0	8.00	7802	723.10	6.89
been mentioned	0	3.00	141086	1017.85	110.13
reaction rate retweet	0	0.75	1	2.55	0.18
reaction rate reply	0	0.33	1	8.67	0.05
verified	0	0.00	1	22.62	0.04
followers	172	9520	15769360	229.46	35271.96
followees	231	2908	583595	74.67	1958.20

Table 5.4: Normalized feature statistics Olympics 2012 - Twitter.

Feature	**Median**	**99%**	**Skew**	**StD**
tweets	0.11	0.31	0.72	0.30
retweets	0.08	0.32	0.75	0.29
replies	0	0.21	2.52	0.16
offtopic messages	0.35	0.71	-0.19	0.18
mentions done	0	0.25	2.26	0.23
been mentioned	0	0.12	6.73	0.02
reaction rate retweet	0	0.75	2.55	0.18
reaction rate reply	0	0.33	8.67	0.05
verified	0	0	22.62	0.04
followers	0.31	0.55	-0.07	0.09
followees	0.41	0.6	-0.97	0.09

Generally speaking, the relative feature distributions after normalization varied only slightly over time from 2012 until early 2023, with minor changes: users tend to move slightly more into *reactive* behavior of *forwarding* than content generating or mentioning, while the `verified` status is now much more prevalent. Overall activity increased moderately, and forwarding actions became more widespread.



(a) Raw user features.

(b) Normalized and standardized user features.

Figure 5.5: Boxplot comparison of features.

5.4.2 Cluster Analysis

Clustering and *Cluster Analysis* are very central matters in this work, finding groups of user roles, as unsupervised learning can witness structures of similar objects as already introduced in Section 1.4 and 2.5. Pointing to the general methodology from Chapter 4 in Section 4.4, the way from coarse to fine-grained clusters is familiarized more in detail, which is the foundation for the manual class labeling in Section 5.4.3.

A very central question was utilizing clusters of users and detecting user roles using hierarchical agglomerative clustering[22] with Ward's linkage, attaining a hierarchical

[22]https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html

structure of users and encouraging an easier explainability among the cluster hierarchy. Mentioned in Section 4.4, a broad range of clustering techniques was evaluated, but neither worked as well as hierarchical clustering with Ward's linkage. An essential issue of hierarchical clustering arising from the technical side is the complexity of $O(n^2)$ requiring longer running times and higher memory costs, which is counteracted using the *Multi-Sampling and Combination Strategy*, as randomly drawn samples can be clustered faster and combined probabilistically afterward. Aspects considering tuning this strategy as well as reasoning and experiments will be introduced more in detail in Section 5.5.2 and 5.5.3 by reasoning *coverage* and *certainty* with experiments w.r.t. several *sampling strategies*, *sample sizes* and the *number of samples* for combination.

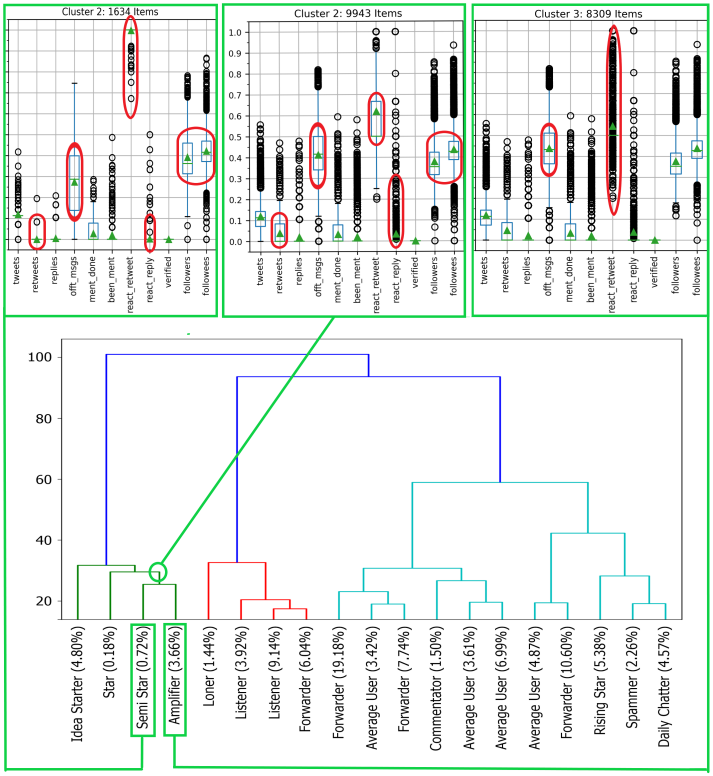


Figure 5.6: Example with dendrogram and boxplots for Twitter.

Focusing now on the analysis of a 10% sample from the *Olympics 2012* data set, which can be seen in Fig. 5.6, one can see the structure of a dendrogram. As part of the

cluster analysis, introduced in section 4.4.2, internal quality metrics play a significant role in finding the most valuable point to look inside the hierarchical clustering. In the very beginning of analyzing entirely new data sets, *internal quality measures* such as the *Silhouette Coefficient* (Section 2.6.2.1), *Davies-Bouldin Index* (Section 2.6.2.2) as well as the *Calinski Harbasz Index* (Section 2.6.2.3), help to find an anchor of coarse-grained structures in the analysis.

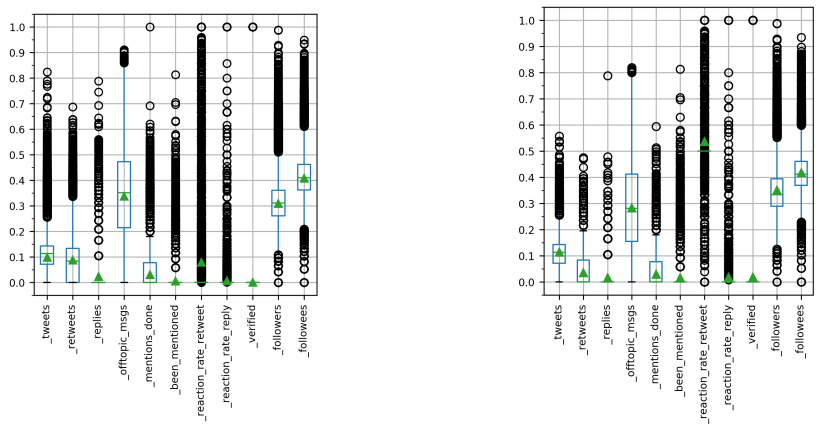
As internal quality measures did not provide a significant mutual deflection for one clustering but rather several deflections for varying samples, a fully automatic approach for analyzing clusterings cannot build upon those metrics. Thus, a suitable approach for cluster analysis was presented in Section 4.4.2 relying on the *elbow* and a *depth-first search* investigating *effect size changes* of features in the cluster hierarchy.

Focusing now on analysis metrics depicting a broad range until 40 clusters, the *Silhouette* values in Fig. 5.8a continually show values below 0, which has hardly explanatory power in terms of cluster analysis, while the best values are delivered for clustering with 3 clusters. When examining those clusters, generalized user roles can be determined, which do not satisfy the contributions of providing fine-grained user role detection. Also, reckoning the *Calinski-Harabasz Index* in Fig. 5.8b indicates the best clustering for 6 clusters. Compared to the *Silhouette* values, slightly more clusters, and thus some more fine-grained clusters, are delivered, but this result is not satisfactory. Likewise, the *Davies-Bouldin Index* furnished similar results, as the possibly best clusterings were those with 3 or 5 clusters. Inspecting other samples, similar observations can be made when varying sample sizes. In most cases, the internal quality metrics delivered higher deviations from each other, considering the number of clusters. Those provided cluster candidates cannot be interpreted as fine-grained user roles and can hardly be used automatically to find the most suitable clustering.

A more structural and heuristic-driven analysis already introduced in Section 4.4.2, forces to assign fine-grained user roles automatically, as internal quality metrics primarily tend to deliver coarse-grained user roles. Regardless, before this is even possible, specific *representatives* of *user roles* from the *literature* have to be found by manually inspecting the boxplots through the dendrogram. Focusing again on the results of the internal cluster metrics, the boxplots of, e.g., 5 clusters and their feature deviations from the boxplot of the whole sample reveal only some *generalized roles*, which deviate only in fewer features. In Fig. 5.7b a boxplot for the clustering with 5 clusters can be seen, while Fig. 5.7a shows the boxplot for the whole sample.

The *Elbow method* (cp. Section 4.4.2) from Definition 32 provided for most samples a more precise starting point for locating generalized roles in clusterings than the analysis using the *internal quality metrics* such as *Silhouette*, *Davies-Bouldin Index* or *Calinski-Harabasz*, showing good separations, w.r.t distances from the linkage matrix, as, e.g., the 10% sample in Fig. 5.8d has its highest acceleration (orange line) at 3

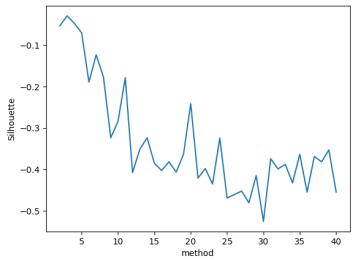
clusters. Generally, the *Elbow* method delivered between 3 and 5 generalized roles for all samples, being a good starting point for a depth-first search (cp. Definition 33 from Section 4.4.2) finding fine-grained user roles.



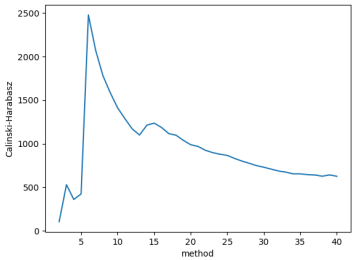
(a) Boxplot of the 10% sample.

(b) Boxplot cluster 0.

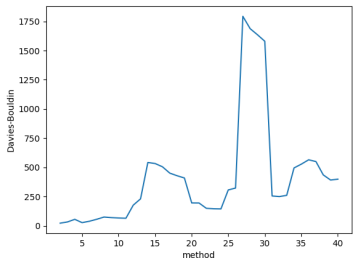
Figure 5.7: Example boxplots for Twitter.



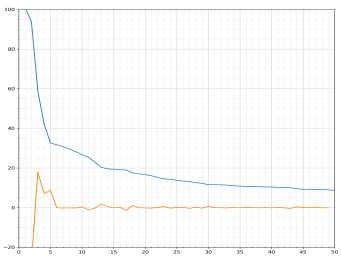
(a) Silhouette of 10% Oly12 sample.



(b) Calinski-Harabasz of 10% Oly12 sample.



(c) Davies-Bouldin of 10% Oly12 sample.



(d) Elbow for 10% Olympics 2012 sample.

Figure 5.8: Comparison of several cluster evaluation metrics.

In contrast to the results of the Elbow method, internal cluster metrics sometimes also delivered suitable clusterings with a number of clusters between 3 and 5 clusters but had a scattering when analyzing the samples, making them hardly appropriate as the starting point for the depth-first search. The direction can now be set more on tuning the significance criteria in the pairwise feature comparison of effect sizes to find a possibly best clustering, which will be discussed later in Section 5.6.

Focusing again on both the boxplots as well as a tool, which analogizes pairwise feature-deviations using the effect size in Fig. 5.9, one can see several colored dots in the dendrogram, indicating the highest occurring effect size considering all features in a pairwise comparison. Moreover, in this tool, the boxplots are illustrated and highlighted in those colors to show the degree of the effect size, which is a vital strategy to substantiate the manual class labeling in the following section.

5.4.3 Manual Class Labeling

The discussion in Section 2.1.2 clearly states that there is no agreement on the types of user roles or precise definitions or models in social media. Only *descriptions* of user

roles were defined, as already clarified in related work of Section 4.8. As established in the previous Section 5.4.2 *Cluster Analysis*, the cluster hierarchy, as well as the tools for the analysis, provided indications on an approximated number of clusters and their separation within a refinement process from clusters representing coarse-grained user roles to those representing fine-grained user roles.

The *Manual Class Labeling* process for the Twitter use case will be performed as already introduced in Section 4.5 by analyzing clusters with *differing sample sizes* from the data sets. Thus, clusters and their deviations within the hierarchy and suitability w.r.t. fine-grained structures were evaluated and labeled manually with user roles from the literature. As part of a student's project and thesis with around three to four weeks in total, this manual class labeling strategy was pursued iteratively without any use of tools until user roles could be allocated unequivocally to the clusters by inspecting and comparing boxplots against each other. Of course, this time-consuming strategy includes several *reallocations* and *comparisons* as supposed clusters from distinct samples may differ more than expected. The insights gained from this strategy were precious for further analysis and building training data for the classifiers, as they gave a sense for manual class labeling.

Pointing back to the boxplot from Fig. 5.7b and the central boxplot in Fig. 5.9 relying on the clustering with 5 clusters at a distance of 40. At this point, mostly 3-5 generalized user groups can be found, which all are expressed by different feature values, which can be seen in Table 5.5 and rely on the dendrogram from Fig. 5.6.

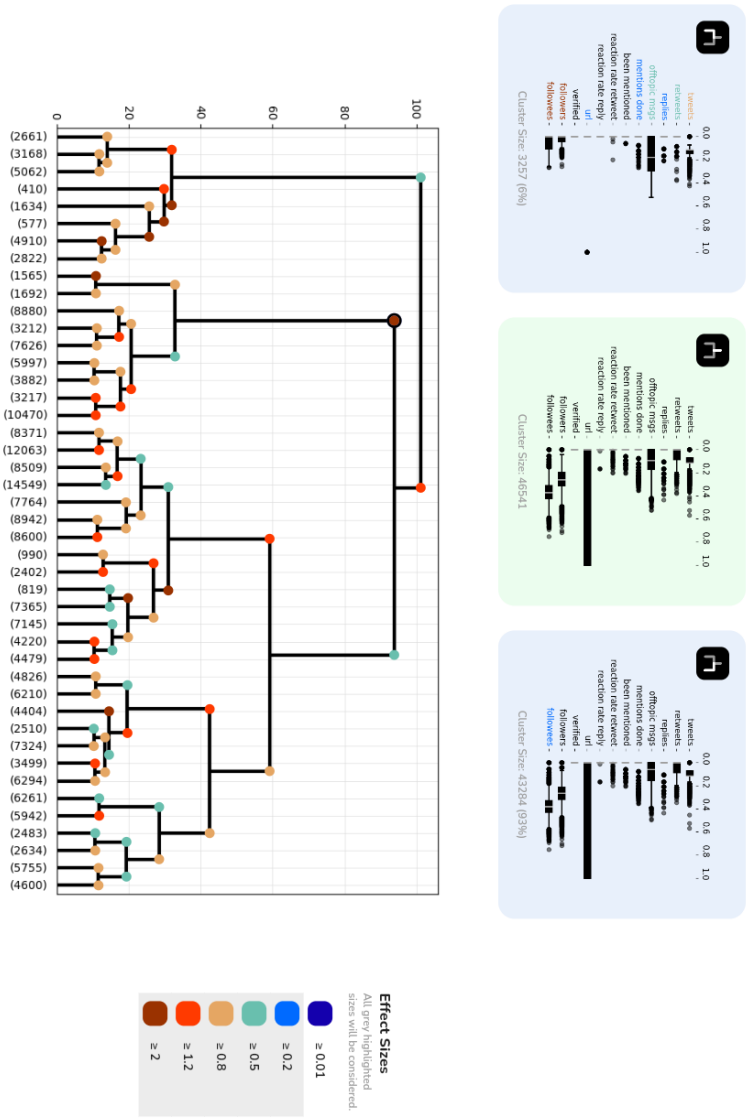


Figure 5.9: Effect size-based feature changes across the dendrogram.

Table 5.5: Significant feature changes for coarse-grained user roles.

Cluster	Description	Significant features
Cluster 0	trigger strong reactions	high tweets, retweets, replies, being mentioned, reaction rate retweet
Cluster 1	passive users	weak network positions (follower, followees) tend to more offtopic messages
Cluster 2-4	variations of intermediate users	higher offtopic messages,

The first substantial group (green) shows users that can trigger *strong reactions* have more `retweets`, `replies`, and tweets and are `being mentioned` more often, the second (red) shows *passive users* with relatively *weak network positions* (`followers`, `followees`), while the group(s) in between show various degree of *moderate activity and impact*. A strong motivation for fine-grained roles can be glimpsed even further down the tree (cp. boxplots in Fig. 5.6). With the aid of the *effect size-based cluster evaluation tool* from Fig. 5.9, a more structural *fine-grained user role detection* is realized starting from the *generalized user roles*, which will be examined more in detail in this section dealing with manual class labeling as already stated in Section 4.5.

The two complementary approaches mentioned in Section 4.5 ensure the *mapping* of user roles from the literature to the unlabeled clusters, namely the *manual-driven analysis* of boxplots and the proving of *training data* with the aid of dimensionality reduction. For the first approach concentrating on a manual analysis of the overall structure of the clustering by focusing on feature deviations in boxplots within the cluster hierarchy, Fig. 5.6 depicts the feature *deviations* between the *parent cluster* and their *two child clusters*, reaching user roles like *Semi-Stars* or *Amplifiers*. The second approach focuses on dimensionality reduction strategies, such as PCA or LDA, forcing the exploration process of user roles in several ways (cp. Section 2.8) shows that correlated features in the reduced dimensions of a PCA are evident, while a LDA cannot gather conclusions on considered features. This aspect makes the PCA more valuable. Finally, feature drifts and changes in user roles across multiple data sets can be accurately identified using LDA and PCA.

The iterative analysis strategy for analyzing boxplots and their feature deviations within the dendrogram introduced in Section 4.5 is applied to the data sets to first find well-defined user roles from literature as an anchor. Well-studied user roles like *Star* users, which have a large number of `followers`, almost always a `verified` status, and a generally high impact despite *relatively low activities* in the social network, can easily be matched on the given aspects, as those roles are portrayed well in the literature. All aspects are mapped to user roles when possible. However, other stable user roles, which match precisely those from the literature but occur often enough,

are considered as they lead to more specific user roles or thoroughly new user roles in between the roles defined in the literature.

Pointing again to the dendrogram from Fig. 5.6 3-5 subtrees can be found in the dendrogram representing the *coarse-grained clusters* from Table 5.5. The coarse-grained user roles are present in several samples across the data sets, also with a varying number of users, delivering three major generalized user roles, which can be seen in the Venn diagram in Fig. 5.10. The user roles are defined as follows: Besides the *Star* user, there were a lot of other *action-triggering* user roles such as *Semi Stars* or *Idea Starters*, which are similar to *Semi Stars* but obtain popularity in the network by *creating more content* and *triggering higher reactions*. *Amplifiers* are *well-networked* users pushing and spreading predominantly (existing) trends and content, while *Rising Stars* are gaining a more impressive number of `followers` as they are more active in the network and thus are receiving significant reactions on `retweets` but not yet at the high level of *Stars* or even *Semi Stars*. Thus, the latter two user roles fit in the generalized user role of *intermediates*. *Intermediate* users mostly rely on the *Average User* role, which can hardly be distinguished considering feature deviations and statistical indicators compared to the whole data set. This role is also the most prominent representative in each sample. There are also user roles of *Spammers*, which have mostly a high activity in the network but are not as popular as the *action-triggering* users; *Daily Chatters*, which are not *as active* and more *moderate* compared to *Spammers* and *Commentators*, who define themselves as *Daily Chatters* by creating more *own content*, a higher rate in `retweeting`, *triggering more reactions* considering `replies` and thus being *more active* in the network.

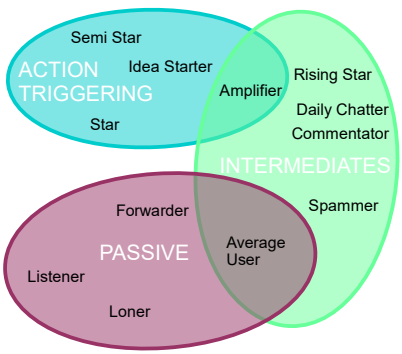


Figure 5.10: User roles - Twitter.

The third generalized group are *Passive* users, such as *Forwarders*, who are defined as *Average Users* with a *better connection* in the social network but only tend to *forward content* and finally receive only *common reactions* in the network. *Listeners* are users who primarily only *consume* content instead of *creating* their own content and thus *hardly trigger* other users. Furthermore, they have a *weak connection* in the network and are only underbid by the user role *Loner*, whom almost *inactive users* represent.

Table 5.6 recapitulates now the insights from the evaluated user roles from Fig. 5.10 as well as the feature descriptions from Table 5.2 by describing the user roles with remarkable feature characteristics, in particular, more substantial *feature deviations from the average*, and comparing them to other similar user roles (second column). Furthermore, in the third column, *quota values* for each user role over all samples were evaluated, resulting in a dominance of passive users, excluding the *Loner* role, while *action-triggering* users like *Stars*, *Semi Stars*, *Idea Starters*, and *Amplifiers* tend to occur rarely in the data sets. Except for the *Commentator*, the quotas for *Intermediate* users lie primarily between the *Action-Triggering* and *Passive* users.

Table 5.6: User roles and their characterization: $\approx$ shows closeness to other roles, $\downarrow/\uparrow$ feature deviation from close role/whole data set, $\searrow$ / $\leftrightarrow$ / $\nearrow$ changes over time

	Role	Characteristics	Frequency
action trigg.	Star	followers > followees, verified, $\downarrow$ activity, $\uparrow$ mentioned	0.2–0.8%
	Semi Star	$\approx$ Stars, $\downarrow$ followers, mentioned, $\uparrow$ react. (re)tweet, retweets, replies	0.2–1.4 %
	Idea Starter	$\approx$ Semi Star, $\downarrow$ followers, $\uparrow$ reactions	1–4%
	Amplifier	$\approx$ Idea Starters, Semi Stars, $\uparrow$ followers, followees	0.5–5%
intermed.	Rising Star	$\approx$ Semi Star, Idea Starter, Amplifier $\uparrow$ followers, (re)tweets, replies	1.5–5.5%
	Daily Chatter	$\approx$ Average User, Spammer, $\downarrow$ (re)tweets, offtopic	5–15%
	Commentator	$\uparrow$ replies, offtopic, reactions	0.3–2%
	Spammer	$\uparrow$ (re)tweets, replies, offtopic $\downarrow$ followers, followees, reactions	1–7%
	Average User	offtopic > tweets, retweets	8–30%
passive	Forwarder	retweets > tweets, $\uparrow$ offtopic, followers, followees. $\downarrow$ reactions	25–65%
	Listener	$\downarrow$ (re)tweets, reactions	6–20%
	Loner	$\downarrow\downarrow$ tweets, offtopic, followers	0–1.5%

Finally, manual class labeling in this use case helped to initially detect and define prominent user roles with aid from literature, which could be mapped on clusters.

Starting with *coarse-grained user roles*, *manual class labeling* paved the way for detecting a plethora of *fine-grained user roles*, which are essential for building and verifying classifiers. Furthermore, the elaborated user roles represent a good foundation when transferring the whole approach and the knowledge of fine-grained structural user roles to other use cases, such as the Telegram use case in Chapter 6.

5.4.4 Building a Classifier

The next step in the KD pipeline from Section 4.6 deals with *classifying* user roles and *building a suitable classifier* to automatically label clusters based on the insights from the manual labeling process presented in the previous section. With a carefully built classifier, it is uncomplicatedly possible to *transfer knowledge* on user roles starting from existing data sets or samples of the identical data set, which have already been labeled, but also from topically related or close-in-time data sets. Based on Section 4.6, where the main characteristics and challenges of classification were already conveyed, several particularities will be presented in more detail.

Essential for classification success are two types of input: Firstly, the *training data* itself consisting of partially manual labeled cluster means, whose finding process was clarified in Section 5.4.3, and secondly, *unlabeled clusters*, which should be classified with user roles. To build valuable training data for the classification process in this use case, a more manual-driven approach (cp. Section 4.5) was pursued instead of an Active Learning (AL) strategy as aspects of explainability within the whole approach was pursued. Even though the effort is higher as much human intervention is needed, it helps to comprehend the whole creation process of training data. To handle this aspect, the best practice is to enrich training data with striking clusters by adding whole feature vectors to the training data representing not single users but whole clusters. As mentioned in Section 4.6, the enrichment of training data at the beginning of building a classifier is aided by a ground truth from the manual labeling process, evaluating deviations between classification results and the ground truth for each user role. The foremost goal is to devise a strategy that has the potential to ensure *scalability* and *user role probabilities* for each user. This goal will be attained by recognizing and scrutinizing crucial features and their deviations as exemplified in Fig. 4.8. Further experiments considering *certainty*, *stability*, and *coverage* of user roles revealing the suitability of a probabilistic classification as part of the novel *Multi-Sampling and Combination Strategy* will be discussed in Section 5.4.

The second input for the classifier are the clustered and analyzed representative samples, resulting in a set of unlabeled clusters. Manual labeling of clusters is inadequate as it usually assigns only one or two user role labels per cluster, which can be unclear due to blurry user roles. In contrast, the classification process is superior as it assigns

a probability vector to each user in the cluster, indicating the likelihood of the user fitting each possible user role based on cluster means in the training data sets.

Due to their significant features, the most notable and distinctive cluster means were selected and verified from the manual labeling process described in Section 5.4.3 to obtain the most optimal training data set for each user role. Furthermore, an *iterative enrichment process* was applied to these clusters, supporting dimension reduction techniques like Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA). For most of the data sets from Table 5.1, *training data sets* were created, while in this section, the process based on the *Olympics 2012* data set will be demonstrated. Fig. 5.11 shows the training data set with reduced dimensions utilizing PCA. This iterative process of creating training data was time-consuming as clusters needed to be selected at the most suitable spot due to their quality and feature significance, as well as a manual classification to map each of those clusters to the most fitting user roles from literature to gain meaningful and well-separated representatives of user role.

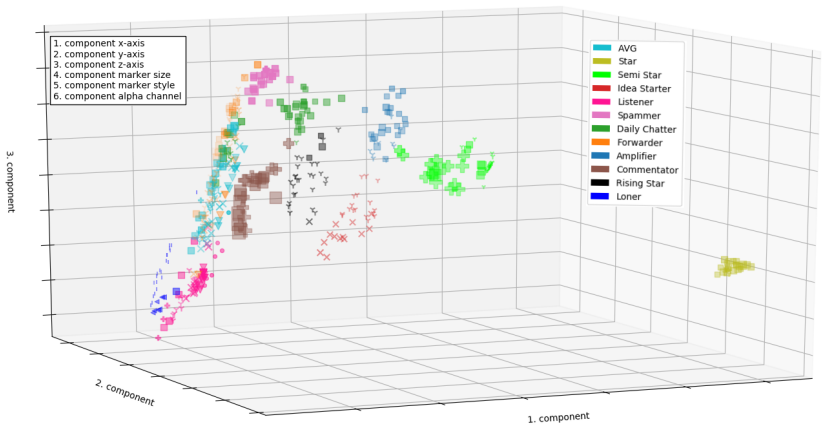


Figure 5.11: PCA of clustered samples from *Olympics 2012* data set.

The PCA was visualized in a *3-dimensional coordinate system* with additional specifications such as *marker size*, *style*, and *alpha channel* to depict the most significant *components*. Table 5.7 shows each component's percentage of the total variance and the top 3 correlated features. The three axes represent more than 85% of the summed-up total variance, and the features `reaction rate retweet`, `followers`, and `verified` and `offtopic messages` are the most correlated features in the top 3 components.

The complexity reduction in the multidimensional cluster means of the training data from the *Olympics 2012* dataset is clearly shown in Fig. 5.11 using the PCA.

The training data reveals distinct clusters for each user role, indicating successful simplification. There are user roles that are very well-separated from others, like the *Star* User or *Semi Stars*, but also user roles that lie close to each other, such as *AVG Users* vs. *Forwarders* vs. *Daily Chatters*. This aspect emphasizes the importance of creating training data thoroughly, precisely, and consequently, to reach the best classification results possible. The LDA delivered a similar outcome and proved the appropriateness of the training data set, which can be seen in Fig. 5.12.

Table 5.7: Variances and Top 3 features of the six present components in PCA of Olympics 2012 training data set.

Component	Variance	Top Features
x-axis	50.07%	reaction rate retweet, verified, followers
y-axis	22.32%	verified, reaction rate retweet, followers
z-axis	14.37%	offtopic messages, followees, verified
marker size	6.72%	reaction rate reply, followees, verified
marker style	4.67%	followees, offtopic msgs, followers
alpha channel	0.87%	tweets, mentions done, replies

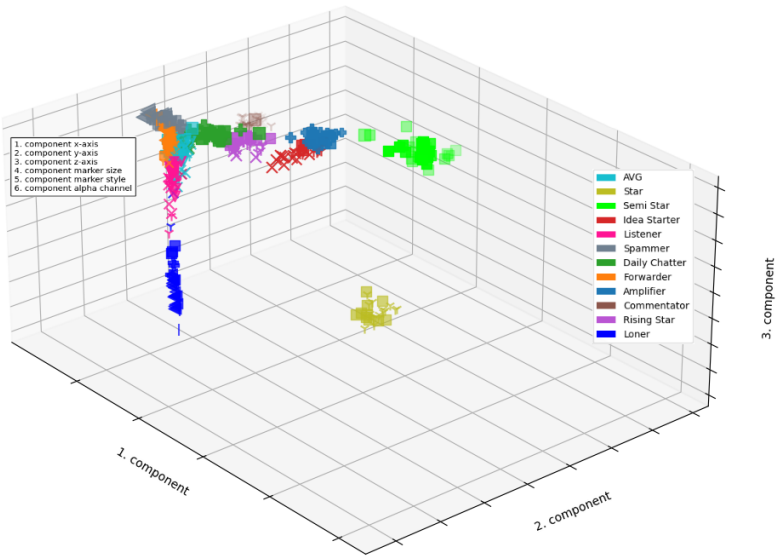


Figure 5.12: LDA of clustered samples from *Olympics 2012* data set.

The *training data set* of the *Olympics 2012* itself consists of clusters from varying sample sizes (16 5% samples and 10 10% samples) with 507 manually analyzed clusters. Table 5.8 shows the distribution of the user roles in the training data set, whereas user roles such as *Average User*, *Forwarder*, and *Listener* are unsurprisingly represented more dominantly, as they place the most dominant clusters in the data sets. The remaining user roles are almost equally represented in the training data.

Table 5.8: A priori distribution of user roles in the Olympics 2012 training data set.

Average User	**Star**	**Semi Star**	**Idea Starter**
87 (17.16%)	26 (5.13%)	45 (8.88%)	30 (5.92%)
Listener	**Spammer**	**Daily Chatter**	**Forwarder**
54 (10.65%)	22 (4.34%)	43 (8.48%)	81 (15.98%)
Amplifier	**Commentator**	**Rising Star**	**Loner**
33 (6.51%)	27 (5.33%)	30 (5.92%)	29 (5.72%)

Since *dimensionality reduction* works well for creating and validating well-separated *training data sets*, it led to issues when reducing the original features before the clustering process. As many various features are given, dimensionality reduction is no option before clustering, as already discussed in Section 4.4 due to a potential loss of quality and explainability of features. After clustering, the *effect size-based depth-first search* for cluster analysis would not give valuable insights into the cluster analysis, as conclusions for feature deviations cannot be drawn clearly. Furthermore, the trade-off between classifying individual users instead of whole clusters was investigated, too, but led to lower classification performance. The whole clustering process could be skipped, but the inherent noise of individual users, especially outliers, did not lend well to training and classification. This was the main reason to focus on training a classifier with *representatives* of clusters, such as the *mean feature vectors*. Furthermore, the *median* and a *boosted technique* where *means* were enriched with *pooled Cohen's d* values were also considered. Means tended to provide better separation than medians, while the pooled Cohen's d can capture the temporal evolution of features.

In Section 4.6, several well-known classifiers were introduced, all considered for experiments to cover several specifications of the training data sets. Since some of the clustered data sets are relatively small and skewed, the support for those classes is low. To counteract these specifications, methods based on ensembles of decision trees, e.g., *Gradient Boosted Decision Trees (GBM)* or *Extremely Randomized Trees (ET)*, multi-class *Support Vector Machines (SVM)*, or *K-Nearest Neighbor (KNN)* turned out to be most suitable.

5.5 Instantiating & Assessing the Classifier

The previous section initially utilized the methodology from Section 4.2 with all the parts starting with *Feature Engineering*, *Sampling*, *Clustering*, *Cluster Analysis* until the *Classification*. The first clustering results gave valuable insights into the approach's suitability, but *optimizations* are possible and may lead to favorable results. On the one hand, the classifiers work but have not been adjusted yet considering the parameters. Moreover, several sampling strategies will be evaluated due to their suitability w.r.t. stability of user roles and coverage of the data sets.

5.5.1 Hyperparameter Tuning

After demonstrating the process of manually labeling and building a classifier for most of the data sets from Table 5.1, the *classifiers* need to be *trained* and *validated*. The setup to build and validate training data sets utilized *repeated stratified cross-validation* with three splits, where one was left out due to the small amount of data, and three repetitions with different permutations to cater for possibly missing groups. Both feature means and enrichment with pooled Cohen's d were considered for each classifier mentioned at the end of the previous section. To prove the quality of the training data, *F1-macro* (cp. Section 2.7.2) was used as a metric to compensate for class imbalance and prevent focus on either precision or recall. Furthermore, a *grid search* was applied to tune the *parameters* of each classifier. As each classifier has specific parameters, a broad range of possible parameter spaces was considered to find the top 3 configurations for each classifier, ensuring a high degree of robustness. The grid search itself was applied to the *Olympics 2012* data, while for additional data sets, the established configurations from the grid search were applied and evaluated with the aid of the *ground truth*.

All investigated classifiers learn and generalize well, leading to a 94-96% score in validation and 96-99% score in training sets with no stronger or weaker candidates, w.r.t. over and underfitting effects proved by an insignificant standard deviation as can be seen in Table 5.9. The objective for each classifier was to identify the single best configuration and enable greater flexibility for adjusting classifiers on new data sets. Thus, the *top 3 configurations* for each classifier were figured out using a *grid search*. In Section 5.6, the created training data for other data sets from Table 5.1 will be utilized to train and validate the classifiers with the knowledge suggested in this Section, while in Section 5.7 several classifiers with variable training data will be used to classify diverging as well as entirely new data sets.

Table 5.9: Results of top 3 configurations for the classifiers.

		Mean			Mean & Pooled Cohen's d		
		Top 1	**Top 2**	**Top 3**	**Top 1**	**Top 2**	**Top 3**
GBM	avg. validation score	0.9400	0.9400	0.9400	0.9459	0.9459	0.9459
	std	0.0120	0.0120	0.0120	0.0094	0.0094	0.0094
	avg. train score	0.9866	0.9866	0.9866	0.9896	0.9896	0.9896
	std.	0.0054	0.0054	0.0054	0.0036	0.0036	0.0036
ET	avg. validation score	0.9507	0.9507	0.9507	0.9486	0.9484	0.9482
	std.	0.0131	0.0126	0.0134	0.0152	0.0148	0.0160
	avg. train score	0.9660	0.9674	0.9675	0.9698	0.9633	0.9657
	std.	0.0049	0.0055	0.0055	0.0035	0.0034	0.0049
SVM	avg. validation score	0.9525	0.9525	0.9525	0.9496	0.9496	0.9495
	std.	0.0071	0.0071	0.0071	0.0112	0.0112	0.0112
	avg. train score	0.9723	0.9725	0.9723	0.9753	0.9753	0.9753
	std.	0.0031	0.0035	0.0031	0.0042	0.0042	0.0042
KNN	avg. validation score	0.492	0.9486	0.9484	0.9425	0.9407	0.9400
	std.	0.0062	0.0098	0.0099	0.0139	0.0132	0.0112
	avg. train score	0.9630	0.9616	0.9606	1.0000	0.9561	0.9554
	std.	0.0039	0.0050	0.0040	0.0000	0.0058	0.0061

5.5.2 Stability & Coverage of User Roles

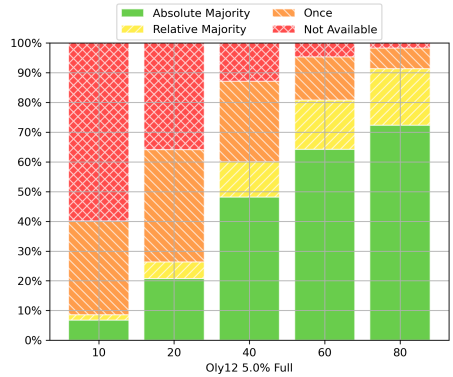
Having built and optimized the classifier in Section 5.4.4 and 5.5.1, it is now possible to investigate several questions considering *user distributions*. In the creation process of *manual training data* in Section 5.4.3, several user roles from the literature were mapped on clusters, revealing a still loose distribution of user roles, which can be seen in Table 5.6. This distribution, as well as the following questions, will now be answered in this section:

- Does varying sample sizes and the number of samples influence the coverage?
- Do multiple captured users impact the stability and certainty of user roles?
- Is a correlation of user roles stemming from the same generalized role w.r.t. second best user roles noticeable?

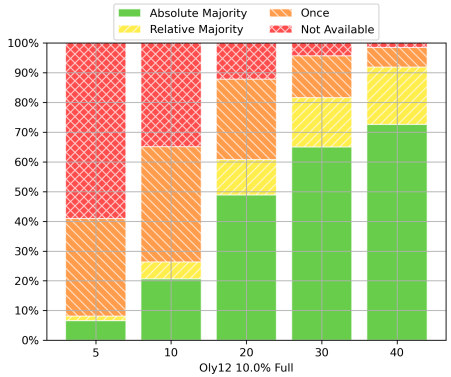
Starting from the *Olympics 2012* data set samples, the trained classifiers labeled each sample to determine how user roles are distributed among the samples. Overall, in

the *Olympics 2012* data set, *12 roles*, which are also described in Table 5.6 and Fig. 5.10, were encountered. Note that these user roles are not strictly defined for all data sets but pretend to be a good starting point for evaluating all further data sets.

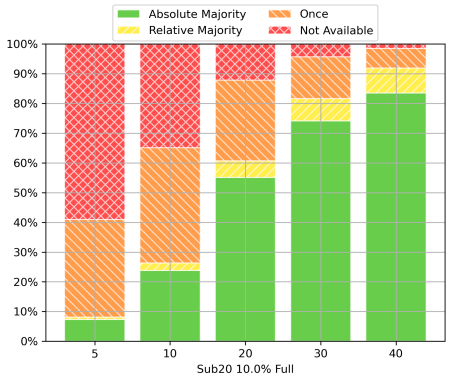
Influence of Sampling on Coverage The roles' *coverage* and *certainty* were analyzed after combining the clustered and classified user data from random samples. The outcomes for the *Olympics 2012* and *Super Bowl 2020* data sets with varying sample sizes (5% vs. 10% and 10% vs. 20%, respectively) were demonstrated in the subfigures of Fig. 5.13, delivering a highly reliable and valuable outcome.



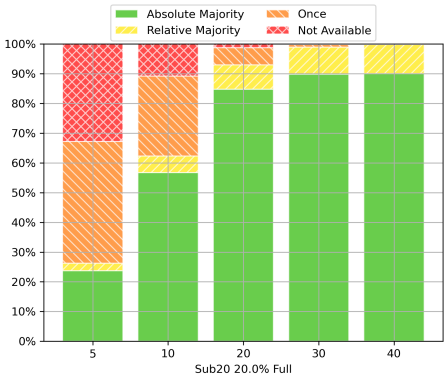
(a) *Olympics 2012* 5% samples.



