A deep matrix factorization method for learning attribute representations

George Trigeorgis, Konstantinos Bousmalis, Stefanos Zafeiriou, Björn Schuller

Angaben zur Veröffentlichung / Publication details:

Trigeorgis, George, Konstantinos Bousmalis, Stefanos Zafeiriou, and Björn Schuller. 2017. "A deep matrix factorization method for learning attribute representations." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (3): 417–29. https://doi.org/10.1109/tpami.2016.2554555.

A Deep Matrix Factorization Method for Learning Attribute Representations

George Trigeorgis, Konstantinos Bousmalis, *Member, IEEE*,
Stefanos Zafeiriou, *Member, IEEE*, and Björn W. Schuller, *Senior Member, IEEE*

Abstract—Semi-Non-negative Matrix Factorization is a technique that learns a low-dimensional representation of a dataset that lends itself to a clustering interpretation. It is possible that the mapping between this new representation and our original data matrix contains rather complex hierarchical information with implicit lower-level hidden attributes, that classical one level clustering methodologies cannot interpret. In this work we propose a novel model, Deep Semi-NMF, that is able to learn such hidden representations that allow themselves to an interpretation of clustering according to different, unknown attributes of a given dataset. We also present a semi-supervised version of the algorithm, named Deep WSF, that allows the use of (partial) prior information for each of the known attributes of a dataset, that allows the model to be used on datasets with mixed attribute knowledge. Finally, we show that our models are able to learn low-dimensional representations that are better suited for clustering, but also classification, outperforming Semi-Non-negative Matrix Factorization, but also other state-of-the-art methodologies variants.

Index Terms—Semi-NMF, deep semi-NMF, unsupervised feature learning, face clustering, semi-supervised learning, Deep WSF, WSF, matrix factorization, face classification

1 Introduction

Matrix factorization is a particularly useful family of techniques in data analysis. In recent years, there has been a significant amount of research on factorization methods that focus on particular characteristics of both the data matrix and the resulting factors. Non-negative matrix factorization (NMF), for example, focuses on the decomposition of non-negative multivariate data matrix X into factors Z and H that are also non-negative, such that $X \approx ZH$. The application area of the family of NMF algorithms has grown significantly during the past years. It has been shown that they can be a successful dimensionality reduction technique over a variety of areas including, but not limited to, environmetrics [1], microarray data analysis [2], [3], document clustering [4], face recognition [5], [6], blind audio source separation [7] and more. What makes NMF algorithms particularly attractive is the non-negativity constraints imposed on the factors they produce, allowing for better interpretability. Moreover, it has been shown that NMF variants (such as the Semi-NMF) are equivalent to a soft version of k-means clustering, and that in fact, NMF variants are expected to perform better than k-means clustering particularly when the data is not distributed in a spherical manner [8], [9].

- *G. Trigeorgis, S. Zafeiriou, and B.W. Schuller are with the Department of Computing, Imperial College London, SW7 2RH, London, United