(b) *Olympics 2012* 10% samples.



(c) *Super Bowl 2020* 10% samples.

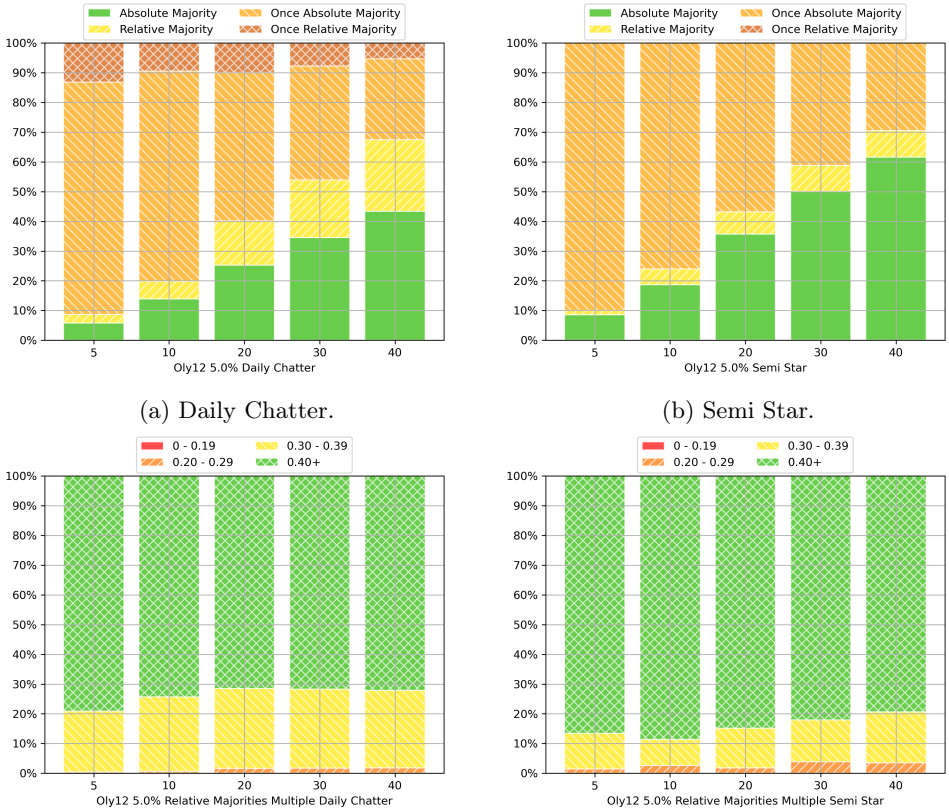


(d) *Super Bowl 2020* 20% samples.

Figure 5.13: Coverage comparison Random Sampling for full data sets.

When increasing the number of samples and thus the number of individually classified users, the number of role assignments per user improves. As a result, the number of users without any role assignment (red bar) drops quickly, while the number of users with multiple, mostly consistent role assignments (green bar) grows rapidly. Furthermore, the number of users that only see a single role assignment (orange bar) becomes smaller, enabling them to perform actual probabilistic assessments on the assignment certainty. In turn, the increasing relative majority part (yellow bar) gives insights on users that are not well identified as the most emerging user role has no absolute majority, - which is data set-dependent. When utilizing bigger, yet fewer samples (Fig. 5.13b and 5.13d) compared to the combination of smaller samples (Fig. 5.13a and Fig. 5.13c) for the same number of users, the quality of the results tends to be slightly better (in particular for the *Super Bowl 2020* data set), yet at much higher resource requirements due to the quadratic complexity for clustering. Hence, opting for smaller yet more frequent samples is typically the better choice as creating and clustering smaller samples grows linearly in complexity.

Stability and Certainty of User Roles Focusing more on specific use roles as the general evaluation of the influence of sample size and number samples showed insights on the whole data set, user roles including *Spammer*, *Loners*, *Commentators* or *Daily Chatter* (cp. Fig. 5.14a) have both higher values for users who occur once having a relative majority (yellow bar) as well as occurring multiple times having a relative majority (dark orange bar). In contrast, stable roles such as *Stars* or *Semi Stars* (cp. Fig. 5.14b), users with absolute majorities both for users occurring once (green bar) and multiple times (light orange bar) are dominating. Considering all those users, who have only a relative majority from Fig. 5.13a and 5.15b, the gap to an absolute majority is relatively close for both the *Olympics* and *Super Bowl* data set. It becomes closer for a growing number of samples as Fig. 5.15a reveals high percentages for most users as the green bar representing a relative majority with values beyond 40% is dominating. Also, for specific roles such as the *Daily Chatter* (cp. Fig. 5.14c) and the *Semi Star* 5.14d, the number of users who have a relative majority have only a close gap, which stays almost stable for a growing number of samples. Focusing only on users who occur once and have a relative majority, e.g., the *Daily Chatter* (cp. Fig. 5.15c), for a growing number of samples, the gap to an absolute majority is also getting closer as the green bar is growing. Thus, for most roles, the percentage for the most decisive role is between 40 and 50%, which is also a very persuasive value in a 12-class classification problem, showing that most of the user roles are stable or very close to stable roles. Moreover, the fact that for a growing number of samples, the probability of capturing a user more than once is also growing, especially for 30 or 40 samples 25% respectively, almost 40 % of users are captured at least three times, improving and stabilizing the user probabilities as Fig. 5.16a reveals.



(a) Daily Chatter.

(b) Semi Star.

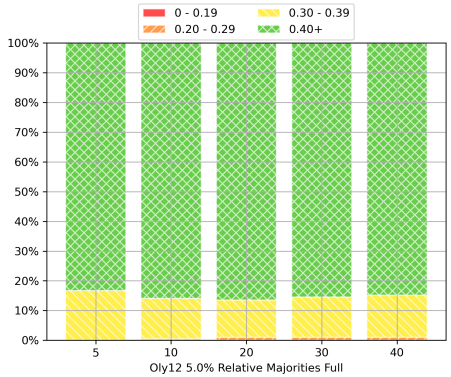
(c) Rel. majority D. Chatter.

(d) Rel. majority Semi Star.

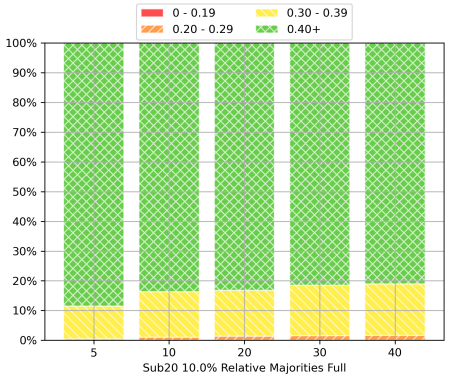
Figure 5.14: Coverage in Random Sampling for user roles of Oly12 5% samples.

Correlation of User Roles As there is also a small number of users who are not stable at all, the user role probabilities are investigated more in detail to visualize the belongings to multiple user roles. In particular, the distances from best to second-best user roles will be examined to show the correlation in terms of originating from the same generalized user role. The distance between the best and second-best user role is generally low, substantiating the significance of second-best ones in the proposed *Multi-Sampling and Combination Strategy*. In Fig. 5.15d, the distance amounts between best and second best roles for all users in the *Olympics 2012* data set with 5% reveals a rather high amount (over 60%) of users who have only a lower distance until ten percentage points the best role for all given samples. As the best role

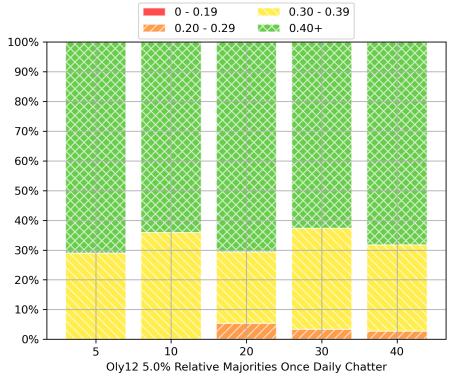
usually dominates with an absolute or close to an absolute majority, an immensely strong second-best role exists for multiple users in the data set. The matrix in Fig. 5.16b shows the correlation of second-best to best roles and reveals that second-best roles principally originate from same generalized user roles compared to the best role. Further, increasing the number of samples does not significantly decrease the share of those users, indicating that these are not artifacts of sampling. Fig. 5.16a shows the number of distinct users for a growing number of samples and reveals an amount of 40% for users who occur three times when combining 40 samples. Overall, the scaling works well, thus validating the approach.



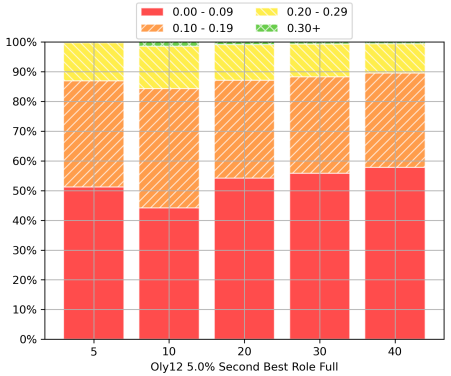
(a) *Oly12* 5% Rel. majority full.



(b) *Super Bowl 2020* 20%.

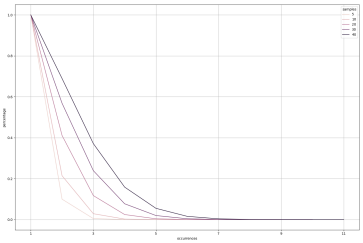


(c) *Oly12* 5% Once rel. majority D. Chatter.

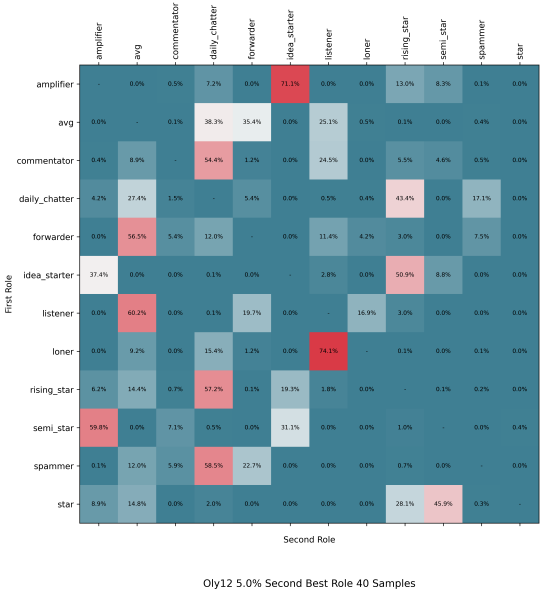


(d) *Oly12* 5% second best roles amounts.

Figure 5.15: Detailed coverage analysis for random sampled user roles.



(a) *Oly12* 5% samples user amount.



(b) *Oly12* 5% samples second best roles matrix.

Figure 5.16: User amount & second-best roles for 5% random samples of Oly12.

Summary The evaluation of the clustered and labeled samples (in total 507 clusters) with the classifiers mentioned in Section 5.4.4 achieved suitable results, as the leftmost data points in Fig. 5.24b show. The substantial variance in the feature distribution present in the boxplots (Fig. 5.6) also shows why training and classifying individual users instead of clusters yields inferior results. Using the *Multi-Sampling and Combination Strategy*, the effect of mislabeling or misclassification is dampened by the fact that the first role is - in most cases - very dominant or - in cases of no majority - either has a significant distance to the second-best role or stems from the same generalized user role. The results demonstrate the effectiveness of both clustering and classification methods. While *expert knowledge* is required to interpret the dendrogram and assign roles, this knowledge can easily be applied to new data sets and their respective clusters. Furthermore, the questions introduced at the beginning of this chapter showed that both *sample size* as well as the *number of samples* influence how fast a whole data set can be covered, depending on the given resources, delivering stable and certain user roles, which impress largely by an absolute majority considering one user role and simultaneously have a high distance to the second best user role. Decreasing user roles shows a *correlation between best and second-best user roles* stemming mainly

from the same generalized user role. Further experiments dealing with even *smaller sample sizes* revealed only conditional suitable results, as a *coverage* of the whole data set is hardly reached with the random sampling strategy. Thus, vast data sets outline a tough challenge for this approach, even though creating a lot of small samples influences the complexity of clustering only linearly. In these cases, splitting the data set into temporary slices, e.g., group stage and knockout stage for data sets dealing with football or week-wise splits for Olympics data sets, would be suitable.

5.5.3 Tuning the Sampling Strategy

The experiments considering user role stability and certainty in the previous Section 5.5.2 revealed the benefits of the proposed novel *Multi-Sampling and Combination Strategy*-based approach. The results showed that samples should be chosen as small as possible due to the quadratic complexity of clustering while creating more samples results in linear growth of running time and memory consumption. Moreover, a sweet spot for the number of randomly created samples was found for the *Olympics 2012* data set at around 30-40 5% samples, which cover around 115K users. For other data sets, the number of covered users was chosen similarly due to the comparability of tuning operations of sampling strategies on several data sets (cp. Table 5.10). Nevertheless, the *random sampling*-based strategy has some issues, such as rarely reaching the full coverage of users. Some users were considered three times or more, leading to oversampling effects, while others were not considered, resulting in a weaker probability support. Reaching full coverage would need many more samples or even bigger ones, degrading the *random sampling* due to an overflowing usage of resources because of the quadratic complexity of clustering. As the kind of sampling is the most straightforward and thus expendable step in the whole strategy, several strategies will now be evaluated against the random sampling strategy, which will serve as the baseline strategy in this section. The following inquests, also discussed in Section 5.5.2, will be discussed again in this section to validate the success of the sampling strategies.

- Are sample sizes limited due to resource costs and representativity?
- How do the number of samples and the sample size affect the coverage of users in the sampling strategies?
- Which impact do the sampling strategies have on the stability and certainty of users and user roles, w.r.t absolute and relative majorities considering their probabilities?
- Does sampling size and the number of samples influence the occurrence and interference of oversampling effects and probability support?

Finding the most suitable *strategy* and the *sweet spots* in dependency of both the *sample size* and the number of *considered samples* is the primary goal of this chapter. However, the comparability has to be guaranteed. Besides the amount considering user frequencies, the optimal point of the baseline with 40 10% samples revealed the aptitude as the best approach, as both coverage and distribution of users were reliable.

Table 5.10: Sample sizes of several data sets for optimization of sampling strategies. Olympic Games: Oly, Super Bowl: SB, Paris: Par, Berlin: Ber

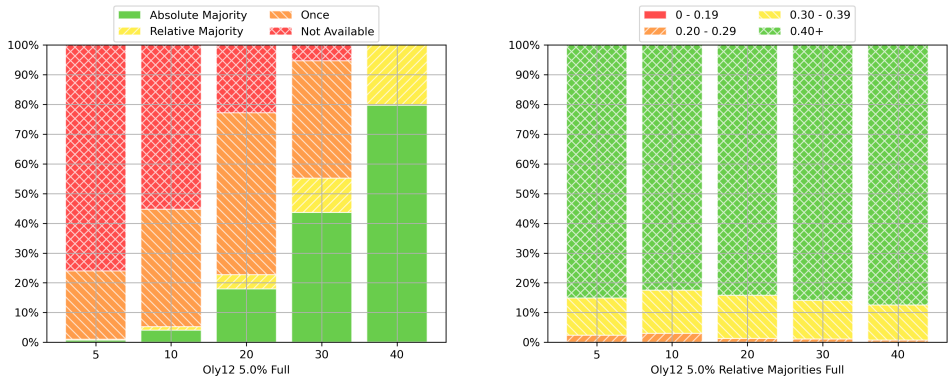
	Data Set					
	Oly12	**Oly14**	**Oly16**	**Oly20**	**Oly22**	-
Small Sample	5%	5%	3%	2.5%	4%	-
Big Sample	10%	10%	5%	3.5%	-	-
	SB13	**SB20**	**SB21**	**SB22**	**Par15**	**Ber16**
Small Sample	15%	10%	10%	10%	20%	10%
Big Sample	30%	20%	20%	-	-	-

Limitations of Sample Sizes When talking about efficiency and resources, the size of samples has to be chosen sagely as on the one side of the spectrum; samples have to be as small as possible to describe a representativity of the whole data set. In contrast, on the other side of the spectrum, the upper bound for sample sizes is determined by the performance of the machines. The extensive sample sizes from Table 5.10 represent the *maximum* that could be clustered on the machines available on an 8-core partition of an AMD Epyc 7401. A small data set like *Berlin 2016* may still be clustered entirely, yet a sample can be generated almost instantly, as seen in Table 5.11. Complete clustering is impossible for large data sets, while samples fit well. The cost is almost entirely consumed by creating the *linkage matrix*, so refinement and exploration steps are interactive in all variants. Pointing now to the *Cluster Expansion* from Section 4.3.6, this strategy comes along with a higher but linear growing runtime when expanding a more extensive set of not observed users, which emerges mainly at a low number of considered samples. Thus, it is a suitable strategy that can be deployed at almost every stage of the combinations of samples.

Table 5.11: Clustering runtime & memory of samples, full data, and approximated(*).

	Oly12 5%	**Oly12 10%**	**Oly12 100%**	**Ber16 10%**	**Ber16 100%**
runtime	19 min	136 min	226 h*	10s	38 min
memory	94 GB	375 GB	375 TB*	1.2GB	184 GB

Influence of Sampling on User Coverage As already introduced in Section 4.3, several promising sampling strategies were discussed due to their benefits and drawbacks, improving the standard random sampling strategy to gain better results such as full coverage and stable and characteristic user roles. All strategies were evaluated considering the *coverage*, i.e., covering enough users with an appropriate number of samples and observing the impact of *oversampling* and *probability support*. As most strategies may reach a point of saturation considering the addition of totally new users when adding new samples, the combination of sampling with an expansion of users, who still need to be incorporated, is also considered. As an expansion does not need the whole process of clustering and classification, *runtime* and *memory* resources can be economized. Outgoing from the sweet spots of saturation when combining the clustered and classified users from the samples, the cluster expansion should not distort role distributions. To prove all of the strategies regarding randomness, the *random sampling* deals as *baseline* to evaluate the benefits and drawbacks by analyzing expected user frequencies against the achieved ones from the strategies.



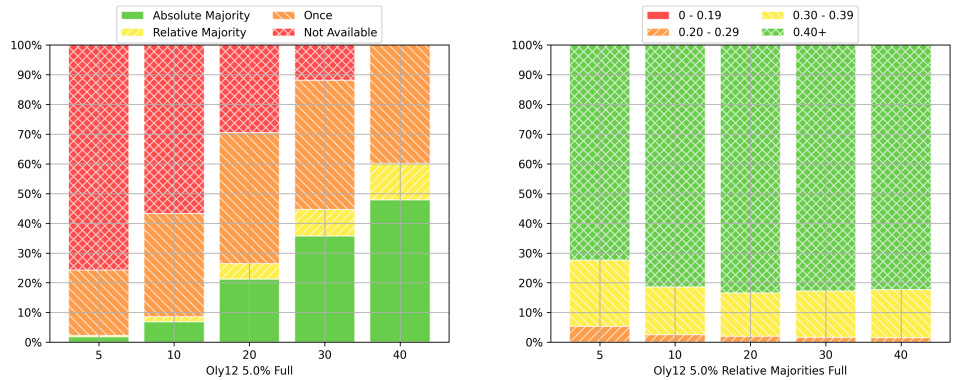
(a) Coverage 5% samples Oly12.

(b) Relative majorities 5% samples Oly12.

Figure 5.17: Coverage comparison for Systematic Random Sampling.

Focusing on reaching full coverage, depending on the sample size, several samples are needed for *Systematic Random Sampling*, introduced in Section 4.3.3 and *Quota Sampling* (cp. Section 4.3.5, which is also dependent on the quota, i.e., users in the samples, which were already considered. In theory, for 5% samples, the *Systematic Random Sampling* would need 20 samples to reach full coverage, while *Quota(50)*, resp. *Quota(25)* would need 40, resp. 80 samples. *Linear Sample Expansion* (cp. Section 4.3.2) does not need to be validated, as the required scope is achieved for each desired number of samples in the implementation. Comparing these strategies to the

baseline of the *Random Sampling* with 5% samples from the *Olympics 2012* data set in Fig. 5.13a the *Systematic Random Sampling* (cp. Fig 5.17a) reaches a full coverage at 40 samples instead of 20 estimated samples. This issue occurs due to creating more samples than needed; in theory, one would expect an ordered combination of the samples, like in the creation process, while in reality, the constellation and the order of sample fragments are shuffled during the clustering process. Thus, some users may be considered twice in an early stage of the combination, while some users may be considered first in a later stage after combining 20 samples in the combination process. Compared to the *baseline*, *Systematic Random Sampling* has an improved coverage to *Random Sampling*, reaching a stable coverage between 30 and 40 samples, while the baseline approach needs between 40 and 60 samples.



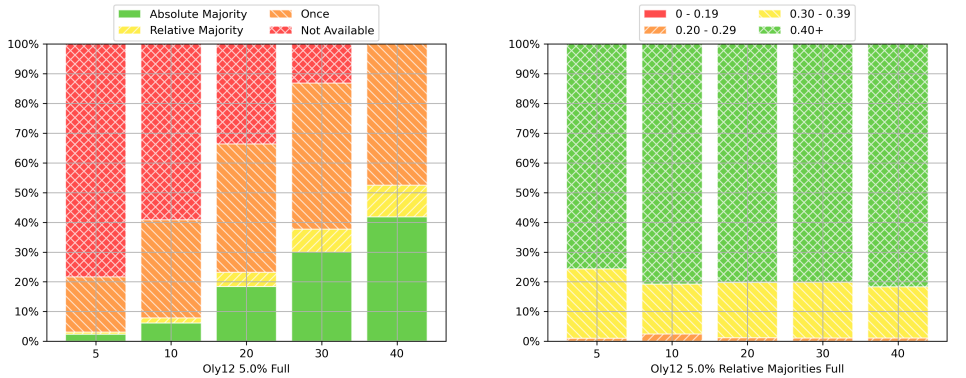
(a) Coverage 5% samples Oly12.

(b) Relative majorities 5% samples Oly12.

Figure 5.18: Coverage comparison of Linear Sample Expansion.

A minor improvement compared to the baseline can also be noticed in the Linear Sample Expansion in Fig. 5.18a as a strategic expansion is more suitable than doubling the 5% samples but is not able to compete with *Systematic Random Sampling* or *Quota*, as the role distributions are not as precise when comparing the distributions to the *baseline* approach using 40 10% randomly created samples. Pointing to the *Quota(50)* strategy (cp. Fig. 5.19a), the estimated number of samples for full coverage is as expected. Nevertheless, the gap between users occurring multiple times in the baseline and *Quota(50)* is almost hardly noticeable. In contrast, the users who occur only once have a relatively small gap to probabilities representing an absolute majority. All in all, the three strategies can cover all users in an adequate number of samples. However, only *Systematic Random Sampling* and *Quota(50)* Sampling are competitive candidates for further investigation, while *Linear Sample Expansion* also has the

drawback of a longer running time, as the samples are enriched with already covered data. *Stratified Random Sampling* from Section 4.3.4 may be a good candidate, too, but was outperformed by *Quota* in all aspects and will not be considered further.



(a) Coverage 5% samples Oly12.

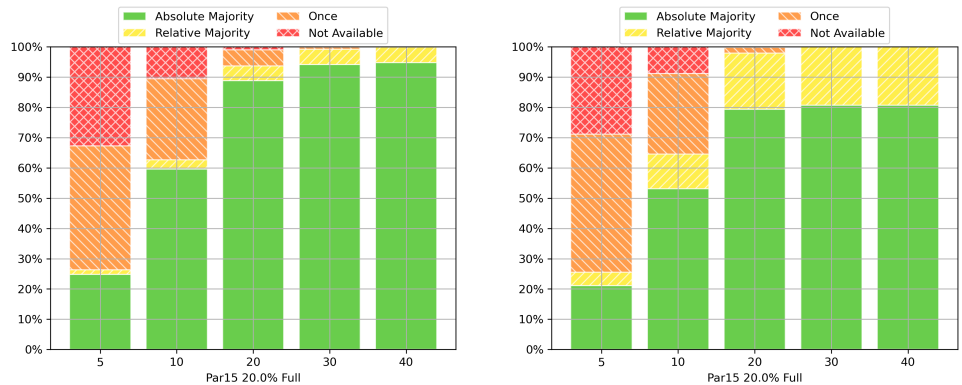
(b) Relative majorities 5% samples Oly12.

Figure 5.19: Coverage comparison for Quota(50) Sampling.

Influence of Sampling on Stability and Certainty In this paragraph users with only a relative majority for the *Systematic Random Sampling* (cp. Fig. 5.17b), *Linear Sample Expansion* (cp. Fig. 5.18b) and *Quota(50)* (cp. Fig. 5.19b) will be investigated. Starting with the *Systematic Random Sampling*, one can notice a high degree of users getting captured only once until 30 samples, as well as a growing number of users having a relative majority, especially between 30 and 40 combined clusters. Even though a higher number of users do not have an absolute majority, over 80 % of those users are close to an absolute majority in each step of combining up to 40 samples. The fact that 80 % of users in the *Systematic Random Sampling* reach an absolute majority for one best user role underpins the suitability of this approach in terms of stability and certainty of user roles. A similar effect for users having a relative majority can also be noticed for the *Linear Sample Expansion*. In contrast, the number of users captured only once is constantly high over all samples. *Systematic Random Sampling* outperforms *Linear Sample Expansion* when focusing on users with only a relative majority in terms of closeness to an absolute majority. Moreover, *Systematic Random Sampling* also dominates the other techniques in the number of users having an absolute majority, reinforcing the merits of Systematic Random Sampling made in the previous paragraph of the coverage. *Quota(50)* is comparable to the *Linear Sample Expansion* when focusing on coverage, even though

the users who got captured once have a higher amount. Having a closer look at users with only a relative majority *Quota(50)* is also an adequate strategy. A similar number of users has a close gap to the absolute majority compared to the *Linear Sample Expansion*. Finally, all three strategies reach well-pronounced user roles regarding stability and certainty. *Systematic Random Sampling* and *Quota(50)* show the best suitability when combining the benefits of stability and certainty with the coverage from the previous paragraph.

Influence of Sampling on Oversampling and Probability Support Focusing again on the insights from the strategies, the sample size also plays an important role when reaching full coverage. If data sets with a nominal size are considered, the sample size has to be adjusted to guarantee representative samples. However, the possibility of oversampling is growing when considering a lot of more extensive samples. This effect of oversampling can be seen in Fig. 5.20a and 5.20b for the *Paris 2015* data set with 20% samples for *Quota(50)* and *Systematic Random Sampling*, where saturation of covered users is noticeable between 20 and 30 samples for both strategies. Most users occur more than once and have a stable absolute majority. In contrast, the users with relative majorities have a close gap to an absolute majority, considering the probabilities of the best user role. Thus, there is no adverse effect of oversampling noticeable, making both strategies valuable.

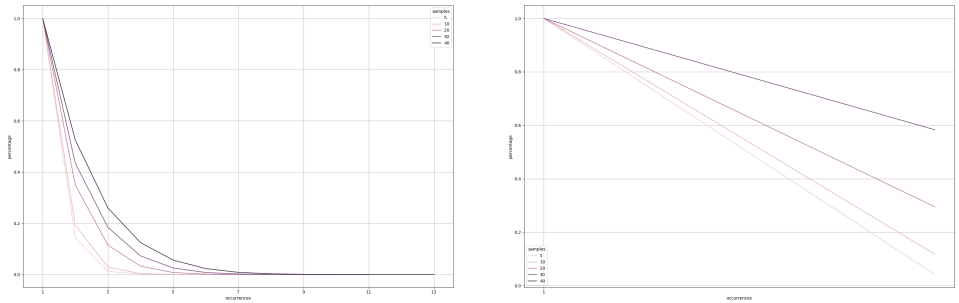


(a) *Paris 2015* 20% samples Quota(50). (b) *Paris 2015* 20% samples Systematic.

Figure 5.20: Coverage comparison: Quota vs. Systematic Sampling.

Compared to the *baseline*, there were several effects of oversampling noticeable, especially for specific user roles, where probabilities of relative majorities deteriorated for a growing number of samples. Nevertheless, in most cases, one must also consider that the number of users who only have a relative majority is decreasing for a growing number of samples. Thus, those plots must be treated with caution, as those amount of user roles with only relative majorities must be placed concerning all remaining users when analyzing the growing number of samples.

Further Observations Thus, depending on the strategies, the shuffling effect while the clustering process also has consequences on the user amounts, i.e., how often users are considered when analyzing the number of observed samples. For a growing number of samples, the *baseline* approach in Fig. 5.16a receives higher probabilities for users covered at least 3 or 4 times. At the same time, the *baseline* receives values of almost around 40%, resp. 20% for 40 samples, while strategies that are not wholly randomized do not show aspects of varying frequencies. The *Quota* strategy (cp. 5.21a) has a relatively similar graph compared to the baseline but does not have as high accumulations of users who occur 3 or 4 times. In contrast, the *Systematic Random Sampling* in Fig. 5.21b strategy reaches only users who occur twice, as the whole data set is partitioned to cover it precisely two times.



(a) Quota(50) Sampling. (b) Systematic Sampling.

Figure 5.21: Comparison of user amounts for 5% samples Oly12.

Detecting the best strategy Table 5.12 shows for 4 data sets with varying sample sizes(columns) as well as a growing number of combined samples(rows) the best sampling strategies considering their closeness of aggregated mean role distributions to the baseline. Furthermore, the cluster expansion and the number of users added at the specific stage of combining samples were scrutinized for each strategy. While *Quota(50)* works well for mostly fewer amounts of combined samples, especially for

data sets with small sample sizes or small absolute numbers of considered users, both for a growing number of considered and combined samples, as well as a growing number of sample size the *Systematic Random Sampling* (Syst.) outperforms *Quota(50)* in stages where more samples are combined, or the sample size is chosen higher. The number of added users with *Cluster Expansion* is relatively high when combining smaller samples; a good trade-off can be reached when a saturation effect of stable user roles is reached. For smaller sample sizes, e.g., *Olympics 2012* 5%, this step is reached way later than for bigger sample sizes such as 10% or even 20 % samples. A striking observation is that *Cluster Expansion* (marked bold) mostly only provides an added value to the *Systematic Random Sampling* considering aggregated mean role distributions to the baseline.

Table 5.12: Most suitable sampling strategies for specific data sets w.r.t sample sizes & number of samples (left). Amount of users added by Cluster Expansion (right).

		Number of Samples									
		5		**10**		**20**		**30**		**40**	
Data Sets	**Oly12 5%**	Quota	77%	Quota	60%	Quota	36%	Quota	21%	**Syst.**	13%
	Oly12 10%	Quota	59%	Quota	35%	**Syst.**	12%	**Syst.**	4%	**Syst.**	1%
	Ber16 10%	Quota	59%	Quota	35%	Quota	12%	**Quota**	4%	Syst.	2%
	Par15 5%	**Syst.**	38%	**Syst.**	11%	**Syst.**	1%	Syst.	0%	**Syst.**	0%

While the means of user roles are hardly affected by cluster expansion, when considering the mean role drift over the complete data set, individual user roles can significantly be affected, such as the *Star User*, which has a compact constellation of fewer users and is well separated to other user roles. Users from the roles of *Commentators*, *Daily Chatters*, *Semi Stars*, and *Spammers* commonly have a higher distance to their second-best user role. Thus, they are not affected as Star Users by the *Cluster Expansion*. All in all, the distinct user roles are expanded almost uniformly.

To sum up, the analysis of various sampling strategies has shown that *Quota(50)* and *Systematic Random Sampling* are highly advantageous for enhancing the coverage and certainty of user roles compared to *Random Sampling*, which serves as the baseline. These options have no adverse effects, such as oversampling or insufficient probability support. While *Quota* works well for smaller sample sizes and a smaller amount of combined samples, *Systematic Sampling* has its benefits and vice versa. Although both strategies reach full coverage as well as certainty faster than Random Sampling, *Cluster Expansion* is a valuable extension, as fewer samples need to be created, clustered, and classified because even for a smaller amount of combined samples, *Cluster Expansion* delivers steady results without negatively affecting the enriched cluster means in terms of feature drifts. To pick out a strategy that works well for all kinds of data sets

due to a fair and beneficial comparison in time series, *Systematic Random Sampling* delivered persuasive results for all data sets. In contrast, *Linear Cluster Expansion* only worked well for earlier data sets, such as the *Olympics 2012* data set.

5.6 Multiple Individual Data Sets

While the first step of the experiments and analysis in Section 5.5.2 dealt with the initial application of the approach from Section 1.2 on single data sets, the second step focuses on analyzing several data sets individually with the given approach to understand if it is more widely applicable individually over a wide variety of data sets stemming from the same social network, as claimed in the contributions of Section 1.3. Besides the question of the applicability of the approach, several other questions dealing with user roles and their evolution will be discussed in this section:

- Is the approach applicable to a wide variety of data sets?
- Do the same or similar user roles reappear among topically similar data sets?
- Do user roles evolve over time w.r.t. quotas?
- Do user roles change over time w.r.t. feature drifts?

Applicability of the Approach Applying the approach to new data sets means applying all steps mentioned in Section 4.2 in principle, such as *Preprocessing*, *Feature Engineering*, *Clustering*, *Cluster Analysis*, *Classification*, and *Sampling*, but with significant conceptual and practical simplifications and savings. When dealing with multiple data sets stemming from the same social media, the preprocessing and feature engineering methods will be the same for all sets. Additionally, the hierarchical agglomerative clustering using Ward's method produced satisfactory results for each data set and will not be impacted. The clustering cutoff as part of the cluster analysis may need to be adjusted slightly to ensure accuracy, as the suggested approach exploiting a feature comparison based on pairwise effect size within the depth-first search may yield varying levels of significance. In addition, a suitable significance criterion for aborting the *depth-first search* has to be found which can deal with all data sets. With the aid of the tool presented in Fig. 5.9, insights over all data sets delivered the following criteria for the significance changes of *effect size* for the approach from Definition 33: At least two large effects, one very large effect or one huge effect delivered suitable results for all clusters. In addition, the difference between the average *Cohen's d* overall features between parent and child nodes was determined. Another criterion for significance changes was found if this distance is greater than

0.1. Thus, not only are deviations in a few features decisive for significance changes, but minor changes in overall features are also considered, capturing clusters more precisely. The only time-consuming step is the building of new training data covering a plethora of different aspects such as type of events, e.g., sports, politics, or tragic incidences, the recording period, e.g., only a few days vs. several weeks or the point of time, as social networks and their users may develop over time considering their behavior. Considering the building process of new training data, the analyst can resort to well-elaborated user roles to cut the process of creating new training data sets short.

Reappearance of User Roles After addressing the general application of the approach on various disconnected data sets, the applicability will witness if the same or similar roles are present in data sets varying in time and topic and how they evolve over time. While the focus in Section 5.5.2 was on coverage as well as certainty and stability of user roles in general, the focus in this section is to analyze user roles more widely in terms of possible feature drifts leading to slightly different characterizations of user roles beyond data sets. As all features in each data set have the same bounds after normalization and standardization, a more in-depth analysis considering user role drifts is possible. This aspect is consequential, as data sets are preprocessed individually, i.e., the normalization and standardization are applied for each disconnected data set individually. Thus, feature stability from one to other data sets over chronological time is also a significant position aiding a robust recognition of user roles, even though slight feature deviations cannot be prevented as users and their written messages in the data sets may differ after preprocessing. Nevertheless, the stability of user roles described by features is an influential significance criterion for role recognition. Moreover, user role evolution, arising from possibly noteworthy feature drifts, i.e., changes in user role distributions, is another beneficial area that will be discussed in this section.

As already mentioned in Section 5.4.3, which is the basis for building classifiers, most observations of both generalized and specified fine-grained roles were observed in most samples as well as in other data sets (cp. Table 5.1), so the whole manual role assignment process consisting of evaluating dendrograms and the deviations of single features in clusters, represented by boxplots, was applied across those data sets by comparing and evaluating clusters against each other and finding correlations between labeled roles. Some roles could be identified more easily as matching descriptions from literature are present. In contrast, other roles need more manual evaluation, as roles from the literature do not match perfectly, or user roles may differ from roles detected in previously analyzed data sets. Tracking user roles across data sets with varying topical aspects as well as time deviations manifested concept shifts and drifts among them, as the frequency and probability of user roles or feature distributions were revealed using PCA or LDA, as these strategies facilitate the comparison of similar

cluster-means due to fewer dimensions to observe. Considering all data sets, at least 10-15 distinct candidate classes could be observed, showing up in varying frequency across the data sets but also disappearing completely in specific data sets, as a topical variation is noticeable or trends over time changed.

The *12 user roles* identified on the *Olympics 2012* data set are also *present* and *well-separated* in the other data sets, though -as the rightmost column of Table 5.6 shows- the frequency for the training data (in percent) varies over data sets and over time. In addition, the experiments from Section 5.5.2 yielded more precise frequencies for each user role over all Olympic data sets, proving most of the frequencies and trends from the training data sets, which can be seen in Table 5.13.

User Role Quota Changes Upon conducting a thorough analysis of the optimal methods for sampling and combining in the last Section, Table 5.13 provides a clear representation of the distribution of user roles examined with the *Systematic Random Sampling* strategy for all *Olympics* data sets between 2012 and 2022. While the top 5 roles stay constant, w.r.t. their position in each data set, *Forwarders* become more present after the *Olympics 2016*. Almost all other user roles have a decreasing amount of users over time. Especially *Average Users*, *Listeners*, *Daily Chatter*, and *Rising Stars* forfeit users. At the same time, *Commentators*, *Semi Stars*, and *Stars*, all roles with a smaller number of users, stay almost stable or increase their number of users.

Table 5.13: User role distribution for Olympics. Position of role in data set in brackets.

	Oly12		**Oly14**		**Oly16**		**Oly20**		**Oly22**		**Trend**
Forwarder	31.73%	(1)	38.55%	(1)	55.41%	(1)	58.12%	(1)	53.88%	(1)	↑
AVG	28.73%	(2)	22.89%	(2)	18.98%	(2)	19.44%	(2)	21.96%	(2)	↓
Listener	12.53%	(3)	13.70%	(3)	9.08%	(3)	8.07%	(3)	8.42%	(3)	↘
Daily Chatter	8.78%	(4)	9.74%	(4)	6.11%	(4)	5.94%	(4)	5.02%	(4)	↘
Rising Star	5.08%	(5)	2.31%	(8)	2.47%	(6)	1.71%	(6)	1.42%	(9)	↘
Idea Starter	3.62%	(6)	2.45%	(7)	1.83%	(7)	1.33%	(7)	1.61%	(7)	↘
Amplifier	3.58%	(7)	2.62%	(6)	1.36%	(8)	1.29%	(8)	2.54%	(5)	↔
Spammer	2.56%	(8)	4.71%	(5)	2.95%	(5)	1.80%	(5)	0.60%	(11)	↘
Loner	1.17%	(9)	0.31%	(12)	0.17%	(12)	0.13%	(12)	0.73%	(10)	↔
Commentator	1.12%	(10)	1.19%	(9)	0.37%	(11)	0.67%	(10)	1.71%	(6)	↗
Semi Star	0.91%	(11)	0.90%	(10)	0.90%	(9)	1.05%	(9)	1.56%	(8)	↗
Star	0.19%	(12)	0.60%	(11)	0.38%	(10)	0.44%	(11)	0.55%	(12)	↗

The data sets for the *2014 Olympics* (278 clusters) and the *2014 FIFA World Cup* (193 clusters) show that there are minimal changes within close time periods in topically related data sets considering sports events. However, the occurrence of *Average User* and *Loner* decrease, while *Forwarder* and *Listener* appear more frequently.

User Role Feature Drifts The first significant feature changes occurred in the *Olympics 2016* data set (355 clusters), while changes in user role quotas were noticeable. The PCA in Fig. 5.22, showing cluster centroids of the training data, provides a salient concept drift for many user roles between the *Olympic 2012* data set (pipe symbol |) and the *Olympic 2016* data set (crosses X). In particular, *Semi Stars* also tend to cover a space much closer to *Stars*, as the `verified` status was more freely distributed by Twitter. There is a noticeable trend among users to retweet content instead of generating original material. This trend has resulted in a rise in the frequency of both *Average Users Loners* and *Forwarders*. This phenomenon has been recognized and is ongoing. The reasons for those drifts can be traced back to changing user behavior in different events, resulting in alternating user features. However, meaningful changes in the social network, e.g., the allocation of the verified status reserved only for star users before 2012 to a broader group of users, are proven reasons for drifts.

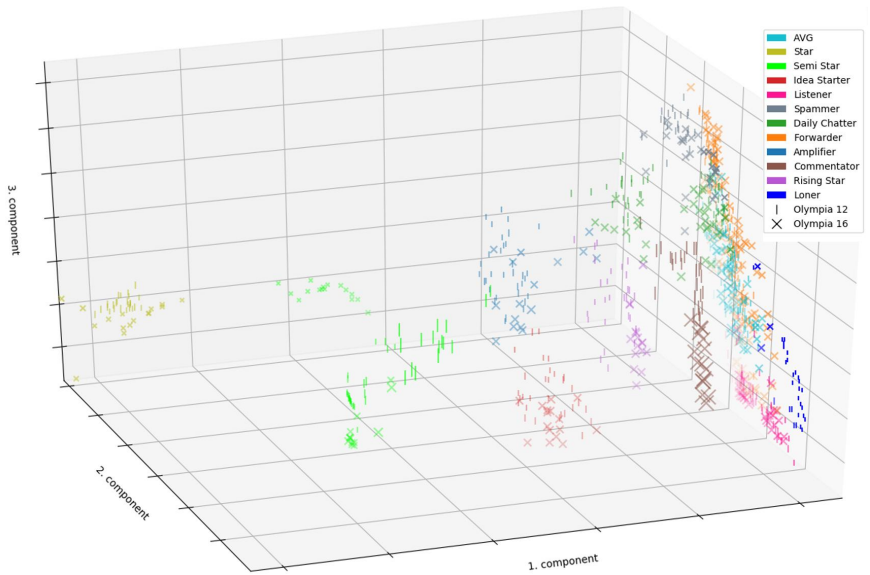
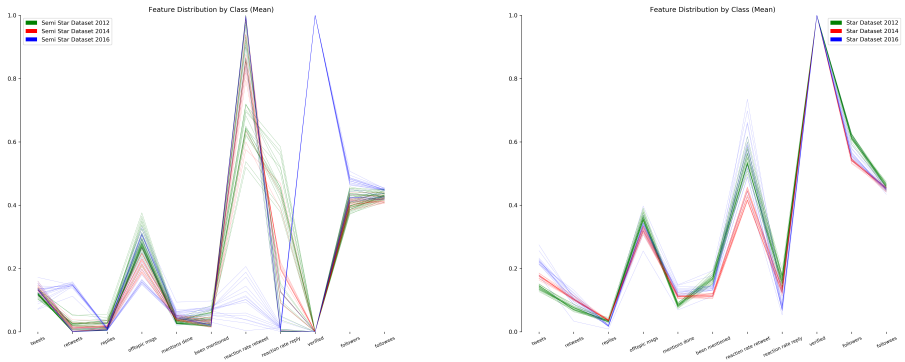


Figure 5.22: PCA of clustered samples from *Olympics 2012* vs. *2016*.

Feature deviations are possible reasons for user role drifts and shifts already discussed based on the PCA. Nevertheless, looking into dimension-reduced plots, single feature deviations cannot be found. Thus, line plots manifest those changes better as the line plots of *Semi Star* Users (cp. Fig. 5.23a) as well as *Stars* (cp. Fig. 5.23b) show. While the *Star* user stays mostly stable w.r.t. feature changes in all three data sets,

the Semi Star has several considerable deviations between the *Olympics 2012/2014* and *2016* data sets, as the blue line reveals. The PCA plot clearly reflects the latest feature deviations, particularly in the `verified` status, `retweets`, and the `reaction rate of retweets`. Much more difficult to prove is the influence of normalization and standardization on feature deviations. As preprocessing is an essential step for clustering, sacrificing normalization and standardization is no option. Moreover, restarting the whole pipeline for each new data set forcing a global normalization is also pointless due to clustering the whole data sets again. Thus, the analyst should carefully inspect line plots to observe feature deviations. Despite minor changes in a few features, the user roles in the training data sets remained remarkably consistent. However, there were a few exceptions, such as the Semi Star.



(a) Lineplot describing features of *Semi Star*. (b) Lineplot describing features of *Star*.

Figure 5.23: Line plots for user roles of *Olympics 2012-2016* training data sets.

The primary user role shift trend continues for the *Super Bowl 2020* (345 clusters) data set, which is otherwise, despite the different sports and the time difference, somewhat similar to *Olympics 2016*. The *2015 Paris Attacks* (160 clusters) data set covers a very different topic and distinct interactions as there are fewer `offtopic messages` and more `retweets`. Some user roles are not present, such as *Commentator* or *Loner*, yet most of the overall trends match the picture of the sports events: forwarding instead of content creation becomes more dominant both as features and as roles, corresponding to the broader trend in all social media. In turn, influencer roles become pronounced, where the *Semi Star* may have to split into two separate sub-roles.

Summary The only exception where the approach could not be applied was the random *Twitter Sample Stream*, as features based on topics lose their usefulness. Regarding the initial inquiry, utilizing the same attributes can facilitate the establishment of uniformly recognizable user roles. Significant changes in the distribution of user roles were observed, and interdependent relationships between these roles were discovered, extending beyond the confines of the data sets. However, at this step, the effort of labeling samples of each data set manually is a limiting factor.

All in all, the transferability of the proposed approach is possible with some limitations but delivers convenient results. Both clustering and classification work well as the user roles are reoccurring in each data set with slightly different characterizations and varying portions, which lies in the nature of the evolution of social networks w.r.t. different user behavior, interaction, and social trends. Focusing on the varying trends of user roles over time delivered by the *Olympic Games* data sets from Table 5.13, a similar development of user roles can be observed in the *Super Bowl* data sets, even though the data sets have entirely different properties considering the recorded period and thus the number of messages and users. For all data sets beginning with those from years ago, e.g., *Olympics 2012* or *Super Bowl 2013*, to the most recent, such as *Olympic Games 2022* or *Super Bowl 2023*, the evolution of both feature drifts and user role shifts is conspicuous.

5.7 Applying Models to New Data Sets

The previous section indicated that applying the approach to new data sets from the same social network is possible but needs many resources, especially in the manual labeling process and the building and optimization of classifiers. So the third step is targeting the question from Section 1.3 if gathered knowledge on user roles from well-understood data sets can be transferred to new data sets.

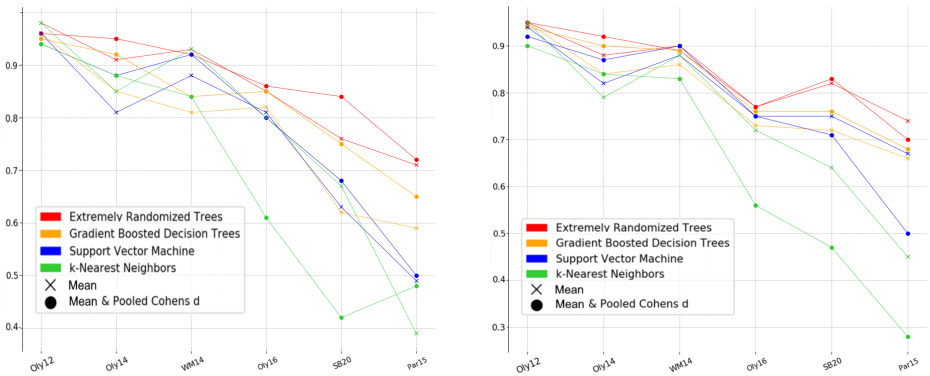
To show this aspect, the following questions will be answered in this section:

- Is a transfer of role knowledge possible by assessing the quality and effort involved?
- How do feature variation and drift impact the process of transfer knowledge w.r.t limitations?

Fig. 5.24a and 5.24b show the *F1 scores* when classifying the data sets based on the *Olympics 2012* as the reference data set, as it provides the most prolonged prediction period. While the *weighted values* in Fig. 5.24b depict the quality of frequently represented user roles, the *macro values* in Fig. 5.24a support the overall performance

of the classifiers. Overall, one can see a gradual degradation over time in the sports events, as the classification methods do not explicitly capture the drifts observed in the previous section but still generalize the roles over time. Nevertheless, the best methods achieve a 0.85 F1 score for late sports events. The *2015 Paris Attacks* data set sees the most extensive degradation, showing topic and interaction differences have a more profound impact than time. When comparing all these results to the slightly worse macro values, one can see that small groups are captured well, while larger clusters tend to be somewhat "*blurry*".

kNN and *SVC* keep up well for short intervals but tend to lose ground on longer distances. *ET* holds a slight edge over *GBM*, while the latter stays competitive and incurs much lower runtime costs. Both strategies benefit from enriching the data sets with the *pooled Cohen's d* values.

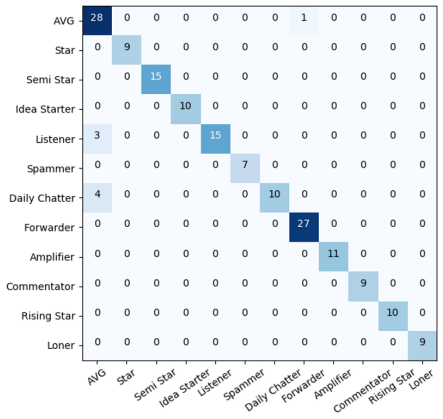


(a) Macro Average F1.

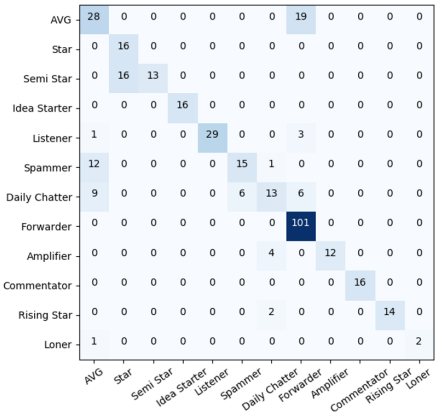
(b) Weighted Average F1.

Figure 5.24: Information retrieval measure F1 for classifiers.

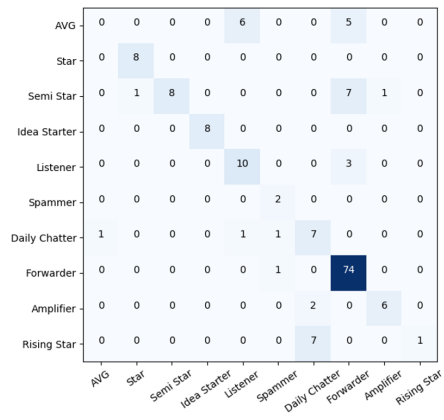
The confusion matrices for the *Olympics 2012* (Fig. 5.25a) *Olympics 2016* (Fig. 5.25b) and *Super Bowl 2020* (Fig. 5.25d) data sets show how roles that were either not well separated in the *Olympics 2012* data or drifted significantly are most affected. However, these misclassifications often lead to adjacent roles, e.g., *Average Users* as *Listener* and *Forwarder*, *Daily Chatter* as *Forwarder* and *Average User* or *Semi Stars* as *Stars*. Specifically, the scores for the *Super Bowl 2020* data set emphasize the drift to forwarding content and the rise of influencers from *Semi Stars* to *Stars*. Only in the *Paris 2015* data set (Fig. 5.25a) some more misclassifications are noticeable due to the topical different training data. Thus, the F1 scores actually understate the quality of the result, as they do not consider the adjacency of roles.



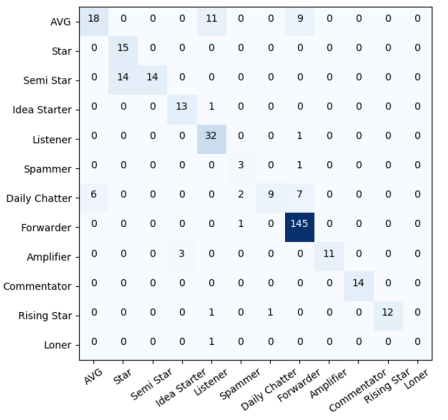
(a) *Olympics 2012*.



(b) *Olympics 2016*.



(c) *Paris 2015*.



(d) *Super Bowl 2020*.

Figure 5.25: Confusion matrices of classifying data sets with Oly12 training data.

The data set of the *2016 Berlin Truck Attack* (Christmas market) that was not evaluated in the previous stages provides topic similarity to *2015 Paris Attacks* while being close to the *Olympics 2016* in time. This data set provides an excellent opportunity to assess the impact of different training data sets: in addition to the baseline of the *Olympics 2012* and close sets (*Olympics 2016*, *2015 Paris Attacks*) as well as the *Super Bowl 2020* data set as a small, recent data set, two combinations of training data were evaluated: *Olympics 2012* and *Super Bowl 2020* for the full-

time range and *2015 Paris Attacks* with those two as a mix of time range and topic proximity. As Table 5.14 shows, these combined data sets provide the best results, matching manual classification or producing misclassifications to close roles. *2015 Paris Attacks* seems too small to provide a sufficiently general model but can boost the full-time range model.

Table 5.14: Classification of the Berlin 2016 data set using several training data sets. Comb1: Oly12 & SB20, Comb2: Oly12 & SB20 & Par15

Classifier	Oly12	Oly16	Par15	SB 20	Comb1	Comb2
XGB	0.58	0.59	0.51	0.70	0.78	0.92
ET	0.74	0.63	0.56	0.73	0.77	0.82

The experiments show that a transfer of labeling knowledge is effective with certain limitations: meaningful topic differences or long-time differentials diminish the usefulness, yet a good choice of reference data can mitigate this effect. Finally, applying classifiers using acquainted training data is possible if a pool of manifold training data is given. Experiments showed that even the classification is suitable primarily for data sets that are topically not related perfectly or have deviations in time.

5.8 Evolution of User Roles Over Time

After optimizing several parts of the approach in Section 5.5 as well as transferring and applying the approach to several data sets in Sections 5.6 and 5.7 the analysis of user roles considering their movement together with their long term evolution will be analyzed and discussed in this Section. In Section 5.8.1, a central aspect is the analysis of user role movement over time, analyzing the distributions of users changing their roles as well as non-observable users and their implying issues in terms of user role movement. In contrast, Section 5.8.2 focuses on whole role chains over a while, generally finding specific patterns occurring multiple times. Both aspects are eminent foundations for building models and simulating existing and completely new data sets, which will be introduced later in Section 5.9.

5.8.1 Analysis of User (Role) Movement

A series of comparable data sets are needed to evaluate and analyze user roles, single users, and their evolution and movement over time, i.e., data sets with a similar topic focus, such as the *Olympics* or *Super Bowl*. For each data set, a disconnected analysis

is done, e.g., the user role quotas for individual datasets are examined as in the previous section (cp. Table 5.6). Also, the training data in Table 5.8 and initial trends in user role changes, such as those found in the Olympics datasets shown in Table 5.13 were analyzed and gave valuable insights on behavioral changes of users. These aspects lay the foundation for a more extensive analysis of user (role) movements by chronologically building whole chains between events for single users.

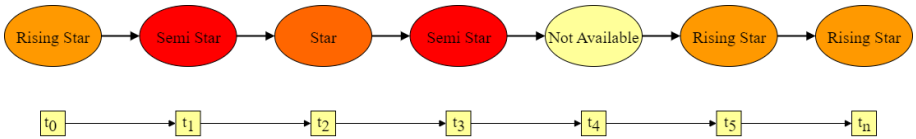


Figure 5.26: Example of a user role chain.

A role-chain can be seen in Fig. 5.26, representing a single user in terms of this work consisting of several user role statuses over time $t_1, ..., t_n$, modeling the user role changes of a user.

Considering those aspects, the following questions are examined in this section:

- Which impact have users who dis- or reappear in data sets?
- How many users change their behavior w.r.t. a specific user role?
- Which amount of users stay in the same (generalized) user role?

When analyzing user role movements across data sets over time, a central point is so-called *non-observable* users, who impede the whole analysis. *Non-observable* users are users who entirely or temporarily disappear and thus cannot obtain a role when absent. While users who reappeared and thus already got allocated to a specific user role in an event before, users who emerged first after the first event in the observed period may be harder to allocate to a role, as they did not deliver any information about their behavior in the social network, yet. Starting from the optimized *Multi-Sampling and Combination Strategy* for each data set using the *Systematic Sampling* strategy, specific observations can be made, which reveal user roles for each user. Thus, pairwise user role transitions between two data sets and even longer chains are possible.

Focusing now on events over a while, such as the *Olympic Games* between *2012* and *2022* as well as the *Super Bowl* between *2013* and *2022*, a lot of user movement can be observed in the data sets pairwise in chronological order as Table 5.15 shows. Only a small number of users in the social network stay in the same role, the same generalized role, or a different generalized role for both time series. For closer gaps between events

and recent events, the number of users who stay is growing for all three mentioned statuses. Moreover, there is a lively exchange considering leaving and re-entering users in both data sets, amplifying the issues of non-observable users. The last row shows the number of unavailable users dependent on all observed users in the *Olympics* and *Super Bowl* data series, calculated by the union of the appropriate data sets.

Table 5.15: User role changes starting from Olympics 2012 and Super Bowl 2013.

	Oly14	**Oly16**	**Oly20**	**Oly22**	**SuB20**	**SuB21**	**SuB22**
Same Role	3.90%	3.07%	4.89%	21.86%	1.04%	8.97%	10.49%
Same Generalized	4.05%	2.54%	3.06%	7.60%	1.20%	4.13%	6.17%
Different Generalized	5.87%	3.30%	3.32%	8.20%	1.42%	4.12%	6.14%
Leaving	88.03%	78.38%	84.35%	82.61%	94.90%	78.48%	74.78%
Returning	-	14.61%	7.32%	5.96%	-	5.90%	9.09%
Not Available	86.18%	91.09%	88.73%	62.34%	96.34%	82.78%	77.20%

It is essential to remember that the distribution of roles in each event, as indicated in Table 5.13, is closely tied to changes in user roles. The top 5 frequently appearing user roles are most susceptible to role switching. Hence, one must be cautious of these patterns when examining the data. While disappearing after an event is quite likely for each role, the changes to a generalized role or a not generalized role balance each other for most of the investigated data sets. When observing the whole chain starting with the *Olympics 2012*, almost every user (99%) who was active in the first data set disappeared at least once. Especially for the *Olympic Games*, which alternate between *Summer* and *Winter Olympics*, individual interest in only winter or summer sports is a possible reason for rejoining but not the only reason, as only 11% of *Summer Olympics* fans from *2012* rejoined in *2016*. The longer the observation period, starting with the *Olympics 2012* as the initial event, the probability of rejoining in *2016* or later users is contracting to 11.2% (2016), 2.8% (*2020*), and 0.5% (*2022*). To sum up these observations, 85% of users who became inactive never returned, while 15% who came back are relevant enough for further investigations. The relatively big gap between *Super Bowl 2013* and *2020* also shows a significant pool of core users (5%) who return in the following events after leaving the social network.

As enough significant users participate in more than one event considering the event series, their user roles, and possible user role changes are of interest. Concentrating only on users who rejoin at least once and thus are observable in at least two events, users who rejoin with at least a similar behavior in a generalized user role are further investigated in both the *Olympics* and *Super Bowl* time series to consider variable gaps between events. The analysis showed that both for short periods of one year as well as more extended periods, the number of users who reappear in another user

role lies between 70% and 75%, which shows that user behavior evolves generally. As each data set was classified with a trained classifier close to the given events, feature drifts of user roles are also considered and substantiate the nature of evolving user behavior. Finally, the observations of role switches in this section show the need to abstract user role changes represented by role chains, displaying eminent patterns for the model-building process. Thus, pairwise transitions of user roles between two data sets will be the central point for the long-term analysis, which will be introduced more in detail in Section 5.8.2.

5.8.2 Long Term Role Chains of User Roles

As user role movement was analyzed in the previous section, the attention will now be drawn more to finding specific patterns in role chains, which result from user role changes across data sets in time series. In related work (cp. Section 5.10.2), such as voter migration in politics and document analysis in medicine, a long-term analysis of existing data is a well-known strategy to enrich several types of models with suitable data. In this work, exploring long-term role chains is quite a powerful way to analyze user behavior over an extended period. A role chain consists of several states representing user roles and their most likely changes across a series of data sets. To handle and specify this case, for each user present in at least 2 data sets, the user roles for each data set are ascertained. The role chains for each role in the data set can be grouped to identify the most frequently occurring user role chains. Experiments showed that many users over all user roles are represented by the same user role chain as Fig. 5.27 reveals, displaying the quotas of the five most frequent role chains for each user role. While the most frequent role chain has an amount of around 25% for *Star* Users, the user roles of *Listeners* and *Loners* are represented by the most frequent chain, around 80%. Overall data sets, most users are covered by the top 5 user role chains. This aspect shows that users are well clustered and classified across all data sets, as they share not only a similar behavior w.r.t to features in single data sets. Of course, the absence of users substantially influences role chains, as the probability of leaving the social network for all roles is relatively high. However, only some roles are likely to leave the social network after being present in one data set as, e.g., *Stars* tend to stay more likely in the data sets than *Loners*. Focusing on the role chains again, *Spammers* and *Star* users tend to have the most extended activity chains in the observed time series. In contrast, users from more passive user roles like *Spammers* or *Loners* tend to leave the social network entirely after the first event they were present. Another worthwhile observation is users' active role chain lengths concerning interests in specific events such as the Summer or Winter Olympics. Investigating only users active in the Summer Olympics, only minor improvements (1-2%) for a more prolonged activity in this time series are noticeable for some roles such as *Semi*

Stars, *Daily Chatters*, or *Forwarder*. At the same time, *Stars*, *Amplifier*, or *Spammer* tend to leave the social network after one event as the number of users deteriorates slightly by 1-2%. Also, the significant gap of 4 years between the *Summer Olympics*, instead of 2 years, affects the behavior negatively, w.r.t. leaving the social network.

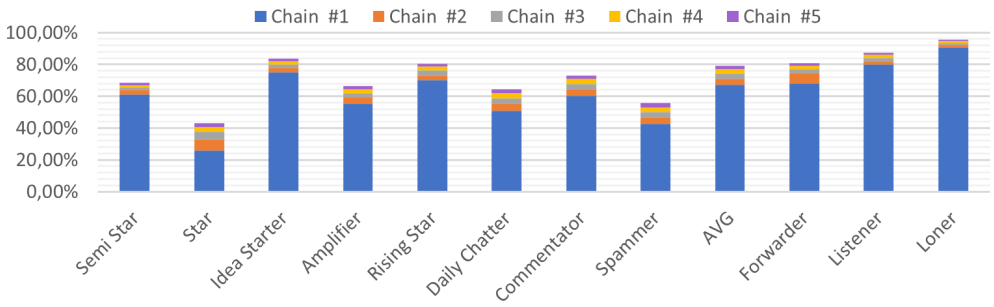


Figure 5.27: Quota of the five most frequent chains per role.

Summarizing the results of user movement, especially leaving and rejoining, is a significant observation that affects almost each user role. Disregarding users who were not available in at least one event can set the focus on user role movement and behavior for specific roles on one side of the spectrum, as active users tend to stay more likely. However, on the other side of the spectrum, it lies in some user roles' nature, such as the passive generalized roles, to leave and rejoin the social network after being absent from an event. Observations for the *Olympic Games* and the *Super Bowl* series showed that user role movement delivers a significant and valuable outcome. Nonetheless, more data sets covering a longer or shorter time are inevitable for more precise long-term trends. As a consequence of these observations, building a probabilistic model upon role chains similarly is a worthwhile strategy to simulate user role changes correctly, which will be discussed in the next section.

5.9 Model Building

After applying the tuned *Multi-Sampling and Combination Strategy* to a series of data sets in the previous section 5.8, building a model that can succinctly describe the evolution of user roles across data sets in a time series is of particular interest. To do this, user-role chains, introduced in Section 5.8.1, are a valuable strategy to expedite the building of models. A role chain of a user describes the role transitions from one data to another data set of each user. Creating whole transition models arises from

the strategy to group distinct role transitions between pairwise user roles to create a transition table for all user roles between data sets.

To do this, *Clustering* and *Classification* of data sets need to be performed, which are time-consuming due to their resource-intensive nature. *Clustering* has quadratic complexity, while *Classification* requires well-elaborated training data sets. Therefore, the model-building process should prioritize creating a non-stationary role-change model that can capture and predict role evolution without requiring extensive knowledge of the underlying data sets. The foundation for the model-building process is well-elaborated and stable user roles arising from the fine-grained user role analysis and role transitions forcing evolution patterns.

The *Olympic Games* and the *Super Bowl* data sets were analyzed using the optimized *Multi-Sampling and Combination Strategy*, leading to an expedited model creation process. Furthermore, role chains were established and analyzed in depth. As for each user role, probabilities for user role changes can be investigated using the proposed approach; the drawback is that this requires much time and resources. Building a model using information from a plethora of events for simulating and predicting data sets is the primary goal of this chapter. First, some basic information about model building using *Markov Chains* and the adaption of the analyzed data sets creating a general model design are introduced. Afterward, this model is used to discuss event pair proximities and role shifts. Furthermore, based on those results, both a naive model and an algorithm were designed to create suitable *Markov Models*. Simulations can be generated using transition models, replicating existing data sets, and enabling predictions about future events based on *Markov Models* developed from past events.

5.9.1 Background on Markov Models

After describing the suitability of user role chains and their possibility to create a transition model between topically related data sets within a time series, the applicability to exploit *Markov Models* to create an automatic model-based approach will be presented in this section. As the purposes of user roles and their transitions to other user roles are described with probabilities, a model exploiting a general *Markov Chain* is possible, as the number of states is always firm. A particular challenge is the time factor, as the data sets considered do not precisely meet the condition of a discrete-time model, with intervals each having the same amount. In addition, the transitions of user roles between different data sets make it difficult to create a completely static model.

For the model-building process, a widespread approach to depict states, as well as transitions to other states, is a *Markov Model*. *Markov Models* are stochastic models based on traditional finite state machines to depict pseudo-random changes, whereas

future states depend only on the current state. Several models were investigated in the past, depending on several properties. Observability of states and manual adjustments based on observations play significant roles in the process.

Starting with traditional *Markov Chains*, where states are fully observable as the whole system is autonomous, a Markov Chain is a random process generating a sequence of states $S = \{S_1, ..., S_n\}$ over some time as discrete-time model consisting of several snapshots or time-slices, whereas each interval has the same amount. A transition model is defined between the states, describing transition probabilities from one state to another from a previous point in time to the current state. In contrast, transitions depend only on the previous state $P(X_t = S_j \mid X_{t-1} = S_i)$. [RN21].

Different approaches, such as the Hidden Markov Model, do not meet the given constraints' expectations, as the models' states need to be visible and observable at each time. Moreover, it is almost impossible to model observations in each state, as there is only the information of a user role as a label for each user. Albeit, in the very beginning for each user, a probabilistic assignment for a role resulting in an absolute or relative majority for the best role could be a pertinent observation, but as this information is lost after simulating the first transition to another role from one to another data set, a *Hidden Markov Model* is no suitable solution for the given problem. Also (partly observable), *Markov Decision Processes* are unsuitable for the given constraints, as each state has a set of actions and a reward function needed, resulting in a decision-maker-led process. In contrast, the simulation of role changes for specific users from one data to another should take part fully autonomous [RN21].

5.9.2 Transition Tables

Starting from the definition of *Markov Chains* in Section 5.9.1, a state machine is defined by a set of states and edges, which model transitions between those states. Transferring the insights regarding varying transitions probabilities for user roles of pairwise data sets from Section 5.8 to such a state machine, there would initially be a state for each role of each data set, such as $\{AVG_{Oly12}$, AVG_{Oly14}, ..., $Amplifyer_{Oly12}$, ..., $NotAvailable_{Oly22}\}$ as well as the pairwise transition probabilities as edges together with the probabilities such as $\{(AVG_{Oly12}$, $AVG_{Oly14})$, ... $(Loner_{Oly20}$, $NotAvailable_{Oly22})\}$ for consecutive data sets such as *Olympics 2012* and *Olympics 2014*, *Olympics 2014* and *Olympics 2016*. By using the insights on role switches and chains from Section 5.8, it is possible to create role transitions for pairs of user roles across consecutive data sets.

Table 5.16 displays the pairwise percental drifts of combined user roles from all possible event pairs, focusing on whole user role shifts. For instance, it compares the drift of, e.g., all *Semi Stars* between the *Olympics 2012* and *2014* transition to the drift of

Semi Stars from *2014* and *2016* using the *weighted Manhattan Metric* from Definition 7 in Section 2.3. This comparison allows to consider the number of affected users within a transition by probabilistically comparing differences in several user roles without losing the aspect of sizes of user roles.

Table 5.16: User role transition distances between Olympic data sets.

Pair	Semi Star	Star	I. Starter	Amplif.	Rising Star	D. Chatter
Oly12/14 → 14/16	4.86%	7.27%	2.70%	8.21%	1.92%	7.40%
Oly12/14 → 16/20	8.04%	5.15%	0.91%	5.08%	1.74%	5.54%
Oly12/14 → 20/22	9.23%	5.60%	1.23%	4.75%	0.24%	7.79%
Oly14/16 → 16/20	3.39%	3.12%	2.01%	2.47%	0.93%	1.96%
Oly14/16 → 20/22	3.51%	10.65%	3.44%	3.68%	2.22%	2.42%
Oly16/20 → 20/22	2.19%	6.26%	1.71%	1.06%	1.48%	1.3%
Pair	**Comment.**	**Spammer**	**AVG**	**Forward.**	**Listener**	**Loner**
Oly12/14 → 14/16	5.59%	5.19%	5.40%	14.28%	4.16%	3.68%
Oly12/14 → 16/20	1.77%	4.13%	3.57%	12.16%	3.12%	1.08%
Oly12/14 → 20/22	0.69%	11.75%	1.70%	7.50%	2.80%	7.43%
Oly14/16 → 16/20	5.55%	2.84%	1.63%	3.59%	1.91%	1.71%
Oly14/16 → 20/22	5.77%	5.05%	4.85%	5.04%	4.13%	2.96%
Oly16/20 → 20/22	2.24%	7.62%	2.84%	6.81%	2.62%	4.46%

In the manual building process and the analysis, some drifts for specific user roles could be observed over time. Especially until the *2016 Olympics*, specific user roles had a more considerable drift, while most user roles stayed largely stable after 2016 until data sets to the present day. Considering the percental drift again, there are user roles that are hardly or only moderately affected, such as *Idea Starter*, *Rising Star*, *AVG User*, *Listener*, or *Loner*, but also user roles that shift over the whole time series, such as *Stars*, *Semi Stars*, *Spammers* or *Forwarder*. In contrast, some user roles are only shifted over some specific periods, e.g., Daily Chatter between the pairs 2012/2014 and 2014/2016 or *Commentator* between the pairs *2012/2014* and *2014/2016*, *2014/2016* and *2016/2020* as well as *2014/2016* and *2020/2022*, while receiving only more minor shifts in the other event pairs.

Focusing on generalized user role shifts from Table 5.17 especially from the very beginning to several intermediate steps, and the end can be noticed for all generalized roles. In contrast, in later event pairs, the shifts are less affected than some of the fine-grained roles from Table 5.16. In Table 5.16, the rightmost column shows how all users in an event pair compare to others when using the *weighted Manhattan metric* (Definition 7), taking into account the number of users affected by a shift. One can see more considerable shifts between the data sets, especially for the first pair *(Olympics 2012/14)* to all others, while the distance to the last data set (*Olympics 2020/22*) is becoming smaller again. The distances are relatively small for all other data sets,

which can also be noticed when calculating the average of all data set pairs and their distances to them. When leaving out the *Olympics 2012/14* data set pair in the average calculation, the distances to this new average are relatively small and even (cp. Table 5.18). Finally, this confirms the observations made in the manual building process when comparing the application to other data sets stemming from the same social network in Section 5.6. This observation affirms the need for several states for each role and each considered data set, as there are noticeable shifts of user roles between data sets. However, later data sets reveal only more minor shifts, which makes several approaches with fewer states possible to consider in the model-building process in the next section.

Table 5.17: User role transition distances between Olympic data sets.

Pair	Action Triggering	Intermediate	Passive	All
Oly12/14 → Oly14/16	7.22%	9.02%	7.49%	9.57%
Oly12/14 → Oly16/20	5.88%	7.23%	6.57%	8.26%
Oly12/14 → Oly20/22	5.06%	9.28%	6.03%	5.54%
Oly14/16 → Oly16/20	2.02%	3.32%	3.02%	2.90%
Oly14/16 → Oly20/22	3.77%	3.29%	1.37%	4.77%
Oly16/20 → Oly20/22	1.83%	1.89%	2.33%	5.13%

Table 5.18: User role transition distances between Olympics data sets and averages.

	Oly12/14	Oly14/16	Oly16/20	Oly20/22
AVG	5.51%	3.96%	3.30%	2.30%
AVG w/o Oly12	7.48%	2.14%	2.67%	2.81%

5.9.3 Model Building Process

By analyzing the transition tables, valuable insights can be obtained to enhance the development of modeling strategies that consider noticeable role changes in data sets. This *knowledge* can be utilized effectively to create *models* that deliver excellent results. Both a *naive manual-based* approach creating several *strawmen* by exploiting the knowledge of *transition tables* and an *algorithmic approach* aiding in designing suitable Markov Models were considered. *Markov Models* are beneficial for simulating user role changes over time, especially for simulating and predicting users of entirely new data sets. Thus, this section will focus on model building and experiments evaluating the simulations and predictions outcome against the strawmen and the actual data created with the aid of the *Multi-Sampling and Combination Strategy*.

In the following chapters, experiments were carried out utilizing the following *strawmen* models that simulate various specifications of the innovative non-stationary high-level threshold model:

- The *Average Over All (AOA)* model combines all user roles probabilities from each event, performing a weighted average depending on the user roles percentage in a data set, resulting in a more coarse-grained model consisting of 13 states.
- The *As Granular as Possible (AGAP)* model forces no aggregation of events but focuses on concatenating all transition tables for all user roles, resulting in a more fine-grained model with 52 states.
- The *Threshold Cutting (TC)* model is similar to the Average Over All model but separates the first event (*Olympics 2012*) from the others and concatenates them with the average of all other events as striking role shifts are outgoing from the first to the following events as figured out in Section 5.9.2 providing a more accurate prediction using 26 states.
- The *Outlying Role (OR)* model is similar to the Threshold Cutting model but separates the *Star* User from the average calculation of each event as it has a massive impact on the transition tables, resulting in 28 states.
- The *Threshold Model (TM)* is an algorithmic approach using several thresholds to approximate the *As Granular as Possible* model using a low threshold, an intermediate case, and a high threshold approximating the coarse-grained *Average Over All Model*.

Before focusing on experiments considering all of the transition models from above, the *Threshold Model* will be introduced in more detail as follows. All of the steps creating the Threshold Model can also be tracked in Fig. 5.28.

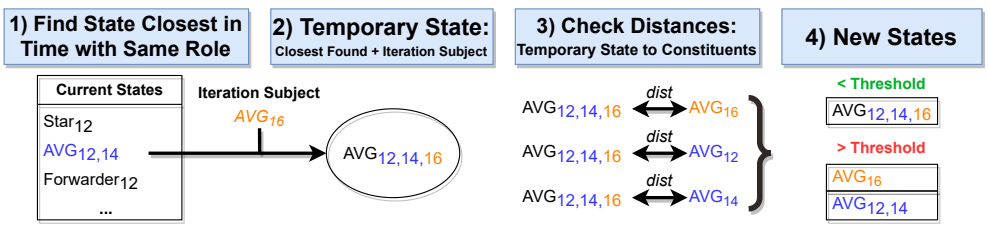


Figure 5.28: The Threshold Algorithm's combine step from [Mac23].

Definition 34 (Threshold Algorithm)

1.) Start in the first event and create for each role a separate state, e.g., AVG_{12}.

2.) Iterate in a breadth-first search over all other events states for each user role and determine a temporary combination of the closest subsequent user role to a (combined) state, e.g., AVG_{16} and $AVG_{12,14}$ to $AVG_{12,14,16}$ or AVG_{20} and $AVG_{14,16}$ to $AVG_{14,16,20}$.

3.) Create a new combined temporary state, e.g.,$AVG_{12,14,16}$ when focusing on AVG_{16} to add, and check if all of the following constraints are fulfilled considering a predetermined threshold $thresh$.

   a) The distance between the currently considered state and the temporary state undercuts the predetermined threshold using the Euclidean distance:

$$| role_{cur} - state_{temp} | < thresh \tag{5.1}$$

   e.g, $| AVG_{16} - AVG_{12,14,16} | < thresh$

   b) For each component $role_{event}$ of the closest state $state_{closest}$ the difference between the currently considered temporary state and the $role_{event}$ has to undercut the threshold:

$$\begin{gathered} for\ all\ role_{event}\ in\ state_{closest}: \\ | role_{event} - state_{temp} | < thresh \end{gathered} \tag{5.2}$$

   e.g, $| AVG_{12} - AVG_{12,14,16} | < thresh$ and $| AVG_{14} - AVG_{12,14,16} | < thresh$

   c) Focusing again on the context of handling transitions with targets, sources, and probabilities. For each transition, where source roles represent the current subject, the probabilities to all targets have to undercut the threshold due to an altering targets effect also caused by many smaller shifts.

$$\frac{| role_{tar,role_{src,event}} - role_{tar,state_{temp}} |}{role_{tar,state_{temp}}} \times P_{role_{src}(role_{tar},event)} < thresh \quad \begin{gathered} for\ all\ role_{src}, role_{tar}, event: \end{gathered} \tag{5.3}$$

   where the fraction indicates the percentage of the source role movement to the target, and P represents the target role probability stemming from the source role in the event,
   e.g, $\frac{|Star_{AVG,12} - Star_{AVG,12,14,16}|}{Star_{AVG,12,14,16}} \times P_{AVG}(Star, 12) < thresh$
   for assuring that transitions from AVG as a source to Star users are shifting the Star users below the threshold.

4.) Replace the closest subsequent state with the temporary state if all constraints are fulfilled, e.g., $AVG_{12,14} \rightarrow AVG_{12,14,16}$ else revert the temporary state to the previous closest subsequent state $AVG_{12,14}$ and create a new one for the currently focused state AVG_{16}

5.9.4 Preparation & Execution of Experiments

After introducing all models, they need to be verified before they can be used for simulating and predicting (future) events. Regardless, the *Threshold Model* must be approximated to the *strawmen* models before the analysis starts. While the *As Granular as Possible* model, with the most achievable states, and the *Average Over All* model, with the least viable states, are both extreme cases and frame the other models, both the *Outlying Role* as well as the *Threshold Cutting* models aim to approximate the *As Granular as Possible* model by separating both single events as well as roles.

The general simulation process to verify the models using previous known information from the clustered and classified data sets is as follows. The process starts with, e.g., the user role labels resulting from the highest probability of each user of the *Olympics 2012* data set, simulating the *Olympics 2014* by drawing a random variable, determining which transition path is chosen in the models, and approximating all of the other events in chronological order using the interim results until reaching a final simulation for, e.g., the *Olympics 2022* data set. At each interim step, the results can be compared to the baseline, represented by the actual data sets, as all of them have been analyzed with the general *Multi-Sampling and Combination Strategy* from Chapter 4, to ensure the effectiveness of the simulation by evaluating the *Average Prediction Error*, i.e., the amount of false simulated users. In contrast to a simulation, predicting new data sets is also possible. When assuming a new data set, e.g., *Olympics 2022* should be predicted, the model-building process takes the *Olympics 2012-2020* data sets into account, while the simulation starts with the *Olympics 2012* until *2021*, as mentioned before, adding another final simulation step.

Firstly, the manually designed models (*strawmen*) are evaluated against the baseline considering the percentage of wrongly assigned users, *Quota* and *Standard Deviation* of specific roles, and most substantial role drifts. In contrast, the second step focuses on validating the algorithmic approach approximating the manually created models. As the models' number of possible states represents the granularity of the model, the influence on the assignment process, w.r.t. to wrong assigned users, is a good quality characteristic evaluating the models' simulations against the actual data sets.

In Table 5.20 for both the simulated time series of the *Olympics* between *2012* and *2022* as well as the *Super Bowl* between *2013* and *2022* can be seen for all manual models as well as *Threshold Models* with several thresholds approximating several

models between the *Average Over All* and the as *Granular as Possible* model, using thresholds of 0.4, 0.6 and 0.8, as Table 5.19 reveals. One can see that the low *TM* with a threshold of 0.4 is close to the *As Granular As Possible* model, the medium *TM (0.6)* lies between the *AGAP* and the *TC* as well as the *OR* model, while the high *TM (0.8)* is closer to the *TC* and *OR* model.

Table 5.19: Number of states for all models and both event time-series simulations.

	AOA	**AGAP**	**TC**	**OR**	**TM(0.4)**	**TM(0.6)**	**TM(0.8)**
Olympics	13	52	26	28	51	49	41
Super Bowl	13	39	26	27	39	36	29

When discussing probability-based models, all mentioned models can deliver a different outcome in the simulations. A promising approach is to multiply simulate each step and compare the Standard Deviation against each other for counteracting role drifts in several directions. This approach comes with linear growth of runtime, including finding the best result for each event with the aid of the Standard Deviation. As the runtime for a single simulation for one event needs only about 5-7s and ten simulations need about 60 - 70s, the effort is quite profitable, as the best simulations are found.

5.9.5 Comparison of Model Approaches

After introducing the general Process of creating Models in Section 5.9.3 as well as focusing on the preparations of experiments in Section 5.9.4, in this Section, a plethora of experiments will be performed and evaluated to discuss following aspects.

- Are the threshold models built upon related data sets able to simulate user role transitions more precisely than static model (strawmen) simulations?
- Is it possible to successfully transfer the whole model-building process from one data set series to another?
- Is it possible to successfully simulate user role transitions exploiting a model built upon another topically related data set?
- Is it possible to successfully predict user roles for an entirely new data data set with a model built upon older related data sets?

Table 5.20 gives a compact overview of all experiments conducted in this section for simulating both the *Olympics* and *Super Bowl* Data sets with the aid of models built from related data sets (first two segments), the simulation of the *Super Bowl* data

set with models built from the *Olympics* data sets, as there is a topical as well as temporal correlation (third segment) and the prediction of the latest *Olympics* data set exploiting the model built with the aid of the older evaluated data sets analyzing the *Average Prediction Error* of all models.

Comparison between Threshold Model and Strawmen The *Average Over All* model delivers the highest percentage of wrongly assigned users for all event steps in both event series (first two segments). While the *Threshold Cutting* and *Outlying Role* models furnish good results within the first simulation step, the following steps decline as they deliver more wrongly assigned users. Only the *As Granular As Possible* model has a consistently low share of wrongly assigned users. When observing the results of the *Threshold* models, all models deliver better results than the *Average Over All*, the *Outlying Role*, and the *Threshold Cutting* model. At the same time, only the *Threshold Model* with a threshold of 0.4 is an almost perfect approximation to the *As Granular As Possible* model for all steps of both event simulations. These insights make both latter methods the most suitable for further simulations.

Table 5.20: Average Prediction Error of the models for simulation and prediction.

	AOA	**AGAP**	**TC**	**OR**	**TM(0.4)**	**TM(0.6)**	**TM(0.8)**
Simulation with Olympics Model							
Oly14	10.98%	0.07%	0.05%	0.09%	0.05%	0.10%	0.38%
Oly16	5.67%	0.02%	3.65%	3.70%	0.04%	0.06%	0.33%
Oly20	7.47%	0.03%	4.12%	4.13%	0.01%	0.03%	1.75%
Oly22	6.81%	0.05%	4.32%	4.26%	0.03%	0.10%	0.26%
Simulation with Super Bowl Model							
SB20	17.43%	0.05%	0.07%	0.06%	0.05%	0.22%	0.47%
SB21	8.05%	0.07%	3.70%	3.59%	0.11%	0.72%	0.90%
SB22	6.23%	0.07%	3.39%	3.37%	0.10%	0.60%	0.90%
Simulation with Olympics Model							
SB21	13.07%	15.85%	16.09%	16.14%	15.86%	15.30%	15.77%
SB22	11.15%	10.71%	12.65%	12.75%	10.68%	10.16%	10.29%
Prediction with Olympics Model							
Oly22	9.04%	6.18%	6.49%	6.50%	6.23%	6.23%	5.93%

When considering user role drifts for the first two segments of Table 5.20, Table 5.21 shows for each model and each data set, the most affected role in terms of the average prediction error together with the deviation from the original values created with the *Multi-Sampling and Combination Strategy*. Especially the *Spammer*, *Loner*, and

Commentator have relatively high deviations when simulating with manually created models except for the *As Granular As Possible* model. Both the *Outlying Role* model as well as the *Threshold Cutting* model show that improvements, w.r.t. granularity for both role states as well as event states can be reached, as drifts for *Star*s can be cut short compared to the drifts between the other events as well as the whole drift between the first and the second simulation deviates for both event-series simulations.

Focusing on the *Threshold* models, it stands out that the *Spammer's* role is not well simulated in the models with a threshold of 0.6 and 0.8 in the *Super Bowl* event series. When pointing to Table 5.13, the trend for the user role *Spammer* is strongly decreasing for later events such as the *Olympics 2022*. As the latest *Super Bowl* data sets were recorded in the same period as the *Olympics 2022*, the amount of *Spammers* is logically decreasing until the beginning of *2020*. However, the high deviations in the later *Super Bowl* data sets cannot be explained solely by the substantial decrease in the user role of the *Spammer* between *2013* and *2020*. A closer look at the development between *2020* and *2022* reveals that the role of the *Spammer* is increasing again very strongly, which can pose problems even for the *Threshold Models*. Only the *Threshold Model* with a threshold of 0.4 copes very well with this anomaly. When focusing on weaker roles or those with anomalies such as the *Spammer* in the *Super Bowl* data sets between *2020* and *2022*, the *Standard deviation* between the original data sets and the simulations also shows higher *Standard deviations*, as a difference of only a few users is noticeable. Nevertheless, the more simulations were done, the weaker user roles or those with anomalies could benefit from finding more stable results.

Table 5.21: Biggest user role drifts of the models' simulations for both time series.

	AOA	**AGAP**	**TC**	**OR**	**TM(0.4)**	**TM(0.6)**	**TM(0.8)**
Oly14	Spammer 42.34%	Loner 1.37%	Loner 1.49%	Star 1.31%	Loner 1.08%	Star 3.80%	Semi Star 11.15%
Oly16	Comment. 192%	Star 0.95%	Comment. 185%	Comment. 184%	Rising Star 0.50%	Semi Star 3.85%	Semi Star 5.91%
Oly20	Loner 198%	Loner 1.56%	Loner 210%	Loner 214%	Star 0.49%	Star 1.22%	Spammer 8.35%
Oly22	Spammer 436%	Loner 0.59%	Spammer 316%	Spammer 317%	Comment. 0.53%	Star 13.03%	Semi Star 9.99%
SB20	Spammer 3.12%	Star 1.24	D. Chatter 1.30	Star 1.15%	Spammer 1.45%	Semi Star 18.03%	Spammer 284%
SB21	Loner 90.26%	Loner 1.30%	Loner 172%	Loner 168%	Rising Star 0.89%	Spammer 12.88%	Spammer 15.38%
SB22	Spammer 88.96%	Amplifier 1.27%	Spammer 185%	Spammer 179%	Semi Star 0.82%	Spammer 50.88%	Spammer 94.45%

The results of the two first segments show that a dynamic threshold model can be built straightforwardly and dominate the results of several static approximations w.r.t. *Average Prediction Error*. Moreover, it is easily possible to transfer the dynamic model-building process from one data set series (*Olympics*) to another topically similar data set series (*Super Bowl*), validating the first two research questions.

Exploiting a Topically Related Model A further step into analyzing new data sets with the aid of existing topical and temporal-related models can be seen in the third segment of Table 5.20. The model exploited in the first segment, was used to simulate user role transitions, starting with the role labels of the *Super Bowl 2020* data set, simulating the events in the following two years. Comparing the average prediction error of the two events, *2021* and *2022*, to those of the second segment, one can see that the prediction is possible. Unsurprisingly, a close match to the average prediction error is impossible, as a comparison between long-term events ,e.g., the *Olympics* and short-term events, e.g, the *Super Bowl* may affect the simulation. However, between 85% and 90% of users were estimated correctly, showing the benefits of using an algorithmic-driven non-stationary model instead of clustering and classifying samples as part of the whole *Multi-Sampling and Combination Strategy*, reducing the runtime from days to a few minutes and thus saving many resources. Even though some unpopular roles were not predicted precisely for the *strawmen* and *non-stationary models*, popular user roles were determined successfully. Comparing the *prediction error* of the *strawmen* against the *non-stationary* models shows the validity and the benefits of the novel algorithmic model-building approach, as all strawmen, except for the *Average Over All* model, are outperformed by at least one non-stationary model. The inaccuracies of the predictions arise from the lack of additional data sets, which could be improved by adding more *Olympics* and *Super Bowl* data sets (cp. Table 5.1), which are currently unavailable.

Prediction of entirely new Data Sets After demonstrating and evaluating effects and observations in simulations of related data sets and topically and temporally related data sets, the logical next step is to predict a completely new data set using the proposed non-stationary model. Focusing on the prediction of the *Olympics 2022* data set, the Markov Model is created with the transitions of the *Olympics* data sets between *2012* and *2020*, sparing the *2022* data set out, as assumed to have no knowledge of this data set. Predicting the *Olympics 2022* data set, a whole simulation starting with the *2012* data until reaching the *2020* data set is performed, always reaching the latest possible state, e.g., AVG_{20} for each user in the *Markov Model*. When trying to predict the outcome for each user in the following transition, the outcome for the 2020 data set is simulated once again, which means that the simulation performs the transitions

from the last iteration ending in the same latest possible states again. This prediction can now be compared to the original sampled, clustered, classified, and combined data set and the simulation outcome with the model, including the transition model inclusive of the data set to predict. The benefit of the prediction is saving a lot of time and resources for clustering, classification, as well as sampling, and combination; as a whole, prediction can be performed within a few minutes, whereas the whole pipeline process consumes about 20 minutes only for clustering a single 5% sample from the *Olympics 2012* data set. The prediction method outperforms the pipeline approach consistently, even when executed in parallel.

Pointing to the results of the simulation (last row in the first segment) and the prediction (last segment) of all models in Table 5.20, one can see that the prediction of the *strawmen* models except for the *As Granular As Possible* model are relatively close to them of the simulated model. At the same time, the *algorithmic approach* at least outperforms the *strawmen* but is not that close to the simulated results. Having a closer look at the *Threshold Model (0.4)* in Table 5.22, the quota values, which describe the difference counts between the original data and the counts of simulation/prediction concerning the counts of the original data set are very critical. Both the *Average Error* and several user roles, especially the *Spammer*, deviate compared to the original data set, which comes along from the anomaly effects described prior in this Section. Unsurprisingly, the simulation performs better than the prediction for all user roles regarding *Quotas* and *Standard Deviation*. However, the prediction still shows good *Quota* and *Standard Deviation* for *Forwarder*, *Average User*, and *Semi Stars*.

Table 5.22: Biggest User role drifts of the Threshold Model (0.4).

	Original	**Simulated**				**Prediction**			
	Counts	**Counts**	**Diff.**	**Quota**	**StD**	**Counts**	**Diff.**	**Quota**	**StD**
Forwarder	1645582	1644363	1219	0.07%	0.07%	1672365	26783	1.63%	0.05%
AVG	670866	672113	1247	0.19%	0.12%	618701	52165	7.78%	0.10%
Listener	257290	256652	638	0.25%	0.12%	217391	39899	15.51%	0.21%
Daily Chatter	153425	153825	400	0.26%	0.28%	222396	68971	44.95%	0.26%
Amplifier	77496	77522	26	0.03%	0.28%	43130	34366	44.35%	0.27%
Commentator	52342	52330	12	0.02%	0.37%	18617	33725	64.43%	0.22%
Idea Starter	49208	49154	54	0.11%	0.27%	38315	-10893	22.14%	0.31%
Semi Star	47531	47465	66	0.14%	0.54%	51189	3658	7.70%	0.34%
Rising Star	43246	43245	1	0.00%	0.52%	59184	15938	36.85%	0.57%
Loner	22227	22305	78	0.35%	0.42%	2999	19228	86.51%	0.29%
Spammer	18191	18470	279	1.53%	0.44%	84932	66741	366.89%	1.69%
Star	16921	16881	40	0.24%	0.63%	25106	8185	48.37%	0.91%
Avg Error				0.06%				6.23%	
Avg StD				0.02%				0.02%	

The remaining roles had some more significant drifts, e.g., the user role of the *Spammer*, which was affected in most of the previous simulation analysis (cp. Table 5.21) but also was affected in the analysis of the *Olympic Games* (cp. Table 5.13), as significantly the *Quotas* show more considerable deviations. At the same time, the *Standard Deviation* is not affected too much, which shows that the prediction approach is primarily pointing in the right direction. *Markov Models* are an ideal opportunity to uncover trends quickly. However, this approach is not yet robust enough to replace a whole clustering, classification, and combination strategy, as results remain not precise enough. Finally, the algorithmic approaches, such as the *TM* with a threshold of 0.4 but also 0.6 and 0.8, representing a less granular model, perform an adequate prediction with similar *Average Errors* and partly persuasive outcomes for some user roles. In contrast, the manual models, excluding the *As Granular As Possible* model, are outperformed by the algorithmic ones.

5.10 Related Work

This chapter introduces further approaches on top of the *Multi-Sampling and Combination Strategy*, which were not yet discussed in Section 4.8. Further related work dealing with model building as part of long-term user role analysis will be discussed.

5.10.1 Fine-grained User Role Analysis

The way after the probabilistic combination of clustered and classified users included several aspects of long-term user role tracing across several data sets, leading to algorithmic steps of model building. When studying the evolution of users along roles, the approaches usually only cover short periods of continuous observations while retaining the scope of coarse-grained groups: Varol et al. [Var+14] present an analysis of user behavior spanning a month, whereas Antelmi et al. [AMS19] investigate the tracking of user role evolution for approximately four months. In turn, approaches targeting multi-role allocation, e.g., Rocha et al. [Roc+11] and Lazaridou et al. [LNN16], are still limited to coarse-grained user groups, while the user role detection in this work focuses on a plethora of fine-grained user roles.

5.10.2 Model Building & Long-Term Analysis

While in social media, the detection of internal threats is a widespread use case, as Legg et al. [Leg+15], Pannell et al. [PA10], and Kim et al. [Kim+19] focus on the detection of user behavior over time finding and detecting changes caused by stolen or

compromised user accounts, this work focuses more on the general evolution of fine-grained user roles caused by typical pervasive evolutionary drifts of user roles. As the focus in this work is the detection of user roles changes of fine-grained user roles, based on tracing and aggregating the behavior of single users, Yu et al. [YHL15] propose a Bayes-model based approach focusing more on anomalies of general groups instead of single users, inferring user roles from input data and possessing a dynamic extension over some time, as the complete user role tracing and model building approach is based on a separate evaluation of disconnected data sets. Model-building approaches forcing the analysis of evolutionary data is a prevalent practice in medicine, classifying and analyzing documents, forcing decisions, such as the work of Sonnenberg et al. [SB93], Komorowski et al. [KR16], Sato et al. [SZ10] as well as Yi [YB09]. In contrast, approaches to tracing user role evolution using models still need to be elaborated. Regardless, classifying and labeling documents is related to analyzing user behavior by their messages. A further similar approach to user role migrations is addressed by Thurner et al. [Thu+21]. They evaluate and predict election results for political parties in electoral districts over voting periods to trace migrations of voters between political parties, non-voters, and first-time voters. With the aid of these transition models and further specifications of voters and their behavior, hybrid models specified in Klima et al. [Kli+17] can be created to estimate the quotas of political parties.

When discussing Markov Models, only a few approaches adjusted on constraints, such as dealing with dynamical models covering states in several years, were published, especially in the early 2000s. The work of [ZHH02] suggests aggregating web pages having a similar action into single states, forcing them to simulate past activities and predict future clicks. Markov Chains were also exploited in the work of [MAB09] dealing with dynamic systems in the use case of calculating reputations of websites aggregating similar states in terms of a Bayesian Information Criterion. While both approaches rely on a plethora of data points, the model-building approach in this work shows remarkable results even with a small amount of data sets, resulting in a similar learning process for both events and roles using a more dynamic kind of aggregation. The work of [MAB09] relies on Machine Learning (ML) techniques dealing with abundant nodes evolution in time series forcing an approach dealing with fewer states but many dimensions. The use case of this approach would result in concatenating multiple chains due to 13 roles in 5 events in contrast to the more suitable merges presented in the transition tables section 5.9.2. While Fu [Fu19] works on tracking users in forums such as Stackoverflow using a time-aware model and only examining two distinct user roles, On-at et al. [Oa+16] applied a time-aware social-profiling model to investigate user-interest evolution over time in the social media service Twitter in an ego-centric network. A further approach by Wang et al. [WZZ15] dealing with the user role evolution in an online health community-based use case focuses on creating user roles with social support types forcing a pairwise

short-term (2 months) observation of user roles with the aid of a transition model for future predictions.

5.11 Conclusion

The methodology discussed in Chapter 4 was utilized and applied to a single dataset. Later, the same approach was applied to multiple disconnected datasets, whereas all steps of the proposed approach delivered suitable, traceable, and comprehensive results. Starting with *Feature Selection* and *Preprocessing*, followed by *Clustering* and (manual) *Classification*, the same fine-grained user roles were detected across all given data sets. As manual classification at the beginning is inevitable and a bottleneck considering time management, while analyzing samples of data sets, training data is built for several disconnected data sets, forcing a less time-consuming process in analyzing user roles. Reducing the size of *samples* and simultaneously increasing the number of samples has a considerable benefit on the runtime and needed resources of the quadratic complexity of clustering. As in most cases, massive data sets are hardly possible to cluster the novel *Multi-Sampling and Combination Strategy* improves the *certainty* and *stability* of *user roles* by considering users multiple. Each user observed in more than one sample is *combined probabilistically*, leading to an averaged user role vector consisting of all observations.

As each step is applied to several *data sets* stemming all from the *same source* with fewer adjustments and minor manual checks, the traceability and certainty of user role detection are given in each step. Moreover, several optimization steps considering the sampling and classification lead to improvements w.r.t. the coverage of users and the stability and certainty of user roles. Transferring the *knowledge* of manually built classifiers to completely new topic-related or close time-related data sets stemming from the same social network also led to promising results. Also, combining several *training data sets* resulted in an auspicious outcome, reducing the necessity of creating training data sets for each new data set.

After processing the novel *Multi-Sampling and Combination Strategy* on each data set, tracing of user roles, their movements, and evolution are strained, observing both transitions of a user from role to role as well as users leaving and returning in observed time series, constituting a peculiar challenge in terms of model building, as knowledge for at least two transitions are missing. The observations considering the *transitions tables* showed that a stable tracing of user roles over time is possible, as only a few role chains represent the most occurring transitions for each role. This degree of user role stability made a *model-building process* possible: An *algorithmic threshold-based model* approximating several degrees of granularity was evaluated against several *static models*, so-called *strawmen* from coarse to fine-grained. With these models' aid,

user role evolution simulations can easily be applied over various data sets and time approximating the actual outcome from the clustered and classified data sets serving as a *ground truth*. The experiments and the observation showed that the algorithmic model-building process could deliver the most widely satisfying results, except that conspicuous anomalies across data sets are hardly graspable.

Moreover, the experiments showed that models could simulate topical and temporally related and predict completely new data sets without clustering and classifying whole data sets. Especially clustering is an acquainted bottleneck due to the quadratic complexity w.r.t runtime and needed memory. In particular, a model-based prediction is an extraordinary opportunity to rapidly assess user role evolution, as the ground truth results were mainly achieved in the experiments. All in all, both the *simulation* of *known data sets* as well as the *prediction* of *new data sets* using the *algorithmic threshold models* showed that the results of the ground truth are accomplishable, but several improvements considering anomalies due to feature drifts, need to be further incorporated, receiving more resilient models.

Chapter 6

Analyzing Fine-Grained User Roles in Telegram

Oh, but I'll take my time anywhere
Free to speak my mind anywhere
And I'll redefine anywhere
Anywhere I roam
Where I lay my head is home

Metallica - *Wherever I May Roam*

After applying the general Knowledge Discovery (KD) approach to the Twitter use case, this section addresses the initial conceptual transfer of the pipeline to an entirely new data set with new features. The Telegram use case presented in this chapter covers all steps of the novel *Multi-Sampling and Combination Strategy* to prove the general application of the approach. In addition, an initial analysis of the stability and certainty of user roles completes this chapter.

6.1 Motivation and Contributions

Compared to the social media service Twitter, presented in Section 5.1, Telegram is more of an *instant messaging service* and thus very popular as fast communication between *two individuals* is possible. Nevertheless, Telegram also enables *group conversations*, distinguishing from Twitter, having an entirely different structure as a global stream, and allowing retweets and forwards within a single thread but no isolated groups. Moreover, hashtags are not an essential issue in Telegram, compared to Twitter, where hashtags are substantial to organize and structure content. In contrast, structures in Telegram are modeled as *groups* or *channels*, where users provide interested users with information or address intense conversations between people. These aspects make Telegram a hybrid between *classic messengers* and *social media services* like Twitter. Furthermore, groups, especially with massive numbers of users, are valuable communication and information diffusion benefits. Especially the considerable rise of Telegram in the last few years, establishing *communities* for conspiracists, alternative media consuming, radicalizing, and safety-conscious people, due to the lack of supervision of administrations or corporations such as Google or Meta yields data for attractive analyses, as the data is broadly available due to the Telegram API [23]. Thus, the community of Telegram is very active in creating and spreading content and information within their community and concentrates primarily on this messenger. The general scope in this chapter concentrates mainly on the universal applicability of the KD pipeline, particularly the first stages, and the analysis of fine-grained user roles. Most of the steps of the approach presented in Chapter 4 were applied to the Telegram data set, while the aspects of *Sampling* and *Combination* are currently at an early stage and need further investigation, as the entire application was not yet accomplished. More data sets are essential for *long-term evaluation* and *model building*. Thus, those aspects and a more detailed analysis of the *Multi-Sampling and Combination Strategy* remain future work.

Finally, the following research questions for the Telegram data set arise from those of Twitter as follows:

- Is it possible to find similar fine-grained user roles as in the Twitter data sets?
- Can the KD pipeline from Chapter 4 applied to the Telegram data sets straightforwardly?
- Can the insights gained in Chapter 5 cut short the application of the KD pipeline approach?
- How does the user behavior in the Telegram data set define from Twitter users?

[23] https://core.telegram.org

6.2 Background on Telegram

Before the pipeline considering the KD approach presented in Chapter 4 and firstly executed on the Twitter data sets in Chapter 5 will be performed on a data set from the instant messaging platform Telegram, more specific details of Telegram, based on the general definition of social media and networks from Section 2.1.1, will be presented to show the suitability of the approach with several contributions.

In Fig. 6.1, a general *taxonomy* of Telegram can be seen, including all types of possible *entities* and *relationships* extracted from the API. In addition, Fig. 6.2 shows the possible types of *communication* in Telegram. A chat can be a *direct chat* representing direct messages (Fig. 6.2, left) between two users. In addition to direct messages between two users, *channels* (Fig. 6.2, middle) and *groups* (Fig. 6.2, right) exist. A channel is also a type of chat where only one user - the owner - sends messages to an unknown set of subscribers, always flagged as a *Broadcast*. Neither the owner nor the subscribers know who follows this channel, but each channel can be linked to a *comment group* where comments on posts stemming from the *channel* can be discussed. A group in Telegram consists of several users who can communicate freely with each other. There are also several groups' specifications, such as *Megagroups* and *Gigagroups*, which are Channels with specific flags representing groups with different amounts of users who can join.

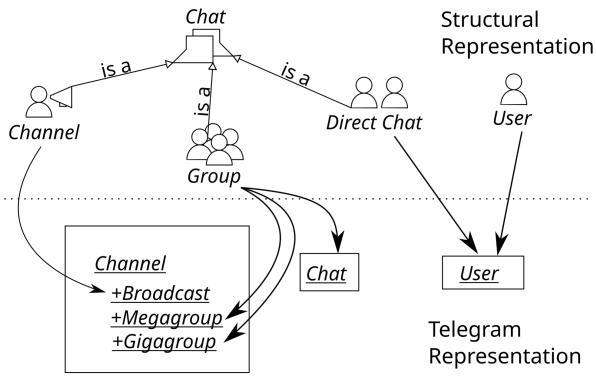


Figure 6.1: Relationships objects - Telegram from [End21].

For this work's data set, *groups* and *channels* are the most suitable conversation methods, as many distinct users and their behavior can be captured. While channels only represent one user being active, the group behavior approximates user behavior from Twitter, as direct intercommunication between users is likely possible. As channels

can have group chats and their administrators may also participate in other groups, the decision to grab them was straightforward because of possibly changing features. In contrast to groups, direct messages between two individuals are unavailable to grab due to privacy reasons. There would be no benefit, even if possible, as typical user behavior can be indicated better in groups.

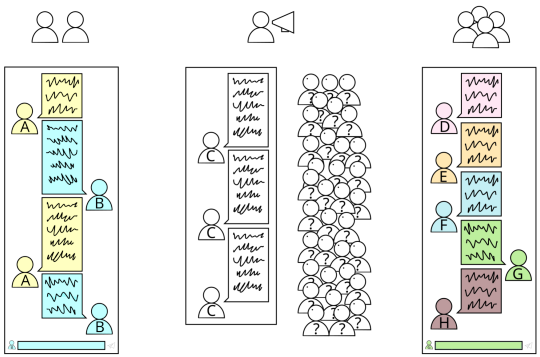


Figure 6.2: Types of communication - Telegram from [End21].

6.3 Data Sets and Preparation

As mentioned in Section 6.1, Telegram is a somewhat controversial instant messaging service. Especially during the coronavirus pandemic, Telegram enjoyed a more specific clientele, as hoaxes considering the pandemic, vaccines, and other famous conspiracy theories were massively spread in social media. In contrast to most social media services such as Twitter, Facebook, or Instagram, where most of the content was moderated and deleted, and users were banned, Telegram tolerated that clientele, leading to a boom in this service. All these aspects made telegram a beneficial service to analyze, as various groups of several users can be scrutinized.

For an initial study in this thesis, several well-known German conspiracy theorists and their activity in groups and channels were examined during the Coronavirus pandemic due to aspects of *activity* and *interconnection* of all participants. The data set has 13254 users who wrote 580201 messages between 03.07.2021 and 19.07.2021. Only 450813 messages were sent by actual users, while moderated groups sent the remaining messages. Compared to the Twitter data sets in Section 5.3, this data set was recorded only over a *limited period* and thus had fewer messages and users. Furthermore, as Telegram provides no hashtags to find only related messages to a specific topic,

several conspiracist-related groups and channels were picked out manually. As the single groups are relatively small and thus not representative at all, those groups were considered as one extensive data set. Like in the Twitter data sets, the aggregation process of building feature vectors considered only users who were active at least twice, i.e., wrote two messages at minimum (also in different groups and channels), aiding a data set with mainly active and connected users. This data set serves as a first approach to apply the whole pipeline from Twitter to Telegram, while further investigations with additional Telegram data sets are planned in the future.

6.4 Adapting the Methodology

As the data set was introduced in the previous section, the essential steps of the proposed approach from Section 4 will likely be applied as in Chapter 5 for the Twitter data sets. Starting with *Feature Engineering*, the process of the KD approach proceeds with *Clustering*, *Cluster Analysis*, *Manual Class Labeling*, and *Building the Classifier*.

6.4.1 Feature Engineering

Like in Section 5.4.1, the first step of the proposed approach is *Feature Engineering* from Section 2.4, comprising several significant steps of Preprocessing such as *Feature Selection*, *Aggregation* of user messages and *Preprocessing*, resulting in *feature vectors* for each distinct user.

Similar to the execution of the Twitter data sets, all messages are also provided as JSON data composed of various features. While some features can easily be transferred entirely from the *raw data set*, others must be processed, as mentioned in Section 5.4.1. As the features from Twitter delivered a satisfying outcome in terms of clustering and classification, the choice for user features followed mainly those from Twitter, but also new features were considered as Telegram provides several different features. Thus, most of the features from Telegram were also established in the literature mentioned in Section 5.4.1, while Telegram-specific literature considering user roles and features is currently unavailable.

Fig. 6.3 shows the chosen features as a Venn diagram, while Table 6.1 briefly describes each feature. In contrast to the features established in Twitter, where *following* relationships between users were established, the Telegram features do not provide any *following* relationships. The number of channels a user *participated* in was tested as a *network-position-based* feature, but it was dropped as it had almost no correlation with other features and negligible impact on the cluster hierarchy. Thus, there is no overlap in *User Activity*, *Network Position*, and *Network reaction* feature areas.

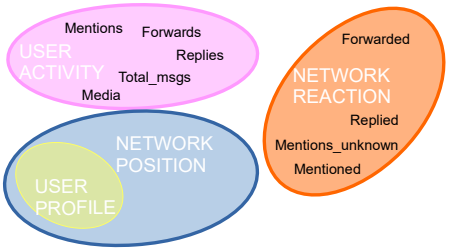


Figure 6.3: User feature classification

Focusing on *User Activity*-based features, there is the most considerable overlap of similar features to the features chosen in the Twitter data sets. `total_messages` describes the number of messages a user sends in groups or channels. At the same time, `replies` is the number of replies to other users' messages, and `forwards` is the number of forwarded messages proportional to all messages a user sent. `Mentioned` delineates the number of additional users mentioned in all messages a user wrote, and `media` describes the percentage of media used in messages. As many memes, pictures, and videos are shared within an instant messaging service, this feature may describe users more precisely. These features describe a user's behavior in creating new content or replying to and forwarding existing content, making it possible to distinguish between different types of users considering their messaging activity. Pointing to *Network Reaction-based* features, `forwarded` characterizes the forwarded number of messages of a user being forwarded by other users, `replied` the number of messages being a reply to a user's message from other users, and `mentioned` the number of user mentions affected by other users. These features describe the ability to *trigger reactions* in the instant messaging service Telegram, as some users are somewhat *active* in reacting to other users' messages, as content can be forwarded, replied to, or users even mentioned by others, describing the standing and popularity of users.

Network-Position and *User Profile* features were present in Twitter but could not be investigated in the Telegram data set, as features such as profile reputation, the number of channels a user is a member of, or activity of users within several time slots all over the day, were too correlated and dropped. In an early iteration of feature engineering, many other features were considered but dropped as they were too correlated and influenced the clustering negatively or had little discriminative power. Besides features defining the number of messages written at a specific time over the day (morning, afternoon, evening, night, business hours), also features describing the amount of `mentioning` users with `unknown` usernames, the number of different channels and groups a user is participating and a `profile reputation`, where several features such as name, the number of uploaded pictures as well as the fact if a user is a bot were

considered, were dropped. As feature engineering for this use case also focused on a straightforward calculation like in the Twitter use case, complex network-based, spatio-temporal, or content-analysis-based features were not considered.

Table 6.1: Overview on Telegram features.

Feature	**Description**
total_msgs	The number of newly created messages of a user during the record of the data set. Describes the activeness of a user.
forwards	The number of messages, forwarding a message of another user. Describes the diffusion of information.
replies	The number of topic-based replies of a user. Describes the communicativeness of users.
replied	Percentage of how many users' messages got at least one reply.
forwarded	Percentage of how many users' messages got at least one forward.
mentioned	The number of times a user has been mentioned in other users' messages.
mentions	The number of times a user mentioned other users in a message.
media	Percentage of messages consisting of media such as images or videos.
mentions_unknown	Percentage of messages consisting of mentioning unknown users.

After choosing suitable features according to the features used in the Twitter data sets, the Telegram features were *validated* to prove their suitability and usefulness, utilizing the *correlation* of features to each other and ensuring the *variance* of domains and *skewness*. The *Feature Engineering* process commonly involves *iteration*, as the success of the clusters can only be assessed during the clustering stage. Occasionally, comparable clusters emerge in different subtrees only differentiated by a feature or two. Consequently, solely features with minimal positive or negative *correlations* were examined, as the interplay between numerous features usually produced more distinct user profiles. Starting with the *correlation* matrices in Fig.s 6.4 and 6.5 displaying the iterative feature engineering process, one can see that the time-based user features are rather non-correlated to the other features, but have a relatively strong anti-correlation to the other time-based features such as `time_morning_frac`. Also, the feature `profile_reputation` and `different_channels` had almost no correlation to any other feature and yielded an unsuitable combination of clusters.

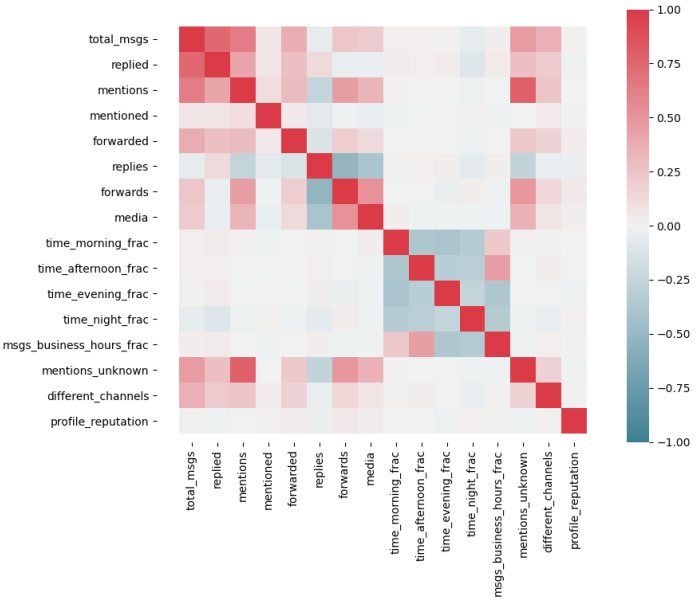


Figure 6.4: Correlation matrix - Step 1.

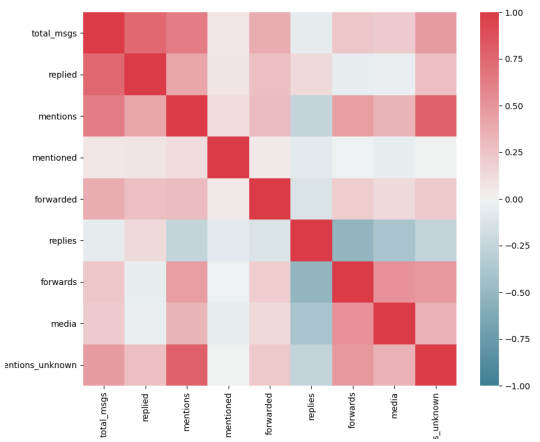


Figure 6.5: Correlation matrix - Step 2.

Stronger positive correlations can be noticed between the features `total_msgs`, `replied` and `mentioned`, `mentions` and `mentions_unknown`, whereas `replies` have a medium anti-correlation to `forwards`, `media`, and `unknown_users`. Finally, the features in Fig. 6.5 were chosen as they delivered the most satisfying outcome after the clustering, leading to suitable clusters in the hierarchy, which will be discussed later in Section 6.4.2.

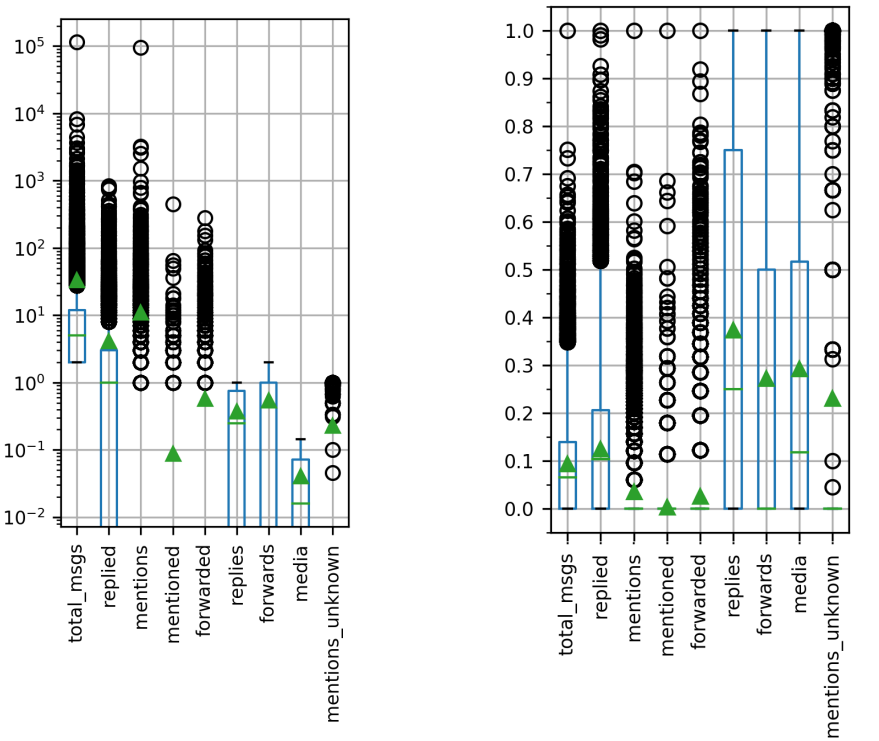
Table 6.2: Original feature statistics for Telegram data set.

Features	**Mean**	**Median**	**99%**	**Max**	**Skew**	**StD**
total_msgs	33.34	5.00	351.94	114679	111.67	1006.29
replied	4.14	1.00	55.0	840	21.01	20.36
mentions	10.97	0.00	64.47	95546	114.47	831.46
mentioned	0.09	0.00	1.0	450	104.35	4.05
forwarded	0.57	0.00	11.0	282	28.27	4.78
replies	0.37	0.25	1.00	1.0	0.49	0.39
forwards	0.54	0.00	2.00	2.0	0.99	0.76
media	0.04	0.02	0.14	0.14	0.83	0.05
mentions_unknown	0.23	0.00	1.00	1.00	1.28	0.42

Having found appropriate features, *feature normalization* (cp. Section 2.4) as well as *feature standardization* were applied to the data set, alterations of features and their influence on *skewness* and *domain variation* between the raw data set and the normalized and standardized data set can be visualized. As the *logarithmic transformation* worked well in the Twitter data set, reducing strongly right-skewed data, the deviations of skewness between the raw features in Table 6.2 and the standardized and normalized features in Table 6.3 show also the benefits for the Telegram data set. Especially for `total_msgs`, `mentions`, and `mentioned`, which all have an extremely high maximum value, the 99th percentile has an increased distance to the maximum, showing the need for standardization as it leads to reducing significant outliers. The *skewness* was decreased for all features to a minimum, except for mentioned showing an advance, too. While mean and median values only show hardly noticeable deviations except for *total_ msgs*, *mentions*, as well as *replies*, the *Standard deviation* was reduced a lot for *total_ messages* and *mentions*. In contrast, the other features already showed a low *standard deviation* in the tables.

Table 6.3: Normalized feature statistics for Telegram data set.

Features	Mean	Median	99%	Skew	StD
total_msgs	0.09	0.07	0.45	1.64	0.11
replied	0.12	0.10	0.60	1.46	0.14
mentions	0.03	0.00	0.36	2.94	0.08
mentioned	0.00	0.00	0.11	14.54	0.03
forwarded	0.03	0.00	0.44	4.46	0.09
replies	0.37	0.25	1.00	0.49	0.39
forwards	0.27	0.00	1.00	0.99	0.38
media	0.29	0.12	1.00	0.79	0.34
mentions_unknown	0.23	0.00	1.00	1.28	0.42



(a) Raw user features.

(b) Processed user features.

Figure 6.6: Boxplots comparison for Telegram data set.

More insights showing the need for *Data Preprocessing* can be seen when pointing to the boxplots in Fig.s 6.6a and 6.6b showing the raw respectively standardized and normalized features. The most eye-catching insights in the raw data sets are the highly high mean values for `total_msgs`, `replied`, and `mentions`, lying outside the box, and the high amount of extreme outliers for `total_msgs` and `mentions`, which are depicted again with the aid of a log y-axis. Nevertheless, features like `replies`, `forwards`, `media`, and `mentions_unknown` already show satisfying *skewness* and *standard deviation*. Furthermore, the boxplot and the table for the preprocessed features show the effectiveness and necessity of *normalization* and *standardization*, a potent indicator for further steps such as clustering. Outliers are smoothed out again, but the characteristics of the features are maintained and set into equal bounds to facilitate a fair comparison of features with diverse domains.

6.4.2 Cluster Analysis

Clustering and *Cluster Analysis* are some of the most significant aspects of the pipeline presented in Chapter 4. The insights gained in Section 5.4.2 for the Twitter data sets play a central aspect in analyzing further data sets stemming from other social media services, e.g., Telegram. A *transfer* of the *methodology* from Twitter to Telegram can easily be realized. For the Telegram data sets, the approach's step from Section 4.4.2 was applied and yielded similar results as for the Twitter data sets when choosing the same configurations for clustering, as hierarchical agglomerative clustering enables aspects of traceability and explainability of data sets. Moreover, *Ward's linkage* showed the most suitable structure in the dendrograms for the Telegram data sets because of their multi-dimensional characteristic, and thus, outstanding roles could be mapped without significant cost to clusters. Even though the data sets can be clustered in their whole size, *sampling and a probabilistic combination of samples* is essential, as stability and explainability in hierarchies remain vital.

Representative samples using *Random Sampling* with varying sizes were created as part of the novel *Multi-Sampling and Combination Strategy*, addressing *certainty* and *explainability* of *user roles* as the data set in this use case is even smaller than the two dealing with tragic incidences from Twitter (cp. Table 5.1). The insights from Section 5.5 for choosing sample sizes were applied in this use case. On the one side of the spectrum, the lower bound is represented by terms of representativity of data sets, while the upper bound guarantees a valuable overlap in sampling, reaching a full coverage not too early. Thus, after some preliminary experiments, the sample size for the *Random Sampling* was determined to be 40% as it worked well.

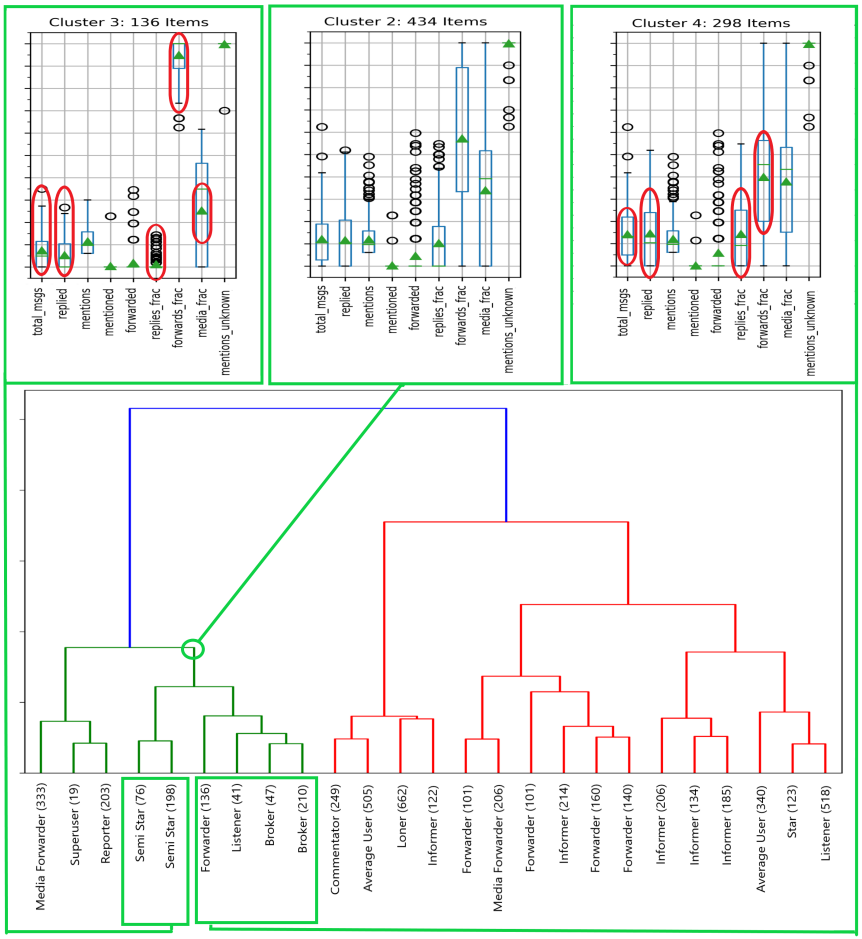
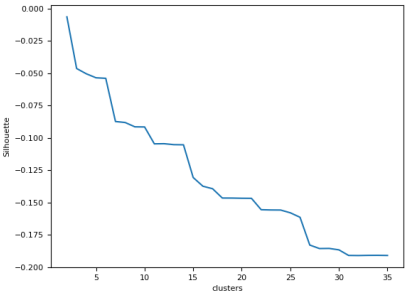


Figure 6.7: Dendrogram with boxplots for Telegram.

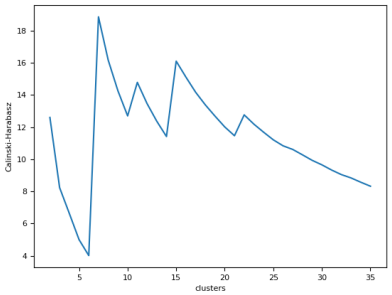
Pointing now to Fig. 6.7 showing the dendrogram of a 40% sample of the Telegram data set, one can see the structure of the dendrogram as well as feature deviations, which is a substantial aspect for the granularity of finding structures and their evolution, which is a central aspect for the analysis of clusterings. For further cluster analysis, the whole data set and several 40% samples of the data set were analyzed to verify if significant feature changes and patterns could also be found in the samples. As the Telegram data set is relatively small compared to most of the Twitter data sets from

Table 5.1, the samples have to be chosen more carefully than those of the Twitter data sets from Table 5.10, to gain *representativity* as well as *stability* and *certainty* of *user roles* when combining the samples. Moreover, a set of varying samples is vital for building and verifying training data, as training data sets need an adequate amount of tuples. Furthermore, a *ground truth* is required.

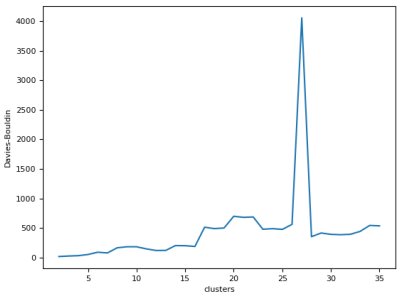
Finding possible best clusters starts again with the analysis of *internal quality metrics* as in Section 4.4.2 focusing on the *Silhouette Coefficient* (Section 2.6.2.1), *Davies-Bouldin Index* (Section 2.6.2.2) as well as the *Calinski Harabasz Index* (Section 2.6.2.3).



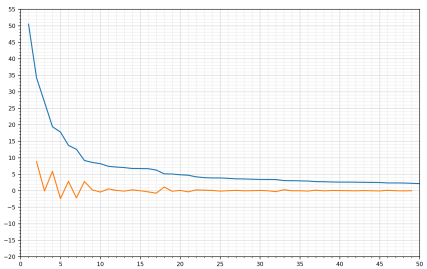
(a) Silhouette of 40% sample.



(b) Calinski-Harabasz of 40% sample.



(c) Davies-Bouldin of 40% sample.



(d) Elbow for 40% sample.

Figure 6.8: Comparison of several cluster evaluation metrics in Telegram.

As already figured out in the Twitter data sets, the *internal cluster analysis* is not suitable enough, as the *Silhouette* (Fig.6.8a) as well as the *Davies-Bouldin Index* (Fig. 6.8c) provide best values for 2 clusters, whereas the *Silhouette* score always is below 0, indicating a low explanatory power. In contrast the *Calinski-Harabasz Index*

(Fig. 6.8b) has its peak at 7 clusters and several local maxima at 12 and 15 clusters, indicating a slightly better explanatory power in terms of cluster analysis. Even the *Elbow* method in Fig. 6.8d provides only the starting point for the analysis as only two generalized clusters are delivered (cp. dendrogram in Fig. 6.7).

The focus for the cluster analysis is purely on the *effect size based feature analysis* within the *depth-first search*, presented in Section 4.4.2 (cp. Definition 33) and further discussed in the analysis of the Twitter data sets in Section 5.4.2 and 5.6. In contrast to Twitter, the significance criteria for the Telegram data set finding the most suitable configuration was adapted as follows. While in the Twitter data set, the significance criteria were finding at least two large effects or, one very large or one huge effect or an *average Cohen's d* more significant than 0.1, the analysis for the Telegram data set delivered better results when only considering the *average Cohen's d*, as the *deviations* in distinct features were not as significant as in Twitter. Several *thresholds* were examined, resulting in varying results w.r.t. the number of clusters. Choosing the average Cohen's d smaller delivers more clusters, as a value of 0.2 delivers 31 clusters, while an average Cohen's d of 0.6 delivers only 13 clusters. A significant change can be detected between an average Cohen's d of 0.2 and 0.3, which results in a refinement testing more samples. Finally, setting the average Cohen's d to 0.1 delivered the most suitable results, as fine-grained user roles for all evaluated samples were provided. The requirements for finding more significant deviations in only one or two specific features did not deliver the best results for all tested samples. Furthermore, the *lower bound* in the *depth-first search* was reduced from 5 to 1 compared to the Twitter data sets, stopping the search when going beyond this *threshold*, as significant changes were observed below, tracing those significant changes back to the nature of the Telegram data set. This aspect avoids creating too many small clusters in the subtrees.

Finally, the *effect size-based depth-first search* in the Telegram data set delivered similar results as in the Twitter data sets. Even though several parameters need to be adjusted for all of the tested samples, a suitable cutoff in the dendrogram, leading to well-separated clusters, could be found, making this approach the most valuable compared to traditional cluster evaluation metrics.

6.4.3 Manual Class Labeling

Once a fitting configuration is identified during the *Cluster Analysis* stage, the *Manual Class Labeling* can confidently be executed. For this step, several 40 % samples were analyzed with the aid of the *Cluster Hierarchy Analysis Tool* (cp. Fig. 5.9) as well as the *Depth-First Search* approach from Definition 33. Similar to the analysis of the Twitter Data (cp. Fig. 5.10 and Table 5.6), sets of well-separated user roles from the literature were delivered for the Telegram use case. In particular, only some of the

established roles could be adopted entirely, as user behavior in Telegram is somewhat different than in Twitter, while others had to be redefined or were completely new.

The general strategy, introduced in Section 4.5 for analyzing clusters in the Telegram data set, did not distinguish much from the process for analyzing Twitter data sets in Section 5.4.3. However, it was cut short, as findings from the Twitter data sets' analysis paved the way for both adjusting the cluster analysis tool from Fig. 5.9 as well as creating and validating training data, reducing human involvement at least partially in the initial step of finding user roles in an entirely new data set. Nevertheless, as most of the features from the Twitter data sets (cp. Fig. 5.3) are present in the Telegram data set, a mapping of the roles from Twitter to Telegram is easily possible.

To achieve the desired outcome while considering fine-grained structures in all samples, the ideal number of clusters is between 15 and 35. This range is based on *manual class labeling* insights from the Twitter use case discussed in Section 5.4.3. Like the Twitter data sets, the dendrogram mostly delivers between two and five subtrees with generalized user roles again, depending on the samples. In Fig. 6.7, most of the occurring user roles in the subtrees are similar and stem from a *generalized user role* with some exceptions, such as the clusters describing the *Media-Forwarder* or *Star* users. These generalized user roles can be mapped due to their allocation into three *coarse-grained user roles*, which can be seen in Fig. 6.9 and Table 6.4.

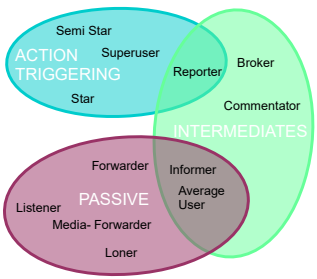


Figure 6.9: User roles in Telegram.

User roles in the *Action Triggering* group can be classified as *Star* users, *Semi Stars*, and *Superusers*. *Star* users have higher feature values than the average, especially for `total_msgs`, `mentioned` and the `reply rate` to other users' messages, but tend to refrain from *forwarding* or *replying* to messages. *Semi Stars* are similar, have more `total_messages`, and are *more active* in forwarding and replying, but are not `mentioned` as often as *Star* users. *Superusers*, describing a completely new user role,

are usually *admins* or *moderators*, have the most `messages` and `mentions`, and often `forward` content, including `media`. The observations made for the Twitter data sets are similar, whereas the quotas are slightly higher. Compared to the *Superuser*, the *Reporter* is similar but *triggers* fewer `reactions`, sharing similarities with the *Action Triggering* and *Intermediates* group. Moreover, *Reporters* tend to `forward` many `messages` and `media`, whereas their content is rarely `forwarded`.

Table 6.4: User roles and their characterization: $\approx$ shows closeness to other roles, $\downarrow/\uparrow$ feature deviation from close role/whole data set, $\searrow$ / $\leftrightarrow$ / $\nearrow$ changes over time

	Role	Characteristics	Frequency
passive interm. act. triggering	Star	$\downarrow$ replies, forwards $\uparrow$ mentioned, replied	0.5–2%
	Semi Star	$\approx$ Stars, $\uparrow$ total_msgs, replied	3–5%
	Superuser	$\approx$ Semi Star, $\uparrow$ total_msgs, replied, forwarded, forwards, media	1–2%
	Reporter	$\approx$ Super-User, Semi Stars, $\uparrow$ total_msgs, replied, mentions, forwards, media	2–4%
	Broker	$\approx$ Commentator, Reporter $\uparrow$ replied	4–6%
	Commentator	$\approx$ Reporter, Broker $\uparrow$ replied, total_msgs, replies	8–12%
	Informer	$\approx$ Average User $\uparrow$ replies, media	10–15%
	Average User	$\approx$ Informer $\uparrow$ replies $\downarrow$ total_msgs	4–8%
	Forwarder	$\approx$ Media-Forwarder $\uparrow$ forwards, mentions, $\downarrow$ replied	8–12%
	Media Forwarder	$\approx$ Forwarder $\downarrow$ replied, mentioned, forwarded $\uparrow$ forwards, media	10–15%
	Loner	$\approx$ Listener $\downarrow\downarrow$ total_msgs, replied	8–12%
	Listener	$\approx$ Loner $\downarrow$ total_msgs	20–23%

Addressing the second *coarse-grained* group of *Intermediates*, *Brokers*, *Stars*, and *Semi-Stars* send similar amounts of `messages`, with *Brokers* receiving an adequate amount of `replies`. However, *Brokers* receive fewer `forwarded` `messages` and `media`. They are more active and frequently `mentioned` than other users, giving them a more prominent network position. *Commentators* have a slightly higher amount of written `messages` and a high amount of `replies`. They `forward` fewer messages and produce less `media` content but trigger more `replies`.

Focusing now on user roles between *Intermediates* and *Passive Users*, the *Average User* perfectly represents the average boxplot. `Total messages` are low, with a higher amount of `replies` and lower amounts of `forwards` and `media` messages. *Informers* are similar to *Average Users* w.r.t. to `written messages` and `replies`, whereas `messages` of *Informer* have a high amount of `replies` and `media`. Representatives

of the *Intermediates*, such as the *Commentator* and *Informer*, are user roles with a higher increase in the Telegram data set than in the Twitter use case. While the *Average User* was the most represented user role in the Twitter data sets, they tend to be less frequent in Telegram.

Pointing now to the group of *Passive* user roles, *Forwarders* are characterized by their immense amount of `forwarded` messages, the ability to *trigger* hardly other users, and cause little *attention*. Similar to the user role of the *Forwarder* is the *Media-Forwarder*, characterized mainly by a very high amount of `media` in their (`forwarded`) `messages`. In contrast to those two user roles, the following two are the most *passive* in the data set. While *Listeners* produce at least a below-average amount of `messages` and generate some `replies` to their messages, *Loners* tend to mainly consume content and refrain from *reacting* and *participating* in other users' content. While only the *Forwarders* and *Listeners* were omnipresent in the Twitter data sets, almost all *passive* user roles are equally present in the Telegram data set.

Unsurprisingly, Telegram differentiates from Twitter, as the user behavior in Telegram is more *defensive* than Twitter's. Channels, where only individuals deliver content to their subscribers, dominate the data set. In addition, there are also groups where many users can share their thoughts. The insights of the Twitter user roles analysis delivered for almost all samples a comprehensible hierarchical structure from *coarse* to *fine-grained user roles*. Focusing on the results of the Telegram analysis, several samples did not have a comprehensible structure at all, as some fine-grained user roles appear in non-related subtrees representing coarse-grained user roles. These observations were also confirmed in some samples' classification processes. Nevertheless, most of the user roles were comprehensible, and the classification results in Section 6.5 confirm the insights of this section for the Telegram use case.

In order to construct a dependable *classifier*, it is imperative to conduct a thorough examination of user roles. While this task may be arduous and time-consuming, it is an essential step that establishes the groundwork for the classifier and supplies the requisite training data for *classification*. Due to the necessity of *expert* insight during the initial trials, it is impossible to circumvent the manual analysis and preparation stages required for constructing a classifier. Compared to the pipeline of the Twitter data sets, the manual analysis and preparations for building the classifier cannot be cut short, as initial trials always need expert knowledge.

6.4.4 Building a Classifier

As outlined in Section 4.6, *classification* is the next significant step in the pipeline presented in Chapter 4. Since the Telegram data sets are entirely new, a new classifier must be established for automatic classification. The foundation for building and

proving the suitability of a classifier was laid in the previous section 6.4.3, where clusters were *labeled manually* with the aid of user roles from literature and those established for the Twitter data sets. In contrast to the analysis of the Twitter data sets, where a more *manually-driven approach* was performed for creating *training data* in an explainable way, for the Telegram data sets, the focus was on using Semi-Supervised Learning (SSL) and Active Learning (AL) to relieve the *analyst* in the *incremental creation process* with the insights gained from the Twitter use case. Moreover, *ground truth* helps to validate the training data, as outliers can be separated more easily when ground truth labels and prediction diverge.

Four representative 40% samples, already evaluated and manually labeled in the sections before, were considered for the training data set as a starting point for the classification-building process. The *feature vector* representing the mean of all data points for each considered cluster was calculated and incorporated with a user role ID into the training data set. In contrast to the process of building training data for Twitter data sets, where most of the suitable cluster means were added manually and proved with cross-validation and dimensionality reduction techniques such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), for the Telegram data, a more AL and SSL driven approach, which was introduced in Fig. 2.2 of Section 2.5 was considered, to cut short the human effort and intervention in the manual-driven steps of the approach.

To complete this task, the KD process was utilized to cluster a sample initially and analyze it with the *depth-first search* presented in Def. 33 using the specifications gained in Section 6.4.2 and label it using the initial training data. While a broad range of classifiers were used to classify the Twitter data sets, only the most suitable from the grid search for the Twitter data were considered for the Telegram data set, leading to using the Support Vector Machines (SVM) classifier, as it provided the best results for the Telegram data. In addition to the mainly *manual-driven approach* presented in Fig. 4.7, further useful tools, such as *distance calculation* between cluster means of the training data set, maximizing the distances between clusters were used to improve the *training data* creation process. Moreover, the *Standard deviation* and the *pooled Cohen's d* for features are also estimated to decide if a cluster fits the training data.

After several steps of the AL approach, the clusters with representatives of the training data set have to be analyzed if there are outliers. To accomplish this task, the *distances to the mean* of each cluster are calculated, and any outliers with high distances are *eliminated* from the training data set. In contrast to the Twitter data sets, where data points were mainly eliminated after visualizing the training data set by dimensionality reduction techniques, the process for the Telegram data sets was aided more methodologically. Cross-validations are performed after several steps to prove the suitability of the training data, examining if the accuracy is growing. Also,

consistently reviewing the training data with dimensionality reduction techniques is very helpful in spotting possible outliers in the training data.

For creating valuable training data, in total, 20 samples were created. In contrast to the approach for the Twitter data set from Section 5.4.4, where all of the samples were manually labeled in advance, this approach does not need as much *human intervention* as only initial samples need to be tagged, and the further decisions of the classifier need only to be validated in case of need with the tool from Fig. 5.9, leading to faster and more automatic creation of training data. Moreover, the insights for the classifiers gained in Section 5.5 for the Twitter data set helped to cut short the process of *instantiating and assessing the classifier*, as no parameter tuning is necessary.

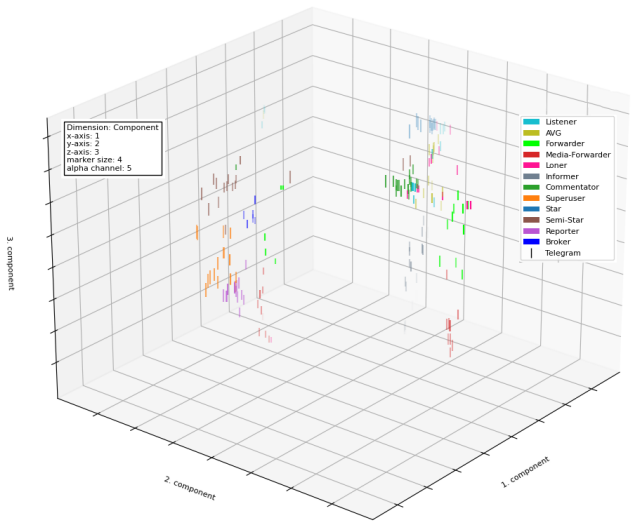


Figure 6.10: PCA of training data from *Telegram*.

The PCA and LDA in Fig.s 6.10 and 6.11 show the *enriched training data* after evaluating and adding 20 samples. One can see that most of the user roles are pretty well separated, such as the Reporter, Superuser, Semi-Star, Informer, or Star. At the same time, Forwarder or Media-Forwarder is split into two sets. These observations lead to breaking the respective user roles into two separate ones.

The PCA was again visualized with the aid of a *3-dimensional coordinate system* underpinned with additional specifications such as *marker size*, *marker style*, and *alpha channel* representing the most significant components in each dimension, which can be seen in Table 6.5. Again, the first three components representing the 3D coordinate

system combine more than 90 % of the total variance, whereas the top correlated features are *to_ unknown*, *forwards*, and *replies*.

Table 6.5: Variances & top 3 features for six components in PCA of training data set.

Component	Variance	Top Features
x-axis	58.75%	to_unknown, forwards, replies
y-axis	18.91%	to_unknown, forwards, replies
z-axis	11.95%	media, replies, forwards
marker size	4.69%	replied, total_msgs, to_unknown
marker style	4.19%	forwards, media, replies
alpha channel	1.09%	forwarded, replied, total_msgs

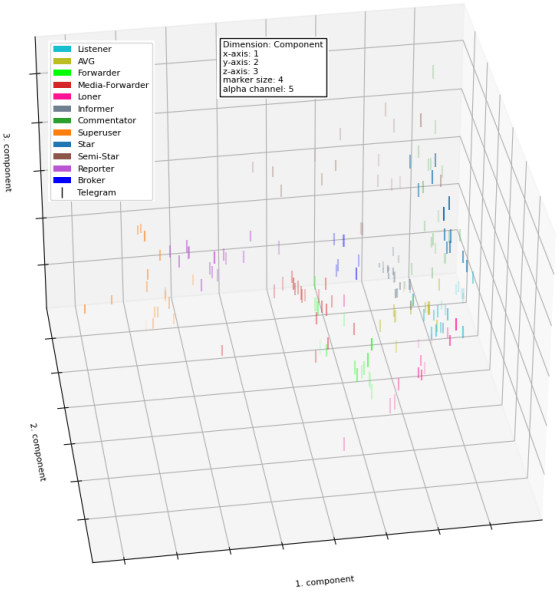


Figure 6.11: LDA of training data from *Telegram*.

The final training data set contains 202 cluster means representing the following distribution in Table 6.6. In comparison, *Forwarder* and *Average User* were represented dominantly in the Twitter data set(cp. Table 5.8), while a more balanced distribution is given for the Telegram data set.

Table 6.6: A priori distribution of user roles in the training data set

Listener	**AVG**	**Forwarder**	**Media-Forw.**	**Loner**	**Informer**
20 (9.90%)	15 (7.43%)	20 (9.90%)	20 (9.90%)	16 (7.92%)	20 (9.90%)
Commentator	**Superuser**	**Star**	**Semi-Star**	**Reporter**	**Broker**
20 (9.90%)	18 (8.91%)	12 (5.94%)	17 (8.42%)	16 (7.92%)	8 (3.69%)

As the training data set is built, it has to be validated. In Fig. 6.12, the matrix shows the validation of just six samples at the beginning of the training process. While the labels in the columns describe the actual classes, the labels in the rows represent the classes a cluster got allocated to a specific user role by the classifier. While the *precision values*, which can be seen at the end of each row for all of the user roles, except for the *Average User*, receive rather high values, the recall values do not yet receive high values, as the *Average User*, *Informer*, *Star*, and *Semi Star* need to be improved. Nevertheless, the classifier already has quite a high rate of correctly classified user roles of 84.14%, showing that the insights gained from building training data sets for the Twitter data sets can be transferred straightforwardly to Telegram.

The matrix in Fig. 6.13 shows the classification results of a more specified classifier, including training data from 20 samples. The clusters from those 20 samples show a relatively similar overall ratio of correctly classified users of 85.03% as in Fig. 6.12 can be seen. Especially the precision values for the user roles, except for the *Listener*, *Loner*, and *Broker*, show adequate values. In contrast, *Average Users* were often classified as *Listeners*, *Forwarders* as *Loners*, and *Forwarders* as *Brokers*. Also, the recall values are for most classes quite good except for the user roles *AVG Users* and *Broker*, as *Average Users* are often classified as *Listener* and *Broker* as *Media-Forwarder*.

Finally, the classifier delivers good results for most of the classes. Only a few refinements are needed for the training data to get more precise results, as already mentioned in the analysis of the training data using PCA. Also, the creation process of training data was optimized compared to the creation of training data for Twitter data sets, as an AL and SSL approach was utilized. While, especially in the beginning of the building of a classifier, much manual labeling was essential for the Twitter data sets, only a few samples needed to be classified manually, cutting the process short. The main work of an expert remains now only for corrective purposes, as only suggestions need to be accepted or declined.

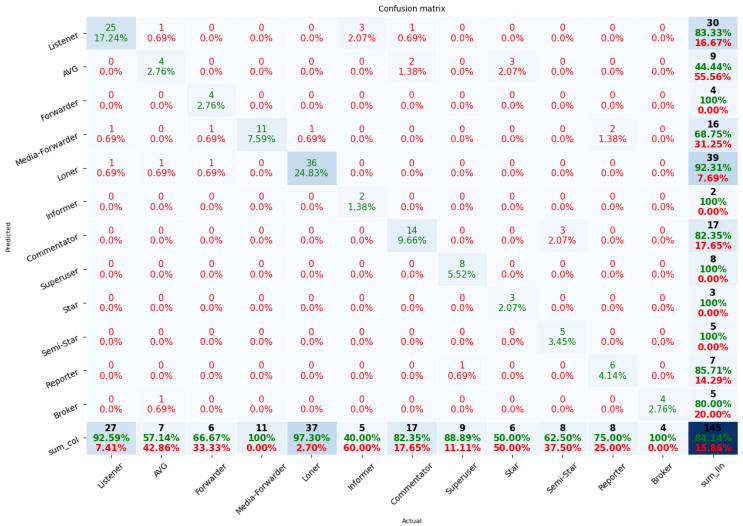


Figure 6.12: Confusion matrix using the initial Training data set.

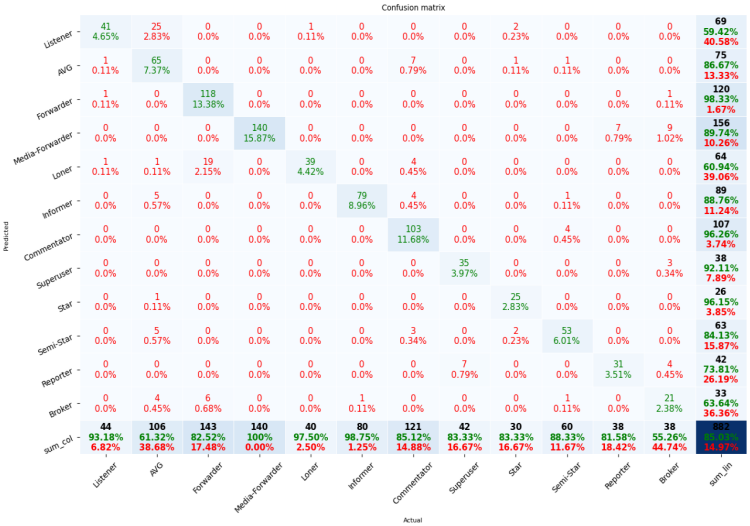


Figure 6.13: Confusion matrix using the final training data set.

6.5 Analysis of User Roles

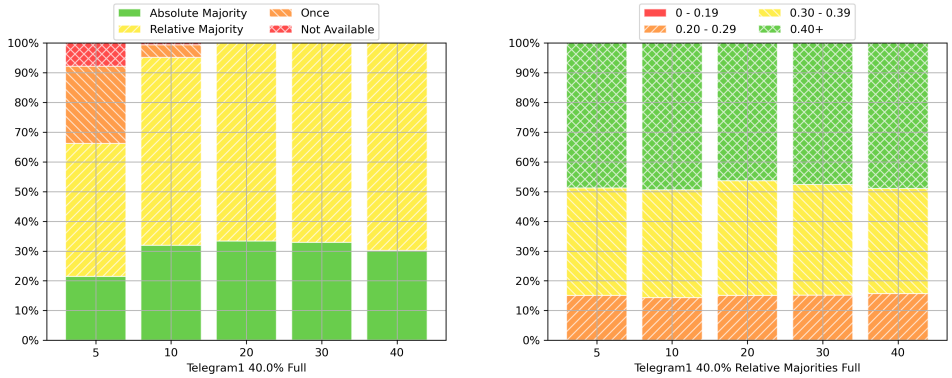
After building and training the classifier in Section 6.4.4, they can work precisely for most user roles in the clustered samples stemming from the same data set as the training data. While the manual labeling process delivered the quotas of user roles (cp. Table 6.4), stable percentages can be gained when using the classifiers on a set of samples, which are combined afterward as stated in the novel *Multi-Sampling and Combination Strategy* from the KD approach introduced in Chapter 4.

To show the benefits of the novel *Multi-Sampling and Combinations Strategy*, the following questions, which were also discussed for the Twitter data sets, are answered in the following section:

- Influence of sample sizes and number of samples on the coverage.
- Impact of multiple captured users on stability and certainty of user roles.
- Correlation of user roles stemming from the same generalized role w.r.t. second best user roles.

In addition to the 20 samples used for creating the training data from Section 6.4.4, 20 more samples were created using the Random Sampling strategy. All further steps, such as clustering, cluster analysis, and classification, were applied to the 40 samples described in the previous sections. Afterward, the combination of users will be performed to analyze the questions.

Coverage of the Data Set As the Telegram data set is relatively small, the sample size has to be chosen more carefully than in the Twitter data sets to guarantee the samples' representativity. A good comparison to the random samples of the Telegram data set is samples from the *Paris 2015 Twitter* data set (cp. Fig. 5.20a and Fig. 5.20b) even though two other sampling strategies were chosen. The general observation in Fig. 6.14a shows that full coverage is reached quickly between 10 and 20 combined samples. Also, the number of considered users is growing pretty fast for a growing number of combined samples. Focusing only on users who got at least considered twice, the amount for users who got considered twice is very prevalent when combining five samples as Fig. 6.15 shows. 80% of the users are considered at least six times when combining 20 samples. In comparison, 80% are considered at least 14 times when combining 40 samples, indicating that the degree of stability is reached at about 20 samples. This observation overlaps mostly with the considerations made in analyzing the Twitter data sets, even though this experiment did not consider tuning the sampling strategies. A complete coverage is reached fast while tuning the sampling strategy is currently optional.



(a) *Telegram* 40% Random Sampling.

(b) *Telegram* 40% relative majorities.

Figure 6.14: Coverage of 40% samples from Telegram data set.

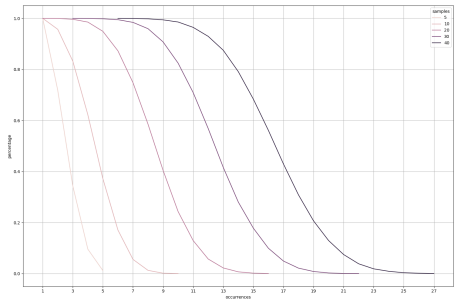
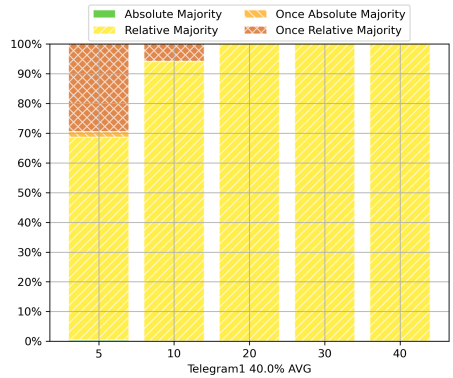


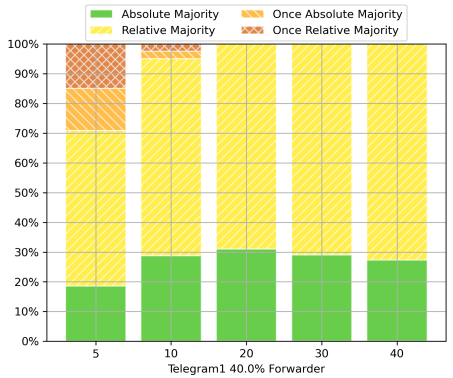
Figure 6.15: *Telegram* 40% user amounts for Random Sampling.

Stability & Certainty of User Roles While the *Twitter Paris 2015* data set (cp. Fig. 5.20) reaches a high amount of stable users having an absolute majority, those with only a relative majority are even close to an absolute majority. In contrast, the amount of users having an absolute majority is lower in the Telegram data set, as Fig. 6.14a reveals. Beyond that, the number of users with an absolute majority stagnates between 10 and 30 and even decreases between 30 and 40 samples, showing a degree of saturation. Users captured multiple times after 20 sample combinations, as in Fig. 6.15, indicate stability and certainty. In Fig. 6.14b, most users with a relative majority are close to achieving an absolute majority, shown by the green and yellow bars representing over 80% for increasing sample sizes.

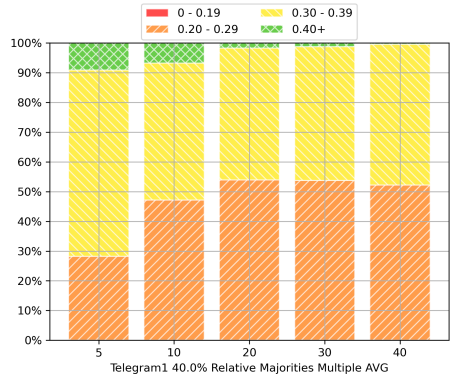
As the entire data set revealed promising results for stability and certainty over the whole data set, the focus is now on specific user roles. As described, some user roles did not have outstanding results in the clustering process, such as the *Average User*, *Broker*, or *Star*. Concentrating on *Average Users*, Fig. 6.16a shows that they have no absolute majority when combining more than 20 samples. In comparison, some users have an absolute majority but only got captured once when combining up to 20 samples. Furthermore, Fig. 6.16c shows that some users are not that close to an absolute majority. Still, most users (almost 50%) have adequate quotas for the best user role, showing that the classification generally works but needs another iteration for the average user to classify them more precisely.



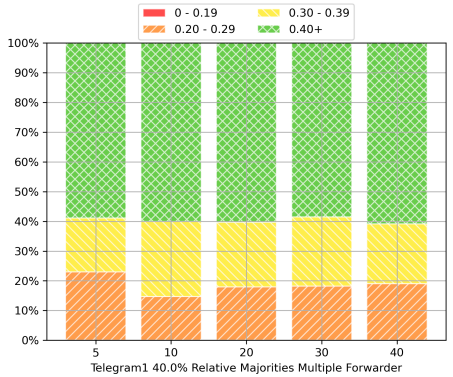
(a) *Telegram* AVG User.



(b) *Telegram* Forwarder.



(c) *Telegram* AVG User relative majorities.

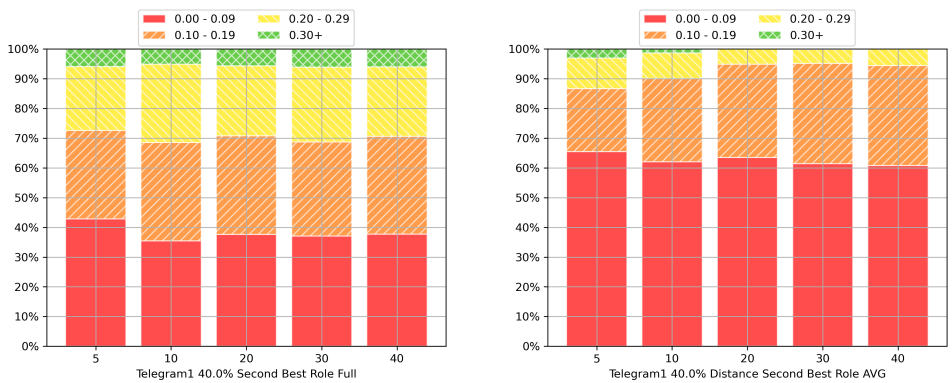


(d) *Telegram* Forwarder relative majorities.

Figure 6.16: Coverage of 40% Random Sampling from Telegram for several user roles.

Similar behavior can also be observed for *Brokers*, *Stars*, and *Loners*. In contrast to *Average Users*, *Forwarder*, *Commentator*, *Informer*, *Listener*, *Media Forwarder*, *Reporter*, *Semi Star*, and *Superuser* show compelling results, as can be seen in Fig. 6.16b for the Forwarder. The behavior is similar to the whole dataset (cp. Fig. 6.14a and 6.14b), but the number of users with an absolute majority is lower than in some Twitter datasets. However, users with only a relative majority are close to an absolute majority, as in Fig. 6.16d. These aspects can also be found in the other user roles with pretty good classification results in Fig. 6.13. Nevertheless, the overall score of the classifier can be reflected in the plots, as both absolute and relative majorities show the presence of stable user roles.

Correlation of Second Best User Roles The third question, considering the correlation of the second best user roles stemming from the same generalized user role as the best user role (cp. Fig. 6.9), will now be answered. In the previous section, the precision and recall revealed misclassification for *Average Users* classified as *Listener*, *Forwarder* as *Loner*, and *Broker*, as well as *Broker* as *Media-Forwarder*, showing that those misclassifications arise mainly from generalized user roles. Focusing on the combined samples again, the general quotas of the second best user role show almost stable values when increasing the number of samples to incorporate. While only 30 percent of users second best roles have a higher distance of 0.2 between the best and second best user role, a higher amount of each 30% has only a distance of up to 10% respective up to 20% between the first and second best user role as Fig. 6.17a shows. Even closer distances can be monitored for the *Average User* in Fig. 6.17b.



(a) Full data set.

(b) AVG user.

Figure 6.17: Comparison of second best user roles of 40% samples with Random Sampling.

Even though the classifier works adequately for the *Average User* but needs improvement, the matrix in Fig. 6.18 shows that *Average Users* tend to have a second-best user role from the same generalized user role as *Informer* and *Listener* are the most represented user roles. The user role of the *Broker* shares more similarities with the user role of the *Semi Star* and *Average User*, showing that this user role needs refinement. Also, *Reporters* who have the role of the *Media-Forwarder* as the second-best user role and *Stars* who have the role of the *Listener* as the second-best user role show unexpected results. For all other user roles, the highest quotas of the second-best user role correlated with the best user role stemming from the same generalized user role. Finally, a refinement of specific user roles of the classifier can counteract these effects.

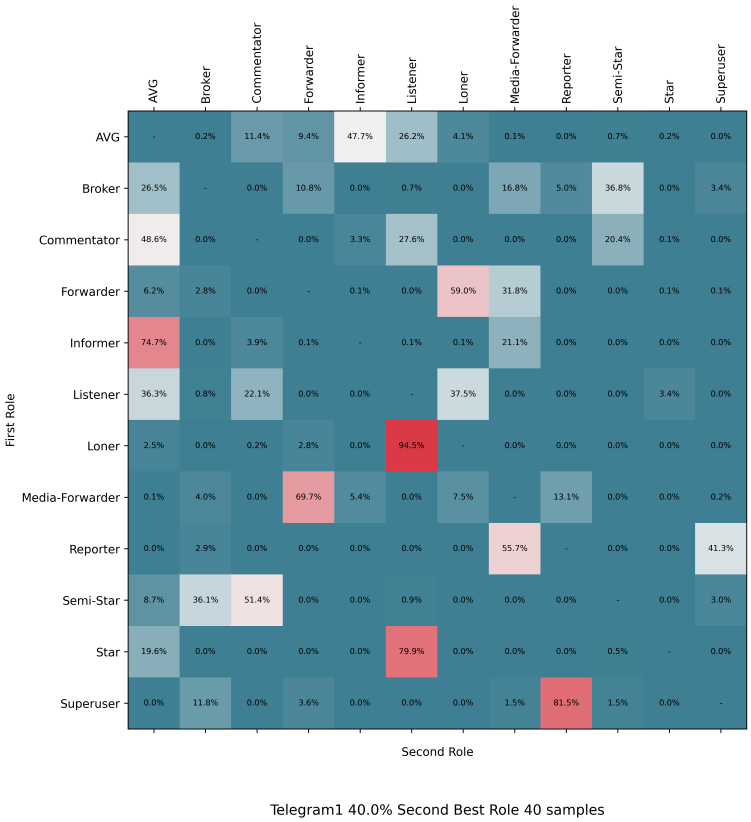


Figure 6.18: *Telegram* 40% second best roles.

6.6 Related Work

This chapter analyzed a data set from the instant messaging service Telegram. As the focus on Telegram was led in the last few years during the Coronavirus pandemic, less related work has been published. Work concentrating on general user behavior representing Telegram data sets has not been published yet. Work such as the publications of Lazaridou et al. [LNN16] or Chu et al. [Chu+10] focused on data sets from Twitter. In addition, this work adapted several strategies for analyzing user behavior, leading to fine-grained user roles. In general, there is work analyzing political parties triggering user behavior in social media, published by Gim et al. [Gim+18] who examined the influence of the German political party Alternative für Deutschland during the elections for the German Bundestag in 2017, resulting in an increasing impact triggered by the party. This work found that political discussions are omnipresent in social media services. Telegram has widely been used for (political) discussions related to fake news and disinformation in scientific work, e.g., the Coronavirus pandemic, as discussed in the work of [NL20]. Moreover, analyzing hate speech (Wich et al. [Wic+21]) and political extremists (Yayla et al. [YS17]) are also widespread phenomena in Telegram. While Hashemi et al. [HZC19] investigated the quality of Telegram groups w.r.t. user role behavior, this work focuses on a more fine-grained analysis of distinct user roles. Further related work, such as Dargahi et al. [DN+20], concentrate on exploring viral messages, while Sutniko et al. [Sut+16] and Hoseini et al. [Hos+20] focus on feature exploration in Telegram. Other work, such as Baumgartner et al. [Bau+20], or Urman et al. [UK20], concentrate more on exploring retrieval of Telegram data sets as online social movements, protests, political extremism, and disinformation is part of their work.

6.7 Conclusion

Finally, the core part of the methodology introduced in Chapter 4 was successfully applied to the Telegram data set. Almost all steps of the KD approach pipeline were executed as in Chapter 5, showing that an application is easily possible. Starting with *Feature Engineering*, consisting of *Feature Selection*, in iterative steps, the most suitable features were found, *normalized*, and *standardized* equally as in the Twitter data sets, focusing on an adequate degree of *correlation*, *Log-standardization*, and *Min-Max normalization*. After *Clustering* and *Cluster Analysis* exploiting the same techniques as for the Twitter data sets, manual analysis was applied, understanding the specifications of the data set and laying the foundation for the following classification process. Compared to the Twitter data sets, the classification process was changed by exploiting an automatic AL and SSL-driven approach, where human supervision is

mainly essential at the beginning of the building process of the classifier and a suitable ground truth for the validation process.

Also, the sampling strategy exploiting *Random Sampling* was applied, as several distinct samples are needed for training data. A *combination* of the *samples* was also performed. Even though it is a tough choice to create *representative* samples for smaller data sets, samples with a size smaller than 40 % are almost not suitable due to small clusters arising after the clustering step. The results of the analysis and the most widely confirmation of the research questions presented in Section 6.5 show that the whole pipeline is also very suitable for Telegram data sets even though the classifier needs another iteration for refining the user roles.

Finally, each step can be cut shorter, as much knowledge was gained when proceeding with the Twitter data sets. While *Feature Engineering* depends on the features and the characteristics of the data sets, *Normalization* and *Standardization* techniques could be easily adapted from the Twitter data sets. Also, clustering and cluster analysis were straightforward, only tuning the parameters for the effect size in the depth-first search. The most elaborate step was building an entirely new *Classifier*, as the Telegram data set was entirely new. However, with the knowledge from the Twitter analysis, the same configurations were tested and delivered suitable results. Moreover, refining and tuning the whole *classifier-building process* helped cut short the entire process. Most of the fine-grained user roles elaborated in the Twitter data sets could also be found in the Telegram data set, enriching them with more specific user roles and showing that the proposed approach applies to entirely new data sets.

As this chapter covered a use case in an early stage of Chapter 4s approach, some steps were only applied, while some needed sharpening and others needed to be performed entirely. While the *classification* needs only refinement in terms of user roles, the *classifiers application* to a variety of further data sets needs to be performed to examine the suitability of the classifier and the detection and precision of the user roles examined in this chapter across data sets. Moreover, the data sets' small size allows only preliminary results for the novel *Multi-Sampling and Combination Strategy*. Thus, the strategies' suitability in terms of stability and coverage of user roles across a variety of topically related but also non-related as well as larger data sets needs to be proved. Also, trends of user role evolution over time, as well as the model-building process, are fascinating areas to pursue for the Telegram data sets use case.

Chapter 7

Analyzing Cascade Shapes from Twitter Data Sets

Träumen Androiden von Datenkraken,
Oder dann doch nur von elektrischen Schafen?
So vernetzt und doch allein,
Du musst das Leben nicht verstehen,
Kauf einfach ein!

Callejon - *Die Fabrik*

In contrast to the both use cases before, this chapter addresses the application of parts of the pipeline from Chapter 4 to an entirely new use case. Analyzing information cascades in terms of graph embeddings is the central point of this chapter. As features are hidden due to the embedding techniques, they are compared against another strategy for summarizing graphs with collapsing and well-known graph metrics.

7.1 Motivation and Contributions

While the previous two Chapters 5 and 6 covered *conceptually similar problem settings* with the structural analysis of fine-grained user roles based on the approach presented

in Chapter 4, the use case in this chapter dealing with analyzing *graph embeddings*, is an *entirely different use case* from the area of *graph analysis*. In particular, the analysis of graph embeddings deals with entirely *different features*, being *not interpretable* or *comprehensible* as they are latent over the whole process. Moreover, the number of features remains tuneable as the dimensions in the embedding technique are not fixed. Furthermore, in this chapter, the analysis of fine-grained structures is central, but the focus is not on user roles but on *shapes of cascades* in terms of *information diffusion* displayed as *graphs*.

A better understanding of information diffusion can be gathered by studying the patterns and relationships within the data sets describing user behavior and interaction. Information diffusion in social media and social networks is a viable and well-established approach to investigating the influence of messages and content on other users. Since news may be spread within only a few minutes around the world and trigger other users, whole cascade shapes and the containing users in those cascades are of research interest in the work of Guille et al. and Taxidou and Fischer [Gui+13; TF13; TF14]. Analyzing information cascades can be accomplished with the aid of *graphs* and *analysis*. As *graph analysis* is a well-known research area, graph embedding is an innovative strategy, targeting to describe whole information cascades into single vectors. The mapping into single vectors is the basis for the further steps of the Knowledge Discovery (KD) process from Chapter 4.

Fig. 7.1 shows two sample graphs, each representing a retweet cascade from the social-media service Twitter. Each of the graphs has a root node (red), representing the source of the cascade, i.e., the original Tweet, while the further colors yellow, green, blue, and black represent the shortest path from the root to the specific node for a path with a length of one, two, three and greater than three representing the diffusion of the original tweet.

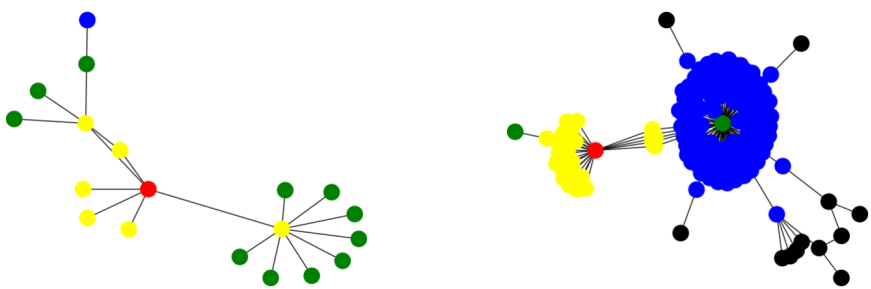


Figure 7.1: Graphs representing retweet-cascades in Twitter.

For this work, data sets from the social media service *Twitter* are investigated, providing *plain messages* and information about *retweet status*. Thus, a *cascade structure* must be created, visualizing the *flow of information* considering the influence of users. Considering cascades, each node represents a user, while the edges visualize the messages of the social graph the users got influenced. Furthermore, *retweet cascades* in Twitter data sets have a root, representing the message's creator. From this root, the message can be forwarded by one or more users, leading to *nodes* of the *forwarders*. Moreover, both initial and forwarded messages can be forwarded inductively, creating a complex information cascade as a social graph with distinct graph structures.

Depending on the number of users and their behavior, a cascade can have a more *solid structure* when messages are spread evenly only from the *root*, representing a *star-like shape*. In contrast, a message can be forwarded repeatedly, leading to *chains*. Also, several in-between structures where *chains* and *stars* can be combined are possible, leading to several *complex structures* that can be challenging to investigate. There can be quite trivial but also complex structures, including many *nodes* and *edges*, forming different shapes of graphs, also leading to high costs for graph analysis. A suitable solution to handle this problem is graph embedding, where whole cascades can be visualized in a low-dimensional space. Fine-grained graph analysis can be affected with the aid of graph embeddings, as several manifold graph shapes exist in this use case. As analyzed in Chapter 5, several distinct user roles exist within a data set. Information cascades arise, w.r.t. the users and their roles creating the initial messages, as well as the users the messages got retweeted. Thus, many different manifold graph structures exist. While the approach in this chapter deals with the analysis of such graph structures, the correlation between user roles and graph structures remains a worthwhile research topic for future work.

The analysis of such embeddings became a well-known research area starting in 2015 until now. A central aspect of graph analysis is to describe *graphs* as *vectors* so they can be *clustered* and *analyzed*. Two central approaches were investigated in the last years, describing graphs with a *summarization* of *graph metrics* or *embedding nodes*, *edges*, or whole *(sub)structures* of the graphs to create a compact characterization in the form of a vector.

As there are several different strategies to analyze information cascades, a strong motivation was to adapt the proposed approach from Chapter 4 on *graph cascades*. For the use case in this chapter, three different strategies to analyze information cascades were investigated due to their suitability to the approach to encapsulate similar structures using clustering. While two strategies entirely rely on opaque features, one deals with analyzing traditional graph metrics as a comparison. The creation process of cascade shapes and the transformation into a vector can be seen as a *preprocessing* step. In addition, in further analysis, *clustering* and *cluster analysis*

play a pivotal aspect in this use case. As the last step, *manual classification* of the clustered graph structures was performed, being a good starting point for further investigations such as *classification*.

For further analysis, the following research questions are discussed and analyzed in this chapter:

- Is a fine-grained structural analysis possible when inspecting retweet-cascades?
- Is the general KD approach successfully applicable for each embedding technique?
- Does the lack of explainability influence the KD approach?
- Is the general approach transferable to new data sets?

7.2 Background

After motivating and introducing the main idea and the contributions in this section, further background on *graphs*, *cascades*, and *graph embedding* techniques are presented.

7.2.1 Graphs & Cascades

A graph $G = (V, E)$ with a set of vertices $V = \{v_1, v_2, ..., v_n\}$ and edges $E = \{e_1, e_2, ..., e_m\}$ will be embedded into a *d-dimensional space* with the condition $d \leq \mid V \mid$. A graph itself can be visualized by a *d-dimensional vector* representing the embedding and a combination of several *d-dimensional vectors*, whereas each vector represents a specific part of the graph, such as vertices, edges, or whole substructures.

Considering the creation process of such *cascades*, reaching *graphs*, they must arise from a *root*, e.g., the *original tweet*, which was the starting point for the *information diffusion*. As already introduced in Section 7.1, cascades are visualized as *graph structures*, where users are represented as nodes and the *retweeted messages* as *edges*, stemming from the original *social graph*, resulting in graphs like those from Fig. 7.1. A homogeneous graph is given, where each node is represented by a distinct social network user and connected by directed edges, whereas edges have no weighting. In particular, in this use case, a user can influence no users, being a leaf in the graph or one or several other users, leading from chains to star-like structures. Of course, a user can also be influenced by more than one user simultaneously, leading to several influence paths in a graph.

Creating graphs using the social graph of a data set, the following example describes the process, which can be seen in Table 7.1 and Fig. 7.2 representing the visualized

graph. Each entry is described by a *Cascade ID*, representing all (re)tweets belonging to a cascade, whereas each Tweet has a distinct *TweetID*, which belongs to a distinct user named the *influencer*. There is always one user whose ID is only present in the Influencer column, representing the root user, i.e., the creator of the original tweet. For each entry, there is another user ID describing the *influenced* user, resulting in a connection from the *Influencer* node to the *influenced* user node and a message ID representing the edge of the connection in the graph. Moreover, each node also has a *timestamp* to comprehend the creation process of the graph.

Table 7.1: Overview on a cascade - Twitter.

Cascade ID	Tweet ID	Influenced	Influencer	Timestamp
231398451722731520	231412774088105984	43865921	52440296	1344008079000
231398451722731520	231451209997905920	488463165	52440296	1344017243000
231500021734977536	231412774088105984	475894591	475894591	475894591

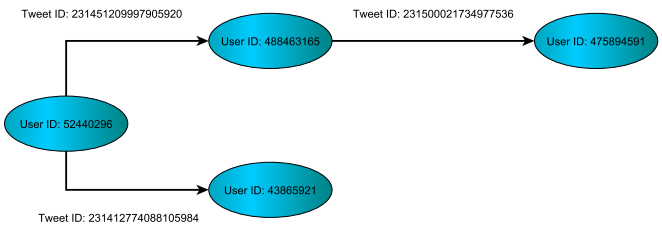


Figure 7.2: Example of a Retweet cascade from Twitter Oly12.

Graph Metrics with Summarization A first attempt to describe graphs as a vector is *Graph metrics with Summarization* from [Kha17; KS16], a technique in graph analysis to describe graphs with well-known graph metrics to build a *feature space* for clustering and classifying them afterward. On the one hand, the features to consider are *metrics* from the *original graph*, which got *pruned* to handle *unconnected nodes* successfully. On the other hand, the *dynamic collapser method* is used to create a *graph summarization*, where aggregated node metrics and the same graph metrics as the original graph were also determined to weigh the graph's structure. Using only graph metrics was not suitable enough to describe a whole graph, so *dynamic collapsing* was performed on the graphs to reduce the graphs only to comprehensible characteristics. Uninfluential nodes, such as *leaves* were, *collapsed recursively*, whereas the weights of the remaining nodes increased when merging leaf nodes into them until

only significant (sub)structures of the whole graph, such as *bridges* or *stars*, remained, describing the *core structure* of graphs. In Fig. 7.3 on the left-hand side, a pruned graph can be seen, whereas the figure on the right-hand side shows a collapsed graph.

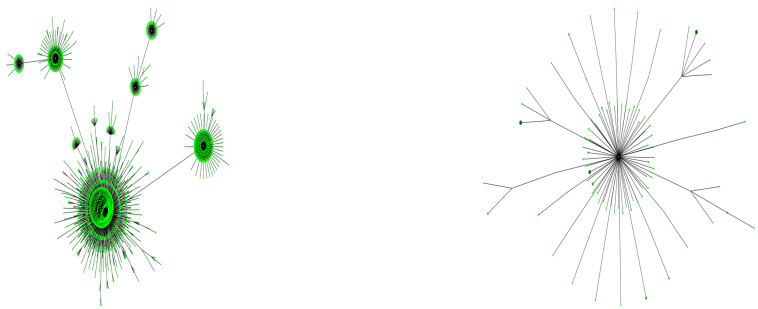


Figure 7.3: Graph Collapsing from [KS16].

Also, for this *collapsed graph*, *graph metrics* were evaluated and combined with those of the *original graph* to get a meaningful graph specification. All of the considered metrics can be seen in Table 7.2. Basic graph metrics, e.g., *number of nodes* and *edges*, *degrees* and *root connectivity*, *path-length* dependent metrics, e.g., *minimum path length* and *diameter* (maximum path length), aggregated metrics such as *arithmetic average*, *median*, *variance*, *slope*, and the *average number of aggregated nodes* were investigated. Moreover, same-level ratio metrics, which are calculated on the same graph level, i.e., *collapsed level* or *original graph level*, such as the *density* describing the ratio between nodes and edges, but also *two-level rations* describing metrics between the original and the collapsed graph, such as *node* and *edge ratio*, but also *outdegree ratio* and *diameter ratio* were considered. In addition, *relational metrics* combining previously mentioned metrics from the original and collapsed graph, such as *sum*, *difference*, *multiplication* of *nodes*, *edges*, *outdegree* and *diameter* were evaluated.

This step in Graph Summarization is a traditional *Feature Engineering* step, as features were chosen by their *correlation* in an iterative KD similar approach. After *Feature Engineering*, *clustering* is performed, while afterward, the results are evaluated to prove the suitability of the chosen features and whether features need to be reduced, achieving better clustering results evaluated with internal clustering metrics and the Elbow technique. While the graph metrics are relatively easy to interpret across each step of the approach, the metrics are partially expensive in terms of calculation. Even though the data sets and the cascades are considering only a few weeks, the *Graph Summarization* has a clear drawback in the *metrics calculation*.

Table 7.2: Overview on graph metrics.

Metric	Description
Number of Nodes	Number of the Nodes (count) from a Graph
Number of Edges	Number of edges (count) from a graph
Degree	Avg. overall node in- and outdegree, describing node position and structure.
Root Connectivity	Amount of the root's outdegree to all edges in the graph.
Minimum Path Length	Shortest path in the graph.
Diameter	Maximum shortest path in the graphs.
Arithmetic Average	Avg. over the number of nodes, edges, outdegree, and path length.
Median	Median of average path length.
Variance	Variance of the path length.
Slope	Linear regression of path length.
AggNodes	Average number of aggregated nodes in the graph.
DensityO	Ratio between the original graph's number of nodes to edges.
DensityD	Ratio between the dynamic graph's number of nodes to edges.
NodeRatio	Ratio between the dynamic graph's number of nodes and the original's
EdgeRatio	Ratio between the dynamic graph's number of edges and the original's
OutRatio	Ratio between the dynamic graph's average outdegree and the original's
PathRatio	Ratio between the dynamic graph's max. path length and the original's
NodeSum	Sum of nodes for the original and collapsed graph.
NodeDiff	Difference of nodes between original and collapsed graph.
NodeMulti	Multiplication of nodes for the original and collapsed graph.
EdgeSum	Sum of edges for the original and collapsed graph.
EdgeDiff	Difference of edges between original and collapsed graph.
EdgeMulti	Multiplication of edges for the original and collapsed graph.
OutSum	Sum of average outdegree for the original and collapsed graph.
OutDiff	Difference of average outdegree for the original and collapsed graph.
OutMulti	Multiplication of average outdegree for the original and collapsed graph.
PathSum	Sum of diameter for the original and collapsed graph.
PathDiff	Difference of diameter for the original and collapsed graph.
PathMulti	Multiplication of diameter for the original and collapsed graph.

7.2.2 Embedding Techniques

In [CZC17], several kinds of *embedding techniques* are presented, which can be seen in Fig. 7.4. The basic embedding techniques, as well as two specialized approaches, will be presented in this section. A graph can be embedded in several ways, such as embedding just *nodes* of a graph, the *edges* of a graph, and an embedding of *complete substructures*, including nodes and edges. Moreover, whole substructures can be embedded into a *Whole-Graph-Embedding*. All these kinds of embeddings result in a low-dimensional space delivering a vector representing nodes and edges of the original graph. After introducing the general idea of embeddings, several strategies leading to mapping graphs into a vector are introduced in the following paragraphs.

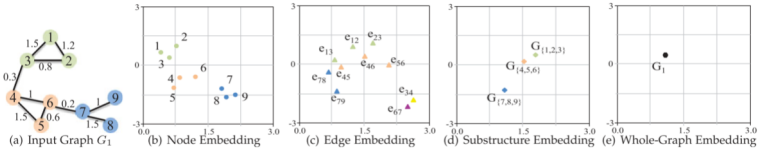


Figure 7.4: Embedding techniques for graphs from [CZC17].

Graph2Vec In contrast to the *Graph Summarization* technique, *Graph2Vec* is an approach originally stemming from language models from the area of *Natural Language Processing (NLP)* and *Deep Learning* based on *Deep Graph Kernels* from Yanardag and Vishwanathan [YV15]. Starting with cascades represented as a graph, they will be partitioned into several *subgraphs*. At the same time, a similarity function is defined. All existing combinations of substructures of a graph will be *enumerated* and compared to those of other graphs using the *similarity function*. Thus, the similarity of two graphs is defined by the *number of similar subgraphs*. An alternative is the *length of shortest paths* and the *neighborhood* set in the graph, generating a *co-occurrence matrix*, representing substructures occurring in both graphs.

In the further steps of this approach, *Deep Learning* algorithms such as *Continuous Bag-of-Words* and *Skip-Gram* are exploited to generate and train a learning model aiming to generate graph embeddings. The general approach can be seen in Fig. 7.5, where for each current word of the *Continuous Bag of Words (CBOW)* model, a prediction is stated by the surrounding words within a given window, while the *Skip-gram model* maximizes *co-occurrence probabilities* for all words within a given window for a current word.

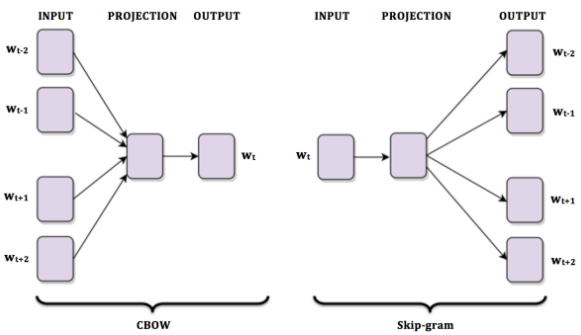


Figure 7.5: Deep Graph Kernels approach from [YV15].

In addition to *Deep Graph Kernels*, Narayanan et al. [Nar+17] modified the approach by refining the term of a *neighborhood* by adding *neighbors of neighbors* until a specific degree representing *substructures* as a *vocabulary* for the *Skip-gram technique* (cp. Fig. 7.6). This extension aims to enhance the probability that embeddings of two graphs in a low-dimensional space exhibit greater proximity when sharing similar substructures.

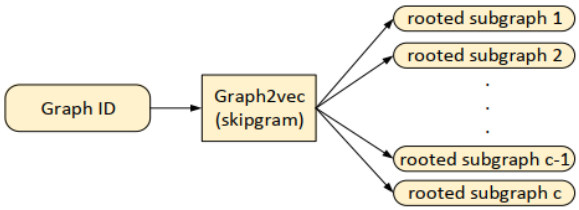


Figure 7.6: Graph2Vec approach from [Nar+17].

A common drawback of this approach is the aspect of *transferability* to *unknown graphs*, as learned embeddings cannot easily mapped on entirely new graphs not present in the training data set. In contrast to *Graph Summarization*, no complex graph metrics need to be calculated. Only for corrective purposes in an early stage of application, graph metrics are valuable to validate the outcome after the clustering as reasonable features in the graph2Vec approach are not apparent. Depending on the embeddings' number of requested dimensions, the runtime for the embeddings calculation can also grow. Thus, the feature space for the dimensions was limited between 4 and 128.

UGraphEmb After introducing the benefits and drawbacks of the *Graph2Vec* approach in the previous paragraph, *UGraphEmb*, another worthwhile Whole-Graph-Embedding technique, will be presented in this paragraph, which was introduced first in [Bai+19]. The strategy of the approach can be seen in Fig. 7.7.

First, for each *node* of a graph, *node embeddings* are generated, fulfilling aspects of *inductivity* and *permutation invariance* describing that embeddings can be adapted on *entirely new graphs*, not present in the *training data* and that permutations of nodes in a graph always deliver the same embedding. *Graph embeddings* are generated after processing the node embeddings, gathering structural differences for various scaling factors enabled for *similarity metrics* such as the *Graph Edit Distance (GED)*. *Neighbor Aggregation* is also a significant aspect in *UGraphEmb*, allowing information flow to neighbors, and for further aggregating layers also an inductive information flow to neighbors of neighbors but also running into danger of information loss when aggregating too often. *UGraphEmb* solves this problem with the so-called *Multi-Scale Node Attention Mechanism (MSNA)*, as for each aggregation layer, the information

of all existing layers is considered and not only the result of the last layer. Also, the training of the Embedding functions has several iterations, where the squared difference between *Graph Edit Distance* and the distance between the Embeddings among themselves is minimized.

Before *UGraphEmb* can start the learning process after step (c) in Fig. 7.7, the information of the similarity metrics of the graphs from the test data needs to be calculated using the *Graph Edit Distance*. *The Graph Edit distance (GED)* is defined as the *minimum needed costs* to transfer a *graph* into another by *inserting*, *deleting*, and *replacing nodes* and *edges*. The GED is a bottleneck as it is very costly compared to *UGraphEmb*, but a worthwhile strategy, e.g., the *diameter*, which is the longest possible shortest path between 2 nodes in a graph, delivered qualitatively inferior results. With the aid of this external data, *UGraphEmb* creates a *matrix*, which is essential for further steps of *UGraphEmb*.

For the generation of the embeddings, the last three *layers* of the *neural network* are essential for generating the *embeddings*. While in the approach of [Bai+19], a 256-dimensional vector was evaluated, for the purposes of this approach, further embeddings with lower *dimensions* such as 16, 32, 64, and 128 were also evaluated. As the leaps in reducing the dimensions in the layers were relatively high for smaller dimensions and thus had a higher influence on previously generated layers, some *in-between layers* were added, having a broad number of possible *nets* for each generated embedding in each dimension.

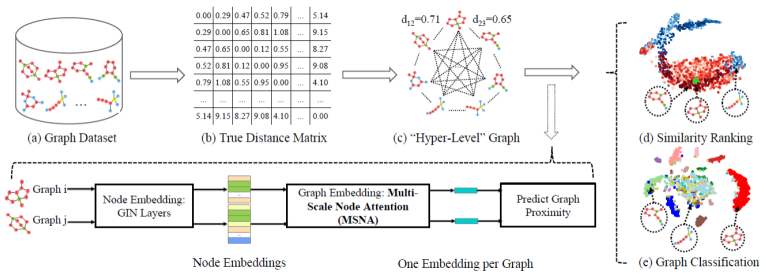


Figure 7.7: UGraphEmb approach from [Bai+19].

7.3 Data Sets & Preparation

After introducing further background on graphs and embedding techniques, the data sets and further preparation will be discussed in this section. For the application of the pipeline in this chapter, a selected *subset* of *high-quality retweet cascades* of the

Olympics 2012, *2014*, and *2016* were created, which can be seen in Table 7.3 as only information on the *social graph* was available for these three data sets. These are excerpts of the same data sets as in Section 5.3.

Table 7.3: Overview on data sets - Twitter cascades.

Data Set	Cascades	Time Period	Category
Olympic Games 2012	4703	Aug. 2012	sports event
Olympic Games 2014	2603	Feb. 2014	sports event
Olympic Games 2016	11170	Jul./Aug. 2016	sports event

In the creation process of *cascades*, only those with at least *five messages*, while in terms of occurring *gaps* in the social graph, only the *connection containing the root node* are considered for further analysis. Having created the *retweet cascades* as *graphs* from the social graph as described in Section 7.2.1, they need to be processed with the three strategies introduced in Section 7.2.2, leading to plain vectors representing each information cascade, which will be presented in the following section.

7.4 Adapting the Methodology

After introducing the data sets and the strategies considering embedding, the main goal is also to find similar structures within the data sets exploiting the proposed approach from Section 1.2, which was introduced more in detail in Chapter 4. Applying the methodology on entirely new data sets, which have, depending on the technique, *completely uninterpretable features*, some steps will distinguish from them of the proposed KD approach, while others are relatively similar. Pointing again to Fig. 4.1 visualizing the flowchart of the proposed approach, the steps and their adjustment to this use case will be briefly introduced before the most significant steps will be presented in more detail in the following sections.

Starting with *retweet-cascades* as raw data sets, introduced in the previous section, only for the *Summarization* method a feature selection as part of the *Feature Engineering* step is needed, as the other embedding techniques provide several dimensions, depending on the specifications, and *UGraphEmb* additionally also for each neural net. An aggregation of the raw data set is performed when creating the *retweet cascades* as a graph, but not after the embeddings are created, as each message of a user who got influenced is part of a cascade. As the embeddings deliver feature values in the space $[-1, 1]$ for each dimension, only minor adjustments in Terms of *Feature Preprocessing* are needed to perform better in the clustering step, as a better cluster assignment can be forced if the bounds are equal. Thus, the feature values must only be set into

equal bounds for all the embedding and summarization techniques, using a *MinMax normalization* reaching a feature space of $[0, 1]$. Furthermore, for relatively small data sets such as those from Table 7.3, a *sampling* is hardly needed and possible in this use case, as the representativity would suffer and lead to tiny clusters being hardly expressive for interpretation and analysis. Thus, clustering and cluster analysis will be performed on the *entire data set*. As that step is somewhat similar to the proposed approach's flowchart, it will be presented in more detail in the following section.

7.4.1 Clustering & Cluster Analysis

After the retweet cascades were built and the embeddings were performed, clustering can cut short the process of analyzing types of cascade structures. While in most of the related work approaches that were investigated in the survey paper of Cai et al. [CZC17], k-means clustering is favored and approved to process and analyze node embeddings, choosing the number of clusters in advance is a well-known challenge in clustering and cluster analysis. Furthermore, the aspects of explainability are infeasible when choosing partition-based clustering techniques. Thus, *Hierarchical Clustering* from Chapter 4's approach, applied successfully on the Twitter and Telegram data sets, this technique was selected for this approach, too. Also, the choice for the most suitable linkage fell in favor of *Ward's linkage*, showing the best results again, as clusters showed uniform structures with well-separated clusters.

During the evaluation of various clustering techniques for *Summarization*, several methods were tested, including k-means, DBSCAN, Mean Shift, BIRCH, and Hierarchical clustering using Ward's, Complete, and Average linkage. Among these, Hierarchical clustering using the Average linkage was found to be the most appropriate approach.

Moreover, a further clustering step is performed for the *UGraphEmb* technique for *outlier detection* and *elimination*. *DBSCAN* is used to reduce the data set by eliminating unusable cascades, as some outliers were noticeable in the data set. in contrast to the data sets from the use cases of Twitter in Section 5.3 and Telegram in Section 6.3, this use case relies only on very small data sets, where outliers can distort the clusters easier, as they have a more significant influence. After eliminating, results for the Hierarchical clustering delivered better results for clusters, as remarkable graphs were no longer visible when inspecting the plots of the graphs.

Initially, the *Olympic Games 2012* data set was used to prove the suitability of the proposed KD approach to cope with information cascades of the three given techniques. Once the data sets were clustered, they must be analyzed to find a suitable number of clusters. The analysis using *internal cluster evaluation metrics* such as *Silhouette*, *Davies-Bouldin*, and *Calinski-Harabasz* and the *Elbow* method is applied to the data sets to enable a fair comparison between the approaches. Moreover, the *effect size-*

based depth-first search was utilized to find more fine-grained shapes, as well as the adaptability on further data sets.

In contrast to *Graph2Vec* and *UGraphEmb*, the *Summarization* technique does not provide varying dimensions. Thus, the internal evaluation metrics vary only for a specific number of clusters. The *Elbow* method, as well as the *Silhouette* score and the *Calinski-Harabasz* score, delivered the best clustering, with *Hierarchical clustering* having a peak at 4 clusters for the *Summarization* technique, showing only a good separation for coarse-grained structures. Compared to analyzing the use cases for user roles in Twitter and Telegram, similar behavior can be observed when using *internal quality metrics* in this use case.

7.4.2 Analysis of Cascade Shapes

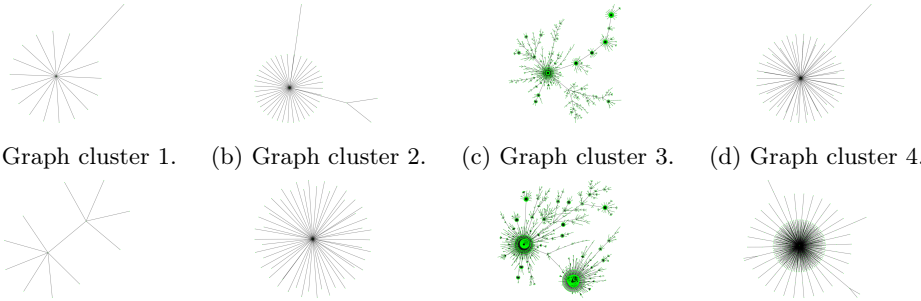
After introducing the methodology in terms of clustering and cluster analysis, a coarse-grained structural analysis is applied to the *Olympics 2012* data set to enable a fair comparison of the three techniques. However, before the general application of the KD approach from Chapter 4 will be evaluated and analyzed, a coarse-grained analysis is performed to identify structures in the following paragraphs. Afterward, the steps from coarse-grained structural analysis to fine-grained results are stated.

Coarse-grained Structural Analysis The first research question deals with terms of *explainability* of the *embedding techniques*. Starting with a coarse-grained structural analysis of *retweet-cascades* using the previously mentioned three embedding techniques leads to a fair comparison between those techniques, as the cluster analysis should be comparable in terms of clusters to analyze. Thus, the best strategy in terms of internal quality metrics is chosen for further fine-grained analysis.

Before analyzing the other embedding techniques compared to the *Graph Summarization*, the cluster results of the *Graph Summarization* will be analyzed by their shape. Thus, some representatives of the clustered graphs were chosen, as shown in Fig. 7.8.

The first of the four clusters delivered a small, sparse set of *star-like structures* with *high root connectivity* and a *diameter* greater or equal to 2. A *star shape* describes a graph where information spreads almost entirely from the *root*. A very similar structure to the first cluster can be found in the second cluster. In contrast to the *star-like* clusters from cluster 1, *denser stars* can be found, as the number of *nodes* and *edges* is much higher. An entirely different structure to the first two clusters can be found in the third cluster, representing structures with way more *nodes* and *edges* as part of *longer chains* with several *children and dense cascades*. *Chains* describe parts of graphs where information is not spread from the *root* but from other *nodes* all over the *graph*. The fourth cluster comprises graphs with many *edges* and *nodes*

with *big and dense structures* and *higher node edge and outdegree ratios* leading to *big and dense cascades* with mainly diameters equal to 1. The representatives of the fourth cluster are *short, dense, and big stars* and *not chains*. The results of the *Summarization* technique can be interpreted easily as the features allow terms of *explainability*, answering the third research question from Section 7.1.



(a) Graph cluster 1. (b) Graph cluster 2. (c) Graph cluster 3. (d) Graph cluster 4.

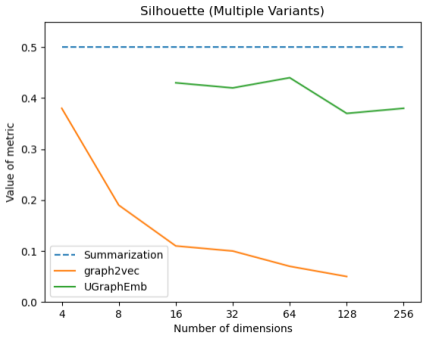
(e) Graph cluster 1. (f) Graph cluster 2. (g) Graph cluster 3. (h) Graph cluster 4.

Figure 7.8: Graphs for clusters 1-4 of Graph Summarization from [Kha17].

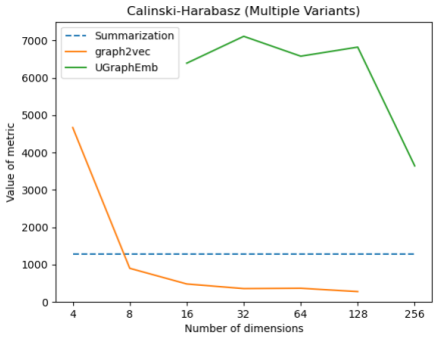
Focusing on the *embedding techniques*, starting with the *Graph2Vec* technique, a higher state-space arises, as the number of dimensions and clusters can vary. First, the *Elbow* method was used to indicate the best number of clusters for clusterings using *Graph2Vec* with 8 and 16 dimensions. A suitable number of clusters for both embedding variants delivered 4 and 8 clusters for the 8-dimension variant, while the 16-dimension variant delivered only a peak at 2 and another close one at 4 clusters. Due to the comparability to the *Graph Summarization* technique, the number of clusters was set to 4. For all possible dimensions in the *Graph-Embedding* the *Silhouette* and *Calinski-Harabasz* scores were evaluated, as can be seen in both subfigures of Fig. 7.9.

The *Silhouette* score of around 0.5 for the *Graph Summarization* technique, delivers a better result than the *graph2vec* technique for all varying dimensions between 4 and 128. In contrast to this approach, the *Calinski-Harabasz* score reaches only better results for 4 dimensions, while for a growing number of dimensions, the score constantly decreases and falls below the *Summarization* techniques' score. This comparison shows that a coarse-grained structural analysis is generally possible for both approaches. A further evaluation comparing the 4 clusters of both approaches delivered similar results. Having a look at the heatmaps in Fig. 7.10, the comparison between *Graph Summarization* and *graph2vec* using the *Jaccard* score shows only medium-high similarities for 2 cluster pairs, as the number of consisting cascades diverges a lot. When creating the same graph metrics from *Graph Summarization*

for graph2vec, the comparison of boxplots showed that clusters were similar but not similar enough to show an equal significance w.r.t. to coarse-grained structures.



(a) Silhouette of embedding techniques.

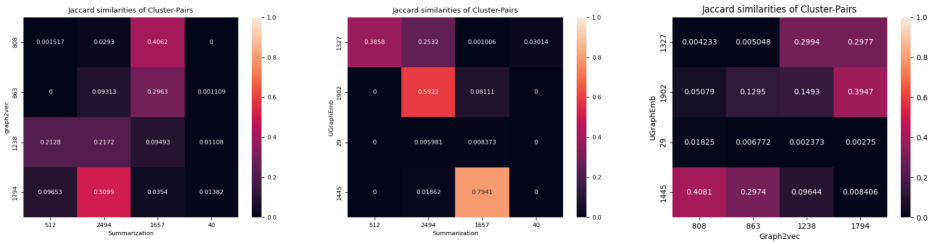


(b) Calinski-Harabasz of embedding techniques.

Figure 7.9: Comparison of several cluster evaluation metrics for all embedding techniques.

The most suitable configuration considering the preferred neural net must be chosen before the *UGraphEmb* technique can be evaluated against the other two approaches. For almost all dimensions, net 4 delivered the best results, especially for 16 and 64 dimensions, leading to only comparing the embeddings using net 4 overall dimensions against the other two techniques. Focusing now on the comparison of *internal clustering quality metrics* for the *UGraphEmb* technique, one can see that the *Silhouette* scores in Fig. 7.9a dominate the *graph2vec* results and are almost as good as the *Graph Summarization* result for all varying dimensions. Also, the results of the *Calinski-Harabasz* show the benefits of the *UGraphEmb* technique as it dominates both approaches for all tested dimensions.

Looking closer at the *heatmaps* in Fig. 7.10, *UGraphEmb* shows relatively high similarities for 2 of 4 clusters compared to *Graph Summarization*. In contrast, the differences between the cluster pairs of *UGraphEmb* and *graph2vec* are moderately high, as no cluster pair has a reasonably high similarity. The comparison of all three approaches led to the choice of the *UGraphEmb* technique as the most suitable for this approach. Even though the features both in *graph2vec* and *UGraphEmb* are hidden, the lack of explainability can be compensated by evaluating several graph metrics from Table 7.2 on the clusters by creating and analyzing boxplots as a kind of reverse feature engineering. Fig. 7.11 shows a boxplot of the whole Olympics 2012 cascades data set with hidden features embedded with *UGraphEmb* in *32 dimensions and net 4*.



(a) graph2vec vs. Summ. (b) UGraphEmb vs. Summ. (c) UGraphEmb vs. graph2vec

Figure 7.10: Heatmaps for cluster comparison of embedding techniques.

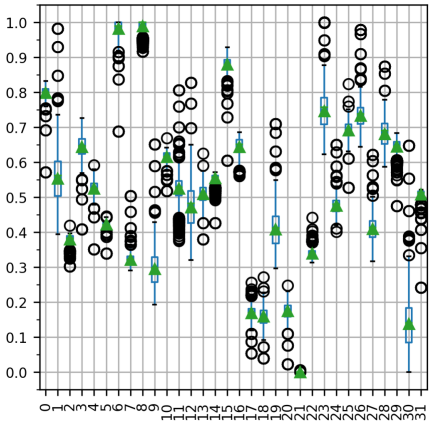
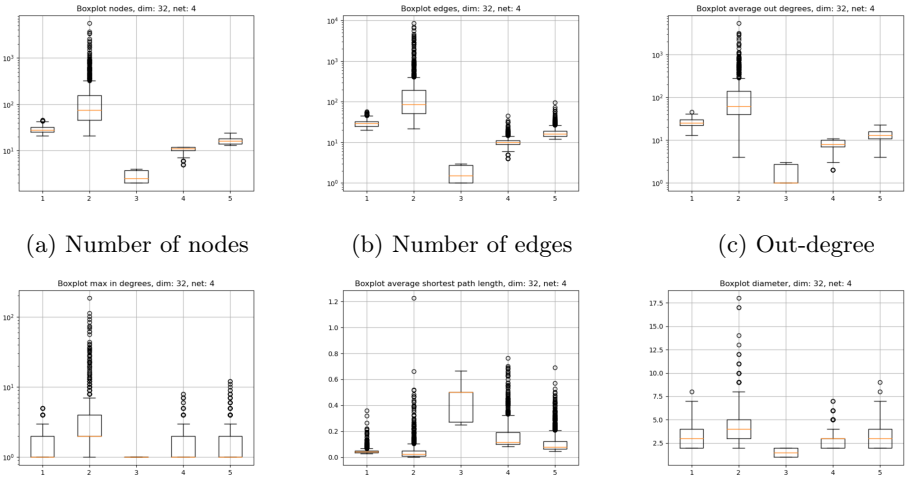


Figure 7.11: Boxplot embedded Oly12 with hidden features - UGraphEmb, 32 dim. net 4.

To *reverse engineer* the clustered data sets' features, several *graph evaluation metrics* from Table 7.2 such as *number of nodes*, *number of edges*, *indegree*, *outdegree*, *minimum shortest path* and *diameter*, were used to describe the clusters. The boxplots in Fig. 7.12 show *graph metrics* for the clusters and reveal significant *deviations*, indicating the suitability of interpreting *hidden features* successfully. This strategy is vital for *manual class labeling*, as the features deliver no suitable information on how clusters are built. With this aid, the third research question can be approved with limitations.



(a) Number of nodes (b) Number of edges (c) Out-degree

(d) Indegree (e) Minimum shortest path (f) Diameter

Figure 7.12: Boxplots with graph metrics for UGraphEmb 32 dim. net4.

Fine-grained Structural Analysis After choosing the *UGraphEmb* technique as the most suitable approach by evaluating internal cluster metrics leading to coarse-grained structures, the existence of fine-grained structures is proved for this use case as well. The results of the *UGraphEmb* technique delivered a similar significance w.r.t. coarse-grained user roles from the *Graph Summarization* technique. Further analysis exploiting the *Effect-size-based depth-first search* showed relatively strong effects when plunging into the *subtrees* of the *dendrogram*. Setting the *significance criterion* to at least two large effects, one very large or one huge effect the dendrogram in Fig. 7.13 delivers significant subclusters, which are marked from 1-11 in the dendrogram and Table 7.4, whereas the first adjective always describes the graphs size w.r.t. nodes and edges, while the rest of the description characterize the graphs' shape.

One can see 4-5 *coarse-grained groups* when looking at the dendrogram. The first group consists of cluster 1 representing *huge stars*, while the second group (clusters 2-4) mainly consists of variations of *big and large stars*, with some *variations* considering *chains* and *subsidiary centers*. *Subsidiary centers* are a *mixture* of *chains* and *star structures*, where information is spread almost evenly from the *root*, but some chains create new *subcenters* spreading the information again, like in star structures. Clusters 5-7 describe the next coarse-grained group consisting mainly of *tiny* and *small stars* with some *variations* considering *doubled stars*, *subsidiary centers*, and *occasional chains*, while clusters 8 and 9 deal with *medium stars* and *variations* considering

occasional small chains and *subsidiary centers*. The last coarse-grained group (clusters 10 and 11) mainly describes *small stars* and *variations* dealing with occasional *small chains* and *subsidiary centers*.

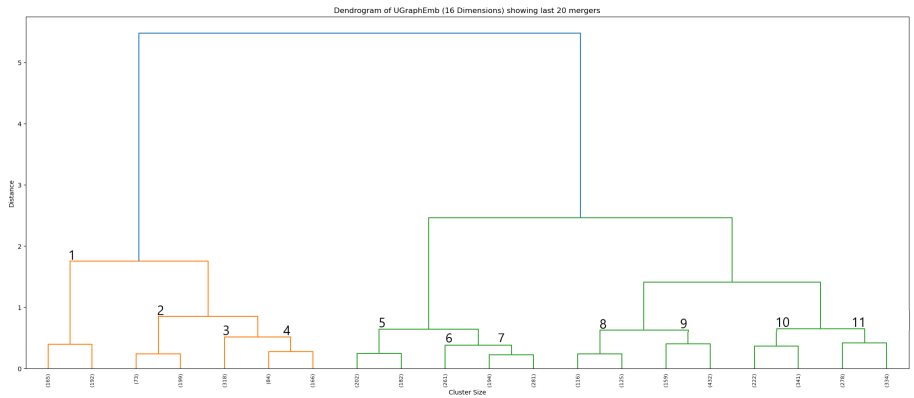


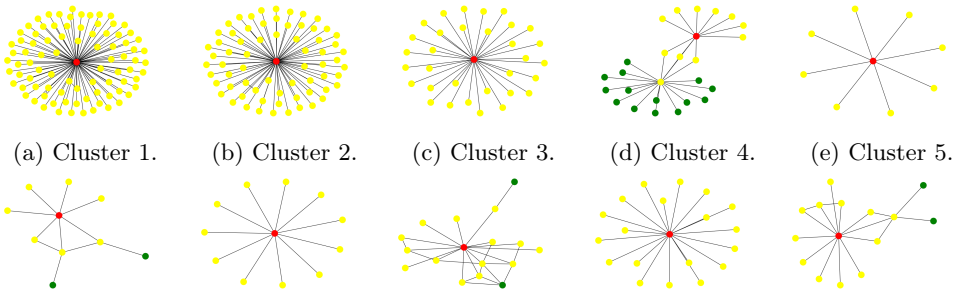
Figure 7.13: Dendrogram of UGraphEmb net 4 with 16 dimensions.

Table 7.4: Graph shapes for clusters of UGraphEmb net 4 with 16 dimensions.

Number	Count	Description
1	357	huge stars
2	272	large stars
3	318	big stars
4	250	big stars, occasional chains, subsidiary centers
5	384	tiny stars
6	261	tiny stars, doubled stars, subsidiary centers, occasional chains
7	475	small stars
8	241	medium stars, occasional small chains, subsidiary centers
9	591	medium stars
10	653	small stars, occasional small chains, subsidiary centers
11	612	small stars

The *hierarchy* is primarily built upon the sizes of the graphs, as the first groups consist of *huge*, the second *big* and *large*, the third *tiny* and *small*, while the fourth and fifth groups deal with *medium* and *small* graph structures. When diving deeper into the dendrogram, each coarse-grained structure is divided into *subgroups*, such as *star-shaped* graphs, graphs with *subcenters*, or many graphs characterized by *chains*.

Finally, the motivation for fine-grained structures in this scenario is given, too, as manifold *graph structures* are built with the approach presented in Chapter 4. The results presented in this paragraph dealing with finding fine-grained structures in the use case of retweet cascades confirm the first research question from Section 7.1.



(a) Cluster 1. (b) Cluster 2. (c) Cluster 3. (d) Cluster 4. (e) Cluster 5.

(f) Cluster 6. (g) Cluster 7/11. (h) Cluster 8. (i) Cluster 9. (j) Cluster 10.

Figure 7.14: Graph structures for clusters 1-11 for UGraphEmb.

Application of the Approach The second research question, dealing with the general application of the approach, will be examined after successfully investigating the aspects of a fine-grained structural analysis. All of the three approaches, *Graph Summarization*, *graph2vec*, and *UGraphEmb*, can be adapted with *minor adjustments* as already stated in Section 7.4. While *Feature Engineering* is only needed in *Graph Summarization*, the other two techniques deliver hidden features for the clustering process. Only the number of dimensions for the embeddings can be chosen, being some *Feature Engineering*, even though the analyst does not influence the features as they remain latent after applying the embedding techniques graph2vec and UGRaphEmb.

Focusing on *clustering* and *cluster analysis*, a contrast between *Graph Summarization* and the two other embedding techniques can be observed regarding the explainability of features and, thus, the interpretability of results, as features can only be tracked successfully in *Graph Summarization*. For the two other approaches, a *workaround* with a subsequent evaluation of *graph metrics* for the clusters is suitable but *rather expensive*. Thus, *metrics* from Table 7.2 were chosen to support the analyst in the manual labeling process of the clusters. Once a data set was evaluated successfully with those graph metrics and the analyst obtained a sense for analyzing latent features, a *transfer* to other data sets is possible.

Also, applying the *classification* of clusters and knowledge transfer is feasible in this use case but not reasonable as the data sets are currently relatively small. Finding suitable and representative cluster means for training data is challenging due to

the small size of the data sets, as representative samples cannot be drawn quickly. Furthermore, the effort in *manual labeling* for creating training data sets iteratively would be costly. A manual analysis by observing the plots of the graphs is sufficient at this point while building a classifier would be valuable when investigating larger data sets or transferring information from one to another data set. Also, applying the novel *Multi-Sampling and Combination Strategy* is possible but makes almost no sense in small data sets, as representativity is no longer guaranteed when choosing the sample size. Furthermore, this strategy makes only sense when utilizing a fully *automatic classification* process and thus remains for future work.

Finally, the results in the previous paragraphs show that a successful *application* of parts of the process is possible even in an *entirely new use case*, exploiting latent features. The most prominent challenges were choosing the features for the *Graph Summarization* technique, the aspects of interpretability and explainability for the *Summarization* techniques due to the lack of original features, and the step from coarse to fine-grained graph shapes. In comparison with the use cases of analyzing user roles from Twitter (Chapter 5) and Telegram (Chapter 6), the *human effort* is higher, as there are several strategies for mapping graphs to a vector with a lot of *preprocessing* and *tuneables*. Moreover, performing and evaluating several *clustering techniques* and the cluster analysis with internal graph metrics and the novel *Depth-First Search Approach*, requiring some *tuneables* for *cluster analysis*, were also time-consuming. The application of this approach until the *manual labeling step* and the results showed that the second research question dealing with applying the KD approach, stated in Section 7.1, can easily be confirmed, while the steps of classification and the *Multi-Sampling and Combination Strategy* remain future work.

Transferability to new Data Sets After reasoning the adaptability of the general approach to an entirely new scenario considering information cascades on the *Olympics 2012* data set, the aspects of *transferability* within data sets stemming from the *same source* and thus having a similar feature space was one of the main contributions when proving in the user roles scenario of the Twitter data sets in Chapter 5. In contrast to the Twitter data sets for discovering user roles, the features in this scenario are entirely latent. Thus, it is more challenging to understand the cluster's *results*, especially as they are dependent on the *embedding*. When using *UGraphEmb* as the most suitable embedding result, choosing the nets appropriately is essential. Finally, it is possible to transfer the approach to each of the other two remaining data sets, also receiving valuable results after the clustering and the cluster analysis. With the insights from these research questions, several aspects for future work can be pursued.

7.5 Related Work

To define from clustering information cascades regarding graphs being a pivotal point in this chapter, much related work has been published in the last 20 years. Work from van Dongen [Don00] dealt with graph clustering and cluster analysis in general, while Schaeffer [Sch07] specified graph clustering by focusing on vertices. Moreover, McGlohon et al. [McG+07] focused on clustering blog entries in a dynamical use case as they concentrated on evolution over time, while the first steps considering the Social Media service Twitter in terms of graph clustering were made by Kafeza et al. [Kaf+14] as they worked on information diffusion of tweets by concentrating on linguistic features. The direction of research went on to analyze structural patterns in information cascades with the work of Zang et al. [Zan+17] concentrating on a small set of metrics describing information cascades and their correlation to each other. All of the mentioned work paved the way for graph analysis with the aid of clustering to find patterns of cascades exploiting graph metrics as features.

Moreover, not only is graph analysis vital for the approach presented in this chapter but also embedding and summarization techniques for graphs are essential for the success of this use case as they need to be processed before clustering is possible. Tian et al. [THP08] provided a summarization technique dealing with grouping nodes based on user attributes and relationships between users, while the work of Koutra et al. [Kou+15] focused more on the explainability of edges leading to predefined overlapping subgraphs with graph structures such as stars or chains. In contrast to those explainable features, Narayanan et al. [Nar+17] focused on exploiting unsupervised learning techniques, presenting their graph2vec approach to transform graphs into embeddings, where no information on manually labeled graph structures is needed. As the results of the embeddings are dependent on the training data they were created on, Bai et al. [Bai+19] focused more on an inductive approach, UGraphEmb, which is an advancement on graph2vec, which also works on unfamiliar data sets.

7.6 Conclusion

The results presented in this chapter showed that the *application* of the *approach* from Chapter 4 is possible to an *entirely new use case* dealing with *retweet cascades* instead of user roles, which were the central components of the use cases in Chapter 5 for the Twitter data sets and Chapter 6 for the Telegram data set. The possible application of parts of the KD pipeline, in particular, a relatively different kind of preprocessing and especially the hierarchical clustering with *Ward's linkage* delivered similar to the other use cases coarse-grained structures when analyzing clusters with internal cluster metrics. The further application of the *effect sized based depth-first*

search from Definition 33 paved the way for fine-grained structural analysis, finding different kinds of *cluster shapes* as *coarse-grained shapes* could be elaborated. Also, the *transferability* to new data sets within this use case showed the suitability of the application as clusters with similar *fine-grained shapes* could be found. Finally, all of the stated research questions from Section 7.1 were confirmed by applying the proposed approach to this *entirely new use case*.

For future work, it is interesting to apply the approach to more data sets with bigger sizes and also apply the novel *Multi-Sampling and Combination Strategy* to this use case. Furthermore, when dealing with more and more extensive data sets, providing an *automatic classification* of clusters is essential. Thus, the Active Learning (AL) approach must also be applied for creating *training data* to classify data sets. The first steps of *manually labeling classes* were already performed in this section, being a good starting point when dealing with *new data sets*. As the cascade data sets from this use case were created with the *social graph* from the Twitter data set evaluated in Chapter 5, it is also of particular interest if there is a *correlation* between *user roles* and *cascade shapes*. In particular, distinct user roles may occupy specific positions in retweet cascades w.r.t. creating and forwarding content and thus influencing other users in several ways in the social media service.

Part III

Conclusion

Chapter 8

Conclusion

Wer kann schon sagen was mit uns geschieht,
Vielleicht stimmt es ja doch?
Dass das Leben eine Prüfung ist,
In der wir uns bewähren sollen

Die Toten Hosen - *Paradies*

In this thesis, a novel Knowledge Discovery (KD)-based approach was established to capture and analyze fine-grained structures in large-scale user-generated data sets. Various major contributions, introduced in Chapter 1 and conceptually refined in Chapter 4, were addressed, namely, the reduction of human effort and the explainability within each step of the approach. The KD pipeline was extended with several steps, such as a novel *Multi-Sampling and Combination Strategy* and explainable labeling to ensure a knowledge transfer to other data sets in various use cases and applied and evaluated on three complementary use cases, Twitter in Chapter 5, Telegram in Chapter 6 and Cascade-Shapes in Chapter 7. Moreover, in Chapter 3, a somewhat orthogonal approach exploiting the *Borda Social-Choice voting* rule to reduce human effort by avoiding normalization was introduced. The following sections summarize the goals, the contributions, and their results in detail.

8.1 Structure Discovery of Fine-Grained User Roles in Social Media

The contributions of the approach presented in Chapter 4 address the detection and analysis of fine-grained user roles in large-scale user-generated data sets, whereas several well-known Machine Learning (ML) techniques, such as clustering and classification, are encapsulated in a novel *Multi-Sampling and Combination Strategy*. The main contributions introduced in Section 1.3 address the provision of a KD approach for detecting and analyzing fine-grained structures in an explainable and comprehensive way, i.e., user roles, which is also applicable to other data sets stemming from topical and temporal related and non-related data sets from the same and diverging sources.

8.1.1 Analyzing Fine-Grained Users in Twitter

The first use case considered in this thesis applied and evaluated in Chapter 5, dealt with data sets from the Social Media service Twitter. The conceptual application of the approach was successful on the initial data set - the *Olympics 2012* data set - as it paved the way from coarse-grained user roles to explainable and discernable fine-grained ones with the aid of comprehensive experiments and analyses answering the first contribution dealing with a complete application of the pipeline to the initial Twitter data set.

Addressing the second research question, the application and analysis of the novel approach to a variety of several other topical related and non-related data sets, such as several other sports events like the *Olympics Games* over a period of 10 years and a short-term event over several years like the *Super Bowl*, but also topically diverse data sets focusing on tragic incidences was confirmed. The knowledge of fine-grained structural user roles gained from the initial data set was transferred successfully to a variety of new data sets, whereas the human effort in adjusting steps of the approach was pared down to a minimum. Most of the same user roles appeared again in all the data sets, with varying quotas paving the way for long-term analyses, as both feature shifts and drifts were noticeable for the investigated user roles.

Further sub-contributions dealt with stability and coverage of fine-grained user roles, providing insights on tuneables in the novel *Multi-Sampling and Combination Strategy*, such as the influence of sample sizes, in due consideration of representativity. Moreover, several further experiments and analyses considering the certainty and stability of user roles, and also the correlation of best and second-best user roles, due to the probabilistic nature of the approach, proved and substantiated the eligibility of the novel sampling and combination strategy-based KD approach for finding fine-grained user roles. Addressing the novel strategy, in particular, several tuneables considering

sampling were investigated for finding the best sampling strategy fitting to varying data sets in terms of sizes and, thus, the number of samples and sample sizes.

The long-term analysis of two time series, the *Olympic Games* and the *Super Bowl*, answered several questions dealing with the impact of dis- and reappearing user roles and the change of their general and fine-grained user-role-based behavior, leading to the intention of building whole models for long-term user role change analyses. The suitability of a dynamic Markov-based threshold model was successfully evaluated regarding the precision of user roles to a variety of static models. Moreover, applying the whole conceptual model to the second time series showed valuable results. In addition, further experiments proved that the transfer of knowledge for simulating known data sets and predicting new data sets works for topically related data sets. These aspects substantiate again the main contribution of saving human resources, as for new data sets, the model building can cut short the whole process by sparing almost the entire KD pipeline.

8.1.2 Analyzing Fine-Grained Users in Telegram

After transferring the application of the proposed approach to several other data sets stemming from the same source for the first use case, the conceptual transfer of the whole KD pipeline to data sets stemming from another source, the instant messaging service Telegram represents the second use case from Chapter 6. The main contribution also addresses the general application for finding fine-grained user roles in an explainable and specific way, as well as some additional sub-contributions.

The general application of the approach presented in Chapter 4 was successfully adapted to Telegram, as some known fine-grained user roles from the Twitter use case and further new ones were detected and analyzed in a comprehensible and explainable way. Only the steps of preprocessing and classification from the KD approach needed to be adjusted for this use case, showing that human effort can be economized, and thus, time can be saved even when transferring the whole approach from one detail-elaborated use case to another similar use case. Especially the Active Learning (AL)-driven approach for building classifiers delivered an added value compared to the Twitter use case, as much time was saved due to a minimized human intervention. Also first experiments considering the novel *Multi-Sampling and Combination Strategy* showed valuable outcomes, as the stability and certainty of user roles were substantiated. In addition, a discussion and analysis of similarities and differences between the two use cases and the occurring user roles delivered party different user behaviors between Twitter and Telegram. Some limitations were noticeable in the building and evaluating the classification step and the following Sampling and Combination strategy due to the sparse number and small size of data sets.

8.2 Analyzing Cascade Shapes from Twitter Data Sets

Last, one entirely different use case to the previous two use cases was addressed in Chapter 7, answering again the research question of transferring and applying the pipeline to a different data set for finding fine-grained structures. For this challenging attempt, Feature Engineering was cut short, as Feature Selection and Preprocessing were not necessary due to the appropriate output of the embedding techniques, whereas tuneables in the embedding strategies needed to be adjusted. Only the clustering, cluster analysis, and manual class labeling steps were considered. At the same time, the *Multi-Sampling and Combination Strategy* was not yet necessary due to the small size of the data set. Also, the AL-driven approach of building training for classification, and thus the classification step, was not yet considered due to the small data set.

Analyzing graph shapes in the context of information cascades from the social media service Twitter shows the KD approach's versatility from Chapter 4, as not only the general application of parts of the KD approach was successfully transferred, but also fine-grained structures were detected. Due to the nature of embedding techniques, the challenge of analyzing the opaque features of the clusters comprehensively and explainably was accomplished successfully with a workaround by applying well-known graph metrics on the clusters. Further experiments on data sets demonstrated the application of other sports data sets, substantiating the reduction of human effort in this entirely new use case. Furthermore, the time effort in reducing human intervention by applying parts of the KD pipeline to this entirely new use case was confirmed.

8.3 Borda Social Choice Voting Rule

Both approaches addressed in Chapter 3 dealt with reducing the result sets in Pareto-optimal use cases such as advertising and recommendations by exploiting user preferences. To summarize similar results, these Pareto-optimal data sets were clustered with two extended partitional clustering approaches presented in Chapter 3, exploiting both the *Pareto-dominance* criteria and the *Borda Social Choice Voting Rule* for a k-means-based cluster allocation to prevent the need for the eminent preprocessing step of Normalization and Standardization. While the first approach is more suitable for a lower number of dimensions, the latter also works in higher dimensional spaces, as the comprehensive experiments showed the competitiveness to the basic k-means clustering w.r.t. runtime and number of iterations.

Considering the main contributions introduced in Section 1.3, the application of the clustering approach, and thus a complete KD pipeline to manifold sets of data

sets is possible, such as advertising and recommendations in a movie-based use case, online-dating and much more. Eliminating normalization and standardization in Preprocessing significantly improves runtime and reduces the need for human intervention, resulting in greater efficiency and cost savings. Limitations for both approaches are the application of massive data sets due to the weak points of partitional clustering. Even though clustering exploiting the *Borda Social Choice Voting Rule* would work in data sets with a high-dimensional feature space, the data sets in the use cases of Chapter 5 and 6 are too large. Moreover, explainability for the cluster composition of k-means clustering is impossible, complicating the cluster evaluation. In summary, both approaches are worthwhile clustering strategies, which can be adapted currently to data sets of small to medium size.

8.4 Summary

In this thesis, many use cases were considered to propose applying an extended KD pipeline to comprehensively and explainably analyze fine-grained structures in user-generated data sets. A plethora of comprehensively realized experiments and analyses proved the suitability of the approach as a valuable alternative to a traditional KD approach. The novel *Multi-Sampling and Combination Strategy* is an added value as it is scalable for many manifold data sets from medium to massive sizes. Moreover, the conceptual transferability of the methodology to a wide variety of use cases provided benefits in terms of time effort, as human intervention could be reduced to a minimum.

Chapter 9

Future Work

> Es ist nicht Deine Schuld,
> dass die Welt ist, wie sie ist
> Es wär nur Deine Schuld,
> wenn sie so bleibt
>
> — Die Ärzte - *Deine Schuld*

A novel Knowledge Discovery (KD)-based approach, the *Multi-Sampling and Combination Strategy*, was developed and presented in this thesis to analyze large-scale user-generated data sets in terms of explainable fine-grained structures. The method was successfully applied in a plethora of various scenarios and thoroughly discussed in Chapter 8. Opportunities for further exploration remain, particularly regarding the KD pipeline presented in Chapter 4 and the case studies in Chapters 5, 6, and 7. Additionally, the *Borda Social Choice Voting Rule* introduced in Chapter 3 offers captivating directions for further investigation.

9.1 Structure Discovery of Fine-Grained User Roles in Social Media

The novel KD pipeline established a broad range of possibilities for future work. Even though many steps of the approach could be optimized yet, some need further attention to reduce runtime and human intervention. While the drawback of the complexity of hierarchical clustering was already addressed with the novel *Multi-Sampling and Combination Strategy*, a parallelization of clustering smaller but more representative

samples would be suitable as time could be saved additionally. Besides that, a fully automatic approach would be beneficial, determining the sampling strategy, sample sizes, and number of needed samples depending on the data set's specifications and the user's desires. For each use case, a tailored strategy could be applied to reduce runtime and human interventions, as the novel *Multi-Sampling and Combination Strategy* deals with many tuneables to adjust. Another conceptual optimization is dealing with building training data for classifiers. As the whole process was initially almost manually driven and only further automatized for the Telegram use case, the Active Learning (AL) approach needs to be optimized to discharge human experts.

Furthermore, the application of a wider variety of data sets, such as events over a whole year or season with several rounds, such as Formula 1 or Football Bundesliga, would be beneficial for analyzing user roles in an entirely different use case, as feature drifts and shifts in short intervals may reveal different user behavior. Considering relatively large data sets such as the FIFA and UEFA football events, slicing the data sets into group stage and knockout stage can also reveal different user behavior within shorter periods of time. Thus, comparing user roles' long-term and short-term evolution and their correlation remains a compelling research area.

Exploring additional data sets would help validate the existence of specific user roles. For instance, analyzing data sets from Telegram that involve political extremists, conspiracy theorists, climate change deniers, support chats, communication between fans, and other chat-like discussions would be beneficial. These directions could provide further evidence for fine-grained user roles.

Considering the model-building process, more refinements and adjustments would be beneficial, as the model needs more data for more precise simulations and predictions of role changes over time. With these refinements, the models' transfer to other data sets can also be improved. As the combination of training data for the classifiers delivered suitable results in Chapter 5, combining role chains from several data sets may be a worthwhile approach for refinement. In addition, the model-building process would also be a valuable approach for short-term round-based events such as Formula 1 or Bundesliga to prove the dynamic threshold models' suitability in further use cases.

9.2 Structure Discovery of Fine-Grained User Roles in Graphs

Graph structures arising from graph embeddings were analyzed successfully in terms of finding explainable fine-grained structures in Chapter 7 by applying parts of the KD pipeline introduced in Chapter 4. As this use case, representing an independent research area has more open issues to pursue, the most inspiring ones with overlaps to

the Twitter user role-based use case from Chapter 5 will be outlined. As information cascades are composed of several users' messages, their influence and interactions among themselves, the roles of users within an information cascade are of particular interest, as the root user may hold an entirely different position as leaf users. The latter, being influenced last do not influence others, while bridge users, spread information into another cliques. Thus, the user roles' correlation from the Twitter data sets found and evaluated in Chapter 5 are of specific interest in analyzing cascade shapes.

9.3 Borda Social Choice Voting Rule

Both approaches presented in Chapter 3 rely only on the clustering and cluster analysis part of the KD pipeline, providing novel k-means clustering-based approaches addressing user-preference-based use cases on Pareto-frontiers. As the exploitation of the *Borda Social Choice Voting Rule* was successfully applied to partitional clustering, the use cases are typically narrowed down to smaller and medium-sized data sets due to the drawbacks lying in the nature of partitional clustering. For larger data sets such as those from the use cases in Chapter 5 and 6 of the KD approach from Chapter 4, partitional clustering stretches to its limits. Thus, a worthwhile approach would be exploiting the *Borda Social Choice Voting Rule* as allocation to other clustering approaches, such as Hierarchical clustering, to benefit from explainability issues. The benefit of saving additional time and discharging experts by avoiding Feature Preprocessing steps such as normalization and standardization would also improve the whole KD pipeline from Chapter 4.

Moreover, implementing a weighting feature, which is very important for users in recommender use cases, would be very beneficial as the clustering result would be tailored individually to users. Further directions are also to map categorical data like sets, e.g., a set of favored genres in a movie recommender using other metrics like TF-IDF, Okapi BM25, or vector space model from the area of information retrieval, instead of the well-known but trivial Jaccard coefficient.

The novel approaches were applied in several use cases, such as a movie recommender. Many use cases are of interest to demonstrate the suitability of this approach. These include micro targeting-based advertisements and music streaming services, books, and video game recommendations.

Bibliography

[Ach+13] Anita Acharya, Anupam Prakash, Pikee Saxena, and Aruna Nigam. "Sampling: Why and How of it?" In: *Indian Journal of Medical Specilaities* (Jan. 2013) (cit. on p. 91).

[AK15] Mariette Awad and Rahul Khanna. "Machine Learning and Knowledge Discovery". In: *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Berkeley, CA: Apress, 2015, pp. 19–38 (cit. on p. 12).

[Alj+18] Elie Aljalbout, Vladimir Golkov, Yawar Siddiqui, Maximilian Strobel, and Daniel Cremers. *Clustering with Deep Learning: Taxonomy and New Methods*. 2018 (cit. on p. 30).

[AMS19] Alessia Antelmi, Delfina Malandrino, and Vittorio Scarano. "Characterizing the Behavioral Evolution of Twitter Users and The Truth Behind the 90-9-1 Rule". In: *Companion Proceedings of The 2019 World Wide Web Conference*. WWW '19. San Francisco, USA: Association for Computing Machinery, 2019, 1035–1038 (cit. on p. 175).

[Ank+99] Mihael Ankerst, Markus Breunig, Peer Kröger, and Joerg Sander. "OPTICS: Ordering Points to Identify the Clustering Structure". In: vol. 28. June 1999, pp. 49–60 (cit. on p. 36).

[AV07] D. Arthur and S. Vassilvitskii. "K-means++: The Advantages of Careful Seeding". In: *ACM-SIAM '07*. SODA '07. New Orleans, Louisiana, 2007, pp. 1027–1035 (cit. on p. 35).

[Bai+19] Yunsheng Bai, Hao Ding, Yang Qiao, Agustin Marinovic, Ken Gu, Ting Chen, Yizhou Sun, and Wei Wang. "Unsupervised Inductive Whole-Graph Embedding by Preserving Graph Proximity". In: *CoRR* abs/1904.01098 (2019). arXiv: `1904.01098` (cit. on pp. 217, 218, 229).

[Bar14] Oana Barbu. "Advertising, Microtargeting and Social Media". In: *Procedia - Social and Behavioral Sciences* 163 (2014). International Conference on Communication and Education in Knowledge Society, pp. 44–49 (cit. on p. 4).

[Bau+20] Jason Baumgartner, Savvas Zannettou, Megan Squire, and Jeremy Blackburn. "The Pushshift Telegram Dataset". In: *Proceedings of the International AAAI Conference on Web and Social Media* 14 (May 2020), pp. 840–847 (cit. on p. 206).

[BD+14] Mariano Beguerisse-Díaz, Guillermo Garduño-Hernández, Borislav Vangelov, Sophia Yaliraki, and Mauricio Barahona. "Interest Communities and Flow Roles in Directed Networks: The Twitter Network of the UK Riots". In: *Journal of the Royal Society, Interface / the Royal Society* 11 (Dec. 2014) (cit. on p. 18).

[Ber20] Andrea E Berndt. "Sampling methods". In: *Journal of Human Lactation* 36.2 (2020), pp. 224–226 (cit. on pp. 92, 93).

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006 (cit. on pp. 42–44).

[BKS01] S. Börzsönyi, D. Kossmann, and K. Stocker. "The Skyline Operator". In: *Proceedings of ICDE '01*. Washington, DC, USA: IEEE, 2001, pp. 421–430 (cit. on pp. 52, 59, 69).

[BNJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". In: *J. Mach. Learn. Res.* 3.null (2003), 993–1022 (cit. on p. 46).

[Bre01] L. Breiman. "Random Forests". In: *Machine Learning* 45 (Oct. 2001), pp. 5–32 (cit. on p. 44).

[BS13] S. Bandyopadhyay and S. Saha. *Unsupervised Classification*. Berlin Heidelberg, Germany: Springer Verlag, 2013 (cit. on pp. 23, 36).

[CG16] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, 785–794 (cit. on p. 43).

[CH74] Tadeusz Caliński and J. Harabasz. "A Dendrite Method for Cluster Analysis". In: *Communications in Statistics - Theory and Methods* 3 (Jan. 1974), pp. 1–27 (cit. on p. 39).

[Cha+06] C.-Y. Chan, H. V. Jagadish, K.-L. Tan, A. K. H. Tung, and Z. Zhang. "Finding K-dominant Skylines in High Dimensional Space". In: *SIGMOD '06*. Chicago, IL, USA: ACM, 2006, pp. 503–514 (cit. on p. 79).

[Cha+17] D. Chatzakou, N. Kourtellis, J. Blackburn, E. De Cristofaro, G. Stringhini, and A. Vakali. "Mean Birds: Detecting Aggression and Bullying on Twitter". In: *WebSci* (2017) (cit. on pp. 109, 116, 117).

[Chu+10] Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. "Who is Tweeting on Twitter: Human, Bot, or Cyborg?" In: *ACSAC (2010)*. 2010 (cit. on pp. 109, 206).

[Coh88] J. Cohen. *Statistical Power Analysis for the Behavioral Sciences*. eng. 2. ed. Literaturverz. S. 553 - 558. Hillsdale, NJ [u.a.]: Erlbaum, 1988, XXI, 567 S. (Cit. on pp. 21, 100).

[CZC17] Hongyun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. "A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications". In: *CoRR* abs/1709.07604 (2017). arXiv: 1709.07604 (cit. on pp. 215, 216, 220).

[D'E14] Ben D'Exelle. "Representative Sample". In: *Encyclopedia of Quality of Life and Well-Being Research*. Ed. by Alex C. Michalos. Dordrecht: Springer Netherlands, 2014, pp. 5511–5513 (cit. on p. 89).

[Dan11] Johnnie Daniel. *Sampling essentials: Practical guidelines for making sampling choices*. Sage Publications, 2011 (cit. on pp. 92, 93).

[DB79] D. L. Davies and D. W. Bouldin. "A Cluster Separation Measure". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1.2 (1979), pp. 224–227 (cit. on p. 39).

[DDZ15] Shifei Ding, Mingjing Du, and Hong Zhu. "Survey on Granularity Clustering". In: *Cognitive Neurodynamics* 9 (July 2015) (cit. on pp. 4, 5).

[Deb92] B. Debord. "An Axiomatic Characterization of Borda's k-choice Function". In: *Social Choice and Welfare* 9.4 (1992), pp. 337–343 (cit. on p. 65).

[DLR77] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. "Maximum Likelihood from Incomplete Data via the EM - Algorithm plus Discussions on the Paper". In: 1977 (cit. on pp. 35, 110).

[DN+20] Arash Dargahi Nobari, Melika Sarraf, Mahmood Neshati, and Farnaz Daneshvar. "Characteristics of Viral Messages on Telegram; The World's Largest Hybrid Public and Private Messenger". In: *Expert Systems with Applications* 168 (Nov. 2020), p. 114303 (cit. on p. 206).

[Don00] Stijn Dongen. "Graph Clustering by Flow Simulation". In: *PhD thesis, Center for Math and Computer Science (CWI)* (May 2000) (cit. on p. 229).

[Dow] Allen B. Downey. “Generating Pseudo-Random Floating-Point Values”. In: (cit. on p. 90).

[DYB10] Jean-Charles Delvenne, Sophia Yaliraki, and Mauricio Barahona. “Stability of Graph Communities across Time Scales”. In: *Proceedings of the National Academy of Sciences of the United States of America* 107 (July 2010), pp. 12755–60 (cit. on p. 18).

[ECR20] M. Espinosa, R. Centeno, and A. Rodrigo. “Analyzing User Profiles for Detection of Fake News Spreaders on Twitter - Notebook for PAN at CLEF 2020”. In: Sept. 2020 (cit. on p. 109).

[EH19] Jesper E. van Engelen and Holger H. Hoos. “A Survey on Semi-Supervised Learning”. In: *Machine Learning* 109 (2019), pp. 373–440 (cit. on p. 29).

[End21] Rainer Endres. *Models and Methods to Trace Information Diffusion on Telegram*. 2021 (cit. on pp. 181, 182).

[ES00] Martin Ester and Jörg Sander. “Clustering”. In: *Knowledge Discovery in Databases: Techniken und Anwendungen*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 45–105 (cit. on pp. 23, 34, 36, 38, 40, 41, 43).

[ESA12] Javid Ebrahimi and Mohammad Saniee Abadeh. “Semi Supervised Clustering: A Pareto Approach”. In: *Machine Learning and Data Mining in Pattern Recognition*. Ed. by Petra Perner. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 237–251 (cit. on pp. 55, 78).

[Est+96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: *Knowledge Discovery and Data Mining*. 1996 (cit. on p. 36).

[FB92] A. Ferligoj and V. Batagelj. “Direct Multicriteria Clustering Algorithms”. In: *Journal of Classification* 9.1 (1992), pp. 43–61 (cit. on p. 77).

[FPSS96] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. “From Data Mining to Knowledge Discovery in Databases”. In: *AI Magazine* 17.3 (1996), p. 37 (cit. on pp. 7, 87).

[FPT04] Ludwig Fahrmeier, Iris Pigeot, and Gerhard Tutz. *Statistik: Der Weg zur Datenanalyse*. Jan. 2004 (cit. on pp. 27, 119).

[Fri01] Jerome H. Friedman. “Greedy function approximation: A gradient boosting machine.” In: *The Annals of Statistics* 29.5 (2001), pp. 1189 –1232 (cit. on p. 43).

[Fu19] C. Fu. “Tracking User-role Evolution via Topic Modeling in Community Question Answering”. In: *Information Processing & Management* 56.6 (2019), p. 102075 (cit. on pp. 109, 110, 176).

[GEW06] Pierre Geurts, Damien Ernst, and Louis Wehenkel. "Extremely Randomized Trees". In: *Mach. Learn.* 63.1 (2006), 3–42 (cit. on p. 44).

[Gim+18] Henner Gimpel, Florian Haamann, Manfred Schoch, and Marcel Wittich. "User Roles in Online Political Discussion: A Typology based on Twitter Data from the German Federal Election 2017". In: June 2018 (cit. on p. 206).

[GMW07] Guojun Gan, Chaoqun Ma, and Jianhong Wu. *Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability)*. Society for Industrial and Applied Mathematics, 2007 (cit. on pp. 28, 32, 34, 36, 98).

[GN02] Michelle Girvan and Mark Newman. "Community structure in social and biological networks." In: *Proceedings of the National Academy of Sciences of the United States of America* 99 (July 2002), pp. 7821–6 (cit. on p. 18).

[Gui+13] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A. Zighed. "Information Diffusion in Online Social Networks: A Survey". In: *SIGMOD Rec.* 42.2 (2013), 17–28 (cit. on p. 210).

[Hal99] Mark A. Hall. *Correlation-based Feature Selection for Machine Learning*. 1999 (cit. on p. 20).

[HBB10] Matthew Hoffman, Francis Bach, and David Blei. "Online Learning for Latent Dirichlet Allocation". In: *Advances in Neural Information Processing Systems*. Ed. by J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta. Vol. 23. Curran Associates, Inc., 2010 (cit. on p. 46).

[HH09] Courtenay Honeycutt and Susan C. Herring. "Beyond Microblogging: Conversation and Collaboration via Twitter". In: *2009 42nd Hawaii International Conference on System Sciences* (2009), pp. 1–10 (cit. on p. 18).

[HMT10] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*. 2010. arXiv: `0909.4061 [math.NA]` (cit. on p. 46).

[Hoa61] C. A. R. Hoare. "Algorithm 64: Quicksort". In: *Commun. ACM* 4.7 (1961), p. 321 (cit. on p. 68).

[Hos+20] Mohamad Hoseini, Philipe Melo, Manoel Júnior, Fabrício Benevenuto, Balakrishnan Chandrasekaran, Anja Feldmann, and Savvas Zannettou. "Demystifying the Messaging Platforms' Ecosystem Through the Lens of Twitter". In: Oct. 2020 (cit. on p. 206).

[HPK11] J. Han, J. Pei, and M. Kamber. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2011 (cit. on p. 30).

[HR21] J. Hacker and K. Riemer. "Identification of User Roles in Enterprise Social Networks: Method Development and Application". In: *Business & Information Systems Engineering* 63 (Aug. 2021) (cit. on p. 110).

[Hua+11] Zhenhua Huang, Yang Xiang, Bo Zhang, and Xiaoling Liu. "A Clustering Based Approach for Skyline Diversity". In: *Expert Syst. Appl.* 38.7 (July 2011), pp. 7984–7993 (cit. on pp. 55, 78).

[HZC19] Ali Hashemi and Mohammad Ali Zare Chahooki. "Telegram Group Quality Measurement by User Behavior Analysis". In: *Social Network Analysis and Mining* 9 (July 2019) (cit. on p. 206).

[Jai10] A. K. Jain. "Data Clustering: 50 Years Beyond K-means". In: *Pattern Recogn. Lett.* 31.8 (June 2010), pp. 651–666 (cit. on pp. 34, 53, 67).

[Jav+07] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. "Why we Twitter: Understanding microblogging usage and communities". In: *of the 9th WebKDD and 1st SNA* 43 (Jan. 2007), pp. 56–65 (cit. on pp. 16, 18, 25, 109).

[JNH07] L. Jing, M. K. Ng, and J. Z. Huang. "An Entropy Weighting k-Means Algorithm for Subspace Clustering of High-Dimensional Sparse Data". In: *IEEE Transactions on Knowledge and Data Engineering* 19.8 (2007), pp. 1026–1041 (cit. on pp. 55, 78).

[JNR05] Anil Jain, Karthik Nandakumar, and Arun Ross. "Score Normalization in Multimodal Biometric Systems". In: *Pattern Recognition* 38.12 (2005), pp. 2270–2285 (cit. on p. 28).

[KA09] T. Kamishima and S. Akaho. "Efficient Clustering for Orders". In: *Mining Complex Data*. Ed. by Djamel A. Zighed, Shusaku Tsumoto, Zbigniew W. Ras, and Hakim Hacid. Springer Berlin Heidelberg, 2009, pp. 261–279 (cit. on p. 78).

[Kaf+14] Eleanna Kafeza, Andreas Kanavos, Christos Makris, and Pantelis Vikatos. "Predicting Information Diffusion Patterns in Twitter". In: vol. 436. Sept. 2014 (cit. on p. 229).

[Kan+02] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. "An Efficient k-means Clustering Algorithm: Analysis and Implementation". In: *IEEE TPAMI* 24.7 (2002), pp. 881–892 (cit. on pp. 55, 78).

[Kao+19] H. T. Kao, S. Yan, D. Huang, N. Bartley, H. Hosseinmardi, and E. Ferrara. "Understanding Cyberbullying on Instagram and Ask.Fm via Social Role Detection". In: *WWW '19 Companion*. 2019 (cit. on pp. 109, 117).

[KDN08] John Krumm, Nigel Davies, and Chandra Narayanaswami. "User-Generated Content". In: *IEEE Pervasive Computing* 7.4 (2008), pp. 10–11 (cit. on p. 4).

[KEW11] W. Kießling, M. Endres, and F. Wenzel. "The Preference SQL System - An Overview". In: *Bulletin of the Technical Commitee on Data Engineering* 34.2 (2011), pp. 11–18 (cit. on pp. 54, 59).

[Kha17] Karim Khalifa. *Twitter Retweet-Cascades Classification using Graph Summarization and Hierarchical Clustering*. 2017 (cit. on pp. 213, 222).

[Kie02] Werner Kießling. "Foundations of Preferences in Database Systems". In: *Proceedings of VLDB '02*. Hong Kong, China: VLDB, 2002, pp. 311–322 (cit. on p. 59).

[Kim+07] D. Kim, K. S. Kim, K. H. Park, J. H. Lee, and K. M. Lee. "A Music Recommendation System with a Dynamic k-means Clustering Algorithm". In: *ICMLA*. 2007 (cit. on p. 79).

[Kim+19] SunYoung Kim, Y-S Park, Honghyun Cho, and Jang-Won Kang. "Insider Threat Detection Based on User Behavior Modeling and Anomaly Detection Algorithms". In: *Applied Sciences* 9 (Sept. 2019), p. 4018 (cit. on p. 175).

[KJ13] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. New York, NY: Springer, 2013 (cit. on p. 26).

[Kle98] J. Kleinberg. "Authoritative Sources in a Hyperlinked Environment". In: *J Assoc Comput Mach* 46 (Nov. 1998) (cit. on p. 18).

[Kli+17] André Klima, Helmut Küchenhoff, Mirjam Selzer, and Paul W Thurner. *Exit Polls und Hybrid-Modelle*. Springer, 2017 (cit. on p. 176).

[Kni+12] B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, and C. Newell. "Explaining the User Experience of Recommender Systems". In: *User Modeling and User-Adapted Interaction* 22.4-5 (Oct. 2012), pp. 441–504 (cit. on p. 73).

[Kot13] Sotiris Kotsiantis. "Decision trees: A recent overview". In: *Artificial Intelligence Review* (Apr. 2013), pp. 1–23 (cit. on pp. 43, 44).

[Kou+15] Danai Koutra, U Kang, Jilles Vreeken, and Christos Faloutsos. "Summarizing and Understanding Large Graphs". In: *Stat. Anal. Data Min.* 8.3 (2015), 183–202 (cit. on p. 229).

[KP17] M. Kunaver and T. Porl. “Diversity in Recommender Systems A Survey”. In: *Know.-Based Syst.* 123.C (May 2017), pp. 154–162 (cit. on pp. 37, 73).

[KR16] Matthieu Komorowski and Jesse Raffa. “Markov Models and Cost Effectiveness Analysis: Applications in Medical Research”. In: Sept. 2016, pp. 351–367 (cit. on p. 176).

[KR90] Leonard Kaufman and Peter Rousseeuw. *Finding Groups in Data: An Introduction To Cluster Analysis*. Jan. 1990 (cit. on p. 38).

[KS16] Karim Khalifa and Tobias Strickfaden. *Graph Summarization using Dynamic Collapsing and MDL Summarizing*. 2016 (cit. on pp. 213, 214).

[Kwa+10] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. “What is Twitter, a Social Network or a News Media?” In: *Proceedings of the 19th International Conference on World Wide Web*. WWW ’10. Raleigh, North Carolina, USA: Association for Computing Machinery, 2010, 591–600 (cit. on p. 17).

[Lan+11] S. Landau, M. Leese, D. Stahl, and B. S. Everitt. *Cluster Analysis*. Wiley Series in Probability and Statistics. Wiley, 2011 (cit. on p. 31).

[LDB08] Renaud Lambiotte, Jean-Charles Delvenne, and Mauricio Barahona. “Laplacian Dynamics and Multiscale Modular Structure in Networks”. In: *arXiv* 1 (Dec. 2008) (cit. on p. 18).

[Leg+15] Phil Legg, Oliver Buckley, Michael Goldsmith, and Sadie Creese. “Automated Insider Threat Detection System Using User and Role-Based Profile Assessment”. In: *IEEE Systems Journal* 99 (June 2015), pp. 1–10 (cit. on p. 175).

[LG19] P. Lasek and J. Gryz. “Density-based Clustering with Constraints”. In: *Computer Science and Information Systems* 16 (Jan. 2019), pp. 7–7 (cit. on p. 110).

[Li+17] H. Li et al. “Bimodal Distribution and Co-Bursting in Review Spam Detection”. In: *WWW (2017)*. 2017 (cit. on p. 109).

[Li+22] Genghui Li, Zhenkun Wang, Qingfu Zhang, and Jianyong Sun. “Offline and Online Objective Reduction via Gaussian Mixture Model Clustering”. In: *IEEE Transactions on Evolutionary Computation* (2022), pp. 1–1 (cit. on p. 79).

[Lin+21] Qiuzhen Lin, Wu Lin, Zexuan Zhu, Maoguo Gong, Jianqiang Li, and Carlos A. Coello Coello. “Multimodal Multiobjective Evolutionary Optimization With Dual Clustering in Decision and Objective Spaces”. In: *IEEE Transactions on Evolutionary Computation* 25.1 (2021), pp. 130–144 (cit. on p. 79).

[LNN16] E. Lazaridou, A. Ntalla, and J. Novak. "Behavioural Role Analysis for Multi-faceted Communication Campaigns in Twitter". In: *WebSci (2016)*. 2016 (cit. on pp. 109, 116, 175, 206).

[LRU20] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of Massive Data Sets*. Cambridge university press, 2020 (cit. on p. 23).

[LZ20] Zhicheng Liu and Aoqian Zhang. "A Survey on Sampling and Profiling over Big Data (Technical Report)". In: *CoRR* abs/2005.05079 (2020). arXiv: 2005.05079 (cit. on p. 31).

[MAB09] Zaki Malik, Ihsan Akbar, and Athman Bouguettaya. "Web Services Reputation Assessment Using a Hidden Markov Model". In: *Service-Oriented Computing*. Ed. by Luciano Baresi, Chi-Hung Chi, and Jun Suzuki. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 576–591 (cit. on p. 176).

[Mac67] J. Macqueen. "Some Methods for Classification and Analysis of Multivariate Observations". In: *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*. 1967, pp. 281–297 (cit. on p. 67).

[Mah+20] Mohammad Sultan Mahmud, Joshua Zhexue Huang, Salman Salloum, Tamer Z Emara, and Kuanishbay Sadatdiynov. "A survey of data partitioning and sampling methods to support big data analysis". In: *Big Data Mining and Analytics* 3.2 (2020), pp. 85–101 (cit. on p. 92).

[Man08] Amit Mani. "Validation of PCA and LDA for SAR ATR". In: Dec. 2008, pp. 1 –6 (cit. on p. 46).

[Mar11] Stephen Marsland. *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC, 2011 (cit. on pp. 28, 30).

[May08] Antony Mayfield. "What is Social Media". In: (2008) (cit. on p. 16).

[MC88] Glenn W. Milligan and Martha Cooper. "A Study of Standardization of Variables in Cluster Analysis". In: *Journal of Classification* 5 (1988), pp. 181–204 (cit. on p. 119).

[McG+07] Mary McGlohon, Jure Leskovec, Christos Faloutsos, Matthew Hurst, and Natalie Glance. "Finding Patterns in Blog Shapes and Blog Evolution". In: (Jan. 2007) (cit. on p. 229).

[MMB09] Ujjwal Maulik, Anirban Mukhopadhyay, and Sanghamitra Bandyopadhyay. "Combining Pareto-Optimal Clusters using Supervised Learning for Identifying Co-Expressed Genes". In: *BMC bioinformatics* 10 (Feb. 2009), p. 27 (cit. on pp. 55, 78).

[Mos52] Claus Adolf Moser. "Quota sampling". In: *Journal of the Royal Statistical Society. Series A (General)* 115.3 (1952), pp. 411–423 (cit. on p. 94).

[MR10] Oded Maimon and Lior Rokach. “Introduction to Knowledge Discovery and Data Mining”. In: *Data Mining and Knowledge Discovery Handbook*. Ed. by Oded Maimon and Lior Rokach. Boston, MA: Springer US, 2010, pp. 1–15 (cit. on p. 11).

[MTL78] Robert McGill, John W. Tukey, and Wayne A. Larsen. “Variations of Box Plots”. In: *The American Statistician* 32.1 (1978), pp. 12–16 (cit. on p. 22).

[MU13] I. Mohamad and D. Usman. “Standardization and Its Effects on K-Means Clustering Algorithm”. In: *Research Journal of Applied Sciences, Engineering and Technology* 6 (Sept. 2013), pp. 3299–3303 (cit. on pp. 56, 79).

[Nar+17] Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. “graph2vec: Learning Distributed Representations of Graphs”. In: *CoRR* abs/1707.05005 (2017). arXiv: 1707.05005 (cit. on pp. 217, 229).

[New04] Mark Newman. “Power Laws, Pareto Distributions and Zipf’s Law”. In: *Contemporary Physics - CONTEMP PHYS* 46 (Dec. 2004) (cit. on pp. 25, 27).

[NL20] Lynnette Ng and Jia Loke. “Analysing Public Opinion and Misinformation in a COVID-19 Telegram Group Chat”. In: *IEEE Internet Computing* PP (Dec. 2020), pp. 1–1 (cit. on p. 206).

[Oa+16] Sirinya On-at, Arnaud Quirin, André Péninou, Nadine Baptiste-Jessel, Marie-Françoise Canut, and Florence Sèdes. “Taking into Account the Evolution of Users Social Profile: Experiments on Twitter and some Learned Lessons”. In: *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)* (2016), pp. 1–12 (cit. on p. 176).

[Osb10] Jason Osborne. “Improving your data transformations: Applying Box-Cox transformations as a best practice”. In: *Pract Assess Res Eval* 15 (Jan. 2010), pp. 1–9 (cit. on pp. 27, 119).

[OW15] Jonathan Obar and Steven Wildman. “Social Media Definition and the Governance Challenge: An Introduction to the Special Issue”. In: *SSRN Electronic Journal* (Jan. 2015) (cit. on p. 16).

[PA10] Grant Pannell and Helen Ashman. “Anomaly Detection over User Profiles for Intrusion Detection”. In: *Australian Information Security Management Conference* (Jan. 2010) (cit. on p. 175).

[Pau+11] Mari-Sanna Paukkeri, Ilkka Kivimäki, Santosh Tirunagari, Erkki Oja, and Timo Honkela. "Effect of Dimensionality Reduction on Different Distance Measures in Document Clustering". In: vol. 7064. Nov. 2011, pp. 167–176 (cit. on pp. 47, 95).

[PHL04] L. Parsons, E. Haque, and H. Liu. "Subspace Clustering for High Dimensional Data: A Review". In: *SIGKDD Explor. Newsl.* 6.1 (2004), pp. 90–105 (cit. on p. 79).

[Qah+15] Abdulhakim Qahtan, Basma Alharbi, Suojin Wang, and Xiangliang Zhang. "A PCA-Based Change Detection Framework for Multidimensional Data Streams". In: Aug. 2015 (cit. on p. 46).

[RN21] Stuart J. Russell and Peter Norvig. *Artificial Intelligence a Modern Approach*. Pearson Education, Inc., 2021 (cit. on p. 164).

[Roc+11] E. Rocha, A. P. Francisco., P. Caladoa, and H. Sofia-Pinto. "User Profiling on Twitter". In: *Semantic Web Journal* (2011) (cit. on pp. 109, 116, 175).

[Rou87] P. Rousseeuw. "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis". In: *Journal of Comp. and Applied Math.* 20 (1987), pp. 53 –65 (cit. on p. 38).

[RRS15] F. Ricci, L. Rokach, and B. Shapira. "Recommender Systems: Introduction and Challenges". In: *Recommender Systems Handbook*. Boston, MA: Springer US, 2015, pp. 1–34 (cit. on p. 52).

[RVW11] Francesca Rossi, Kristen Brent Venable, and Toby Walsh. *A Short Introduction to Preferences Between Artificial Intelligence and Social Choice*. Morgan & Claypool Publishers, 2011 (cit. on p. 56).

[Saw09] S. Sawilowsky. "New Effect Size Rules of Thumb". In: *Journal of Modern Applied Statistical Methods* 8 (Nov. 2009), pp. 597–599 (cit. on pp. 21, 100).

[SB18] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning - An Introduction (Second Edition)*. Adaptive Computation and Machine Learning. MIT Press, 2018 (cit. on p. 30).

[SB93] Frank Sonnenberg and J. Beck. "Markov Models in Medical Decision Making: A Practical Guide". In: *Medical decision making : an international journal of the Society for Medical Decision Making* 13 (Dec. 1993), pp. 322–38 (cit. on p. 176).

[Sch07] Satu Elisa Schaeffer. "Survey: Graph Clustering". In: *Comput. Sci. Rev.* 1.1 (2007), 27–64 (cit. on p. 229).

[Sen99] A. Sen. "The Possibility of Social Choice". In: *The American Economic Review* 89.3 (1999), pp. 349–378 (cit. on p. 65).

[Set10] Burr Settles. *Active Learning Literature Survey*. Tech. rep. University of Wisconsin-Madison, Department of Computer Sciences, July 2010 (cit. on p. 29).

[Shu+19] K. Shu, X. Zhou, S. Wang, R. Zafarani, and H. Liu. "The Role of User Profiles for Fake News Detection". In: *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ASONAM '19. Vancouver, British Columbia, Canada: Association for Computing Machinery, 2019, 436–439 (cit. on p. 109).

[SKT14] Arthur Szlam, Yuval Kluger, and Mark Tygert. *An Implementation of a Randomized Algorithm for Principal Component Analysis*. 2014. arXiv: `1412.3510 [stat.CO]` (cit. on p. 46).

[SL09] Marina Sokolova and Guy Lapalme. "A Systematic Analysis of Performance Measures for Classification Tasks". In: *Information Processing & Management* 45 (July 2009), pp. 427–437 (cit. on pp. 44, 45).

[SM14] M. Sarstedt and E. Mooi. "Cluster Analysis". In: *A Concise Guide to Market Research: The Process, Data, and Methods Using IBM SPSS Statistics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 273–324 (cit. on p. 53).

[Smi15] Lauren Smith. "I'm Going to Instagram It! An Analysis of Athlete Self-Presentation on Instagram". In: *Journal of Broadcasting & Electronic Media* In Press (May 2015) (cit. on p. 18).

[Sta] Statista. *Most popular social networks worldwide as of January 2022, ranked by number of monthly active users* (cit. on p. 16).

[Sut+16] Tole Sutikno, Lina Handayani, Deris Stiawan, Munawar Riyadi, and Imam Subroto. "WhatsApp, Viber and Telegram which is Best for Instant Messaging?" In: *International Journal of Electrical and Computer Engineering (IJECE)* 6 (June 2016), p. 909 (cit. on p. 206).

[SZ10] Renato Sato and Desiree Zouain. "Markov Models in Health Care". In: *Einstein* 8 (Sept. 2010) (cit. on p. 176).

[TB15] Duy Tin Truong and Roberto Battiti. "A Flexible Cluster-Oriented Alternative Clustering Algorithm for Choosing from the Pareto Front of Solutions". In: *Machine Learning* 98.1 (2015), pp. 57–91 (cit. on pp. 55, 78).

[TF13] Io Taxidou and Peter Fischer. "Realtime Analysis of Information Diffusion in Social Media". In: *Proc. VLDB Endow.* 6.12 (2013), 1416–1421 (cit. on p. 210).

[TF14] Io Taxidou and Peter M. Fischer. "Online Analysis of Information Diffusion in Twitter". In: *Proceedings of the 23rd International Conference on World Wide Web*. WWW '14 Companion. Seoul, Korea: Association for Computing Machinery, 2014, 1313–1318 (cit. on p. 210).

[Tha18] Alaa Tharwat. "Classification Assessment Methods: A Detailed Tutorial". In: *Applied Computing and Informatics, Vol. 17 No. 1, 2021, pp. 168-192 Emerald Publishing Limited* (Sept. 2018) (cit. on pp. 44, 45).

[THP08] Yuanyuan Tian, Richard Hankins, and Jignesh Patel. "Efficient Aggregation for Graph Summarization". In: June 2008, pp. 567–580 (cit. on p. 229).

[Thu+21] Paul W. Thurner, Fiona Kunz, Andrea Miclut, Ingrid Mauerer, André Klima, and Helmut Küchenhoff. "Die Schätzung von Wählerwanderungen zwischen den Bundestagswahlen 2013–2017 mithilfe von Online-Paneldaten und Aggregatdaten in Hybridmodellen". In: *Wahlen und Wähler: Analysen aus Anlass der Bundestagswahl 2017*. Ed. by Bernhard Weßels and Harald Schoen. Wiesbaden: Springer Fachmedien Wiesbaden, 2021, pp. 205–226 (cit. on p. 176).

[Tin+12] Ramine Tinati, Leslie Carr, Wendy Hall, and Jonny Bentwood. "Identifying Communicator Roles in Twitter". In: (Apr. 2012) (cit. on pp. 18, 109).

[UK20] Aleksandra Urman and Stefan Katz. "What they do in the Shadows: Examining the Far-Right Networks on Telegram". In: *Information, Communication & Society* 25 (Aug. 2020), pp. 1–20 (cit. on p. 206).

[Ukk11] A. Ukkonen. "Clustering Algorithms for Chains". In: *Journal of Machine Learning Research* 12 (2011), pp. 1389–1423 (cit. on p. 78).

[Var+14] O. Varol, E. Ferrara, C. L. Ogan, F. Menczer, and A. Flammini. "Evolution of Online User Behavior during a Social Upheaval". In: *WebSci* (2014) (cit. on pp. 109, 117, 175).

[VSM15] D. Virmani, T. Shweta, and G. Malhotra. "Normalization Based K Means Clustering Algorithm". In: *CoRR* abs/1503.00900 (2015). eprint: `1503.00900` (cit. on pp. 56, 79).

[Wei+12] S. Wei, N. Ye, S. Zhang, X. Huang, and J. Zhu. "Collaborative Filtering Recommendation Algorithm Based on Item Clustering and Global Similarity". In: *BIFE '12*. 2012, pp. 69–72 (cit. on p. 79).

[Weia] Eric W. Weisstein. *Covariance. From MathWorld–A Wolfram Web Resource. https://mathworld.wolfram.com/Covariance.html.* (visited on 02.01.2023) (cit. on p. 20).

[Weib] Eric W. Weisstein. *Statistical Correlation. From MathWorld–A Wolfram Web Resource. https://mathworld.wolfram.com/StatisticalCorrelation.html.* (visited on 02.02.2023) (cit. on p. 20).

[WFH11] Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques.* 3rd ed. Morgan Kaufmann Series in Data Management Systems. Amsterdam: Morgan Kaufmann, 2011 (cit. on p. 30).

[Wic+21] Maximilian Wich, Adrian Gorniak, Tobias Eder, Daniel Bartmann, Burak Çakici, and Georg Groh. "Introducing an Abusive Language Classification Framework for Telegram to Investigate the German Hater Community". In: (Sept. 2021) (cit. on p. 206).

[WWP88] S. J. Wan, S. K. M. Wong, and P. Prusinkiewicz. "An Algorithm for Multidimensional Data Clustering". In: *ACM Trans. Math. Softw.* 14.2 (June 1988), pp. 153–162 (cit. on pp. 55, 78).

[WX08] Donald Wunsch and Rui Xu. *Clustering.* John Wiley & Sons, 2008 (cit. on pp. 24, 28, 32, 34, 36, 97).

[WZZ15] Xi Wang, Zhiya Zuo, and Kang Zhao. "The Evolution and Diffusion of User Roles in Online Health Communities: A Social Support Perspective". In: Oct. 2015, pp. 48–56 (cit. on p. 176).

[YB09] Kwan Yi and Jamshid Beheshti. "A Hidden Markov Model-Based Text Classification of Medical Documents". In: *Journal of Information Science* 35 (Feb. 2009), pp. 67–81 (cit. on p. 176).

[YB14] Keming Yang and Ahmad Banamah. "Quota sampling as an alternative to probability sampling? An experimental study". In: *Sociological Research Online* 19.1 (2014), pp. 56–66 (cit. on p. 94).

[YHL15] Rose Yu, Xinran He, and Yan Liu. "GLAD: Group Anomaly Detection in Social Media Analysis". In: *ACM Transactions on Knowledge Discovery from Data* 10 (Oct. 2015), pp. 1–22 (cit. on p. 176).

[YS17] Ahmet Yayla and Anne Speckhard. "Telegram: the Mighty Application that ISIS Loves". In: *International Center for the Study of Violent Extremism (ICSVE)* (May 2017) (cit. on p. 206).

[YV15] Pinar Yanardag and S.V.N. Vishwanathan. "Deep Graph Kernels". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* KDD '15. Sydney, NSW, Australia: Association for Computing Machinery, 2015, 1365–1374 (cit. on p. 216).

[ZAL14] Reza Zafarani, Mohammad Ali Abbasi, and Huan Liu. *Social Media Mining: An Introduction*. New York, NY, USA: Cambridge University Press, 2014 (cit. on pp. 8, 25, 27, 87).

[Zam16] A. Zambelli. "A Data-Driven Approach to Estimating the Number of Clusters in Hierarchical Clustering". In: *F1000Research* 5 (Aug. 2016) (cit. on p. 99).

[Zan+17] Chengxi Zang, Peng Cui, Chaoming Song, Christos Faloutsos, and Wenwu Zhu. "Quantifying Structural Patterns of Information Cascades". In: Apr. 2017, pp. 867–868 (cit. on p. 229).

[ZC18] A. Zheng and A. Casari. *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly, 2018 (cit. on pp. 26, 119).

[ZHH02] Jianhan Zhu, Jun Hong, and John G. Hughes. "Using Markov Chains for Link Prediction in Adaptive Web Sites". In: *Soft-Ware 2002: Computing in an Imperfect World*. Ed. by David Bustard, Weiru Liu, and Roy Sterritt. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 60–73 (cit. on p. 176).

[Zhu08] Xiaojin Zhu. "Semi-Supervised Learning Literature Survey". In: *Comput Sci, University of Wisconsin-Madison* 2 (July 2008) (cit. on p. 29).

[ZK00] Dan Zwillinger and Stephen Kokoska. "CRC Standard Probability and Statistics Tables and Formulae". In: (Jan. 2000) (cit. on p. 25).

[Zyg17] Anna Zygmunt. "Role Identification of Social Networkers". In: *Encyclopedia of Social Network Analysis and Mining*. Ed. by Reda Alhajj and Jon Rokne. New York, NY: Springer New York, 2017, pp. 1–9 (cit. on p. 19).

[ZZX08] Z. Zhang, J. Zhang, and H. Xue. "Improved K-Means Clustering Algorithm". In: *CISP '08: Proceedings of the Congress on Image and Signal Processing 2008*. Vol. 5. 2008, pp. 169–172 (cit. on pp. 55, 78).

[Du+16] F. Du, Y. Liu, X. Liu, J. Sun, and Y. Jiang. "User Role Analysis in Online Social Networks Based on Dirichlet Process Mixture Models". In: *2016 International Conference on Advanced Cloud and Big Data (CBD)*. 2016, pp. 172–177 (cit. on p. 109).

List of Figures

List of Tables

Part IV

Appendix

Appendix A

Publications

Some contents of this thesis were published in the following peer-reviewed papers, journals and book chapters.

[EKR18] Markus Endres, Johannes Kastner, and Lena Rudenko. "Analyzing and Clustering Pareto-Optimal Objects in Data Streams". In: *Learning from Data Streams in Evolving Environments: Methods and Applications*. Ed. by Moamar Sayed-Mouchaweh. 2018 (cit. on pp. 51, 77).

[KE17] Johannes Kastner and Markus Endres. *Multidimensional Clustering Approaches for Pareto-Frontiers*. Tech. rep. 2017-02. Fakultät für Angewandte Informatik, 2017 (cit. on p. 51).

[KE19] Johannes Kastner and Markus Endres. "You Have the Choice: The Borda Voting Rule for Clustering Recommendations". In: *Advances in Databases and Information Systems - 23rd European Conference, ADBIS 2019, Bled, Slovenia, September 8-11, 2019, Proceedings*. Ed. by Tatjana Welzer, Johann Eder, Vili Podgorelec, and Aida Kamisalic Latific. Vol. 11695. Lecture Notes in Computer Science. Springer, 2019, pp. 321–336 (cit. on p. 51).

[KEK17] J. Kastner, M. Endres, and W. Kießling. "A Pareto-Dominant Clustering Approach for Pareto-Frontiers". In: *EDBT/ICDT '17, Venice, Italy, March 21-24, 2017*. Vol. 1810. Workshop Proceedings. 2017 (cit. on p. 51).

[KF21] Johannes Kastner and Peter M. Fischer. "Scalable and Explainable User Role Detection in Social Media". In: *New Trends in Database and Information Systems - ADBIS 2021 Short Papers, Doctoral Consortium and Workshops: DOING, SIMPDA, MADEISD, MegaData, CAoNS, Tartu,*

Estonia, August 24-26, 2021, Proceedings. Ed. by Ladjel Bellatreche, Marlon Dumas, Panagiotis Karras, Raimundas Matulevicius, Ahmed Awad, Matthias Weidlich, Mirjana Ivanovic, and Olaf Hartig. Vol. 1450. Communications in Computer and Information Science. Springer, 2021, pp. 263–275 (cit. on pp. 83, 111).

[KF23] Johannes Kastner and Peter M. Fischer. "Detecting and Analyzing Fine-Grained User Roles in Social Media". In: *Computer Science and Information Systems*. 2023 (cit. on pp. 83, 111).

[KRE19] J. Kastner, N. Ranitovic, and M. Endres. "The Borda Social Choice Movie Recommender". In: *BTW '19, 4.-8.3.2019 in Rostock, Germany*. 2019, pp. 499–502 (cit. on pp. 51, 73, 76, 80).

Appendix B

Teaching

B.1 Lectures, Courses, & Seminars

An overview of lectures, courses and seminars the author of this thesis assisted to Prof. Dr. Peter M. Fischer during his time as a researcher at the University of Augsburg.

- Datenbanksysteme I
- Datenbankprogrammierung (Oracle)
- Analyzing Massive Data Sets
- Seminar Datenbanksysteme für Bachelor
- Seminar Informationssysteme für Geoinformatiker
- Seminar Informationssysteme für Master

B.2 Supervised Theses

An overview of students theses and other students projects the author supervised during his time as a researcher at the University of Augsburg.

B.2.1 Bachelor Theses

[Fra20] Michael Franz. *Klassifizierung erkannter Nutzergruppen in Social Media.* 2020.

[Lin20b] Nicole Lindolf. *Skalierung und Stabilität der Erkennung von Nutzerrollen in sozialen Medien.* 2020.

[Lis19] Rebecca Listle. *Entwicklung und Implementierung von Matchingverfahren für ein Skill-Management-System anhand einer Neo4j-Graphdatenbank.* 2019.

[Nac20] Luise Nachbar. *Entwicklung von Benutzerrollen in sozialen Strömen.* 2020.

[Sei19] Jonas Seitz. *Skalierbare Analyse von Autoren- und Artikelähnlichkeiten in der Wikipedia.* 2019.

[Sof22] Ahmed Sofu. *Analyse und Klassifkation von Benutzerclustern in Telegram Datensätzen.* 2022.

[Som22] Felix Sommer. *Vergleichende Analyse von Benutzerclustern von Telegram Gruppenverläufen.* 2022.

[Web22] Florian Weber. *Konzeptionierung und Entwicklung eines Webapp-Tools zur statistischen Analyse von hierarchischen Clusterverfahren.* 2022.

B.2.2 Master Theses

[Hof18b] Moritz Hofmaier. *Personalized Route Planning with User Preferences - Design and Implementation.* 2018.

[Mac23] Felix Mack. *Evaluating and Modelling Evolution in User Role Detection.* 2023 (cit. on pp. 91–94, 167).

[Ran18] Nemanja Ranitovic. *Konzeption und Entwicklung eines Recommender Systems zur Analyse von Clusterverfahren.* 2018.

[Thi19] Tobias von Thienen. *Erkennung von Benutzerrollen auf sozialen Strömen mit Clusteringverfahren.* 2019.

B.2.3 Projektmodule

[Fre21] Jonathan Freund. *Graph Embeddings for Information Diffusion Graphs.* 2021.

[Hof18a] Moritz Hofmaier. *Ein Java Programm zum Test des Borda Sozialwahlverfahrens in Verbindung mit dem agglomerativen hierarchischen Clusterverfahren.* 2018.

[Mac22] Felix Mack. *Analysis of User Roles Within and Between Data Sets.* 2022.

B.2.4 Forschungsmodule

[Kuc22] Victoria Kuch. *Forschungsmodul zum Embedding und Clustern von Ausbreitungsgraphen.* 2022.

[Lin20a] Nicole Lindolf. *Finden von Nutzerrollen in Twitter Datensätzen.* 2020.

[MLL23] Ram Mosco, Adelina La, and Lennart Linz. *Nutzerrollen über verschiedene Sportevents mit größeren Samplezahlen.* 2023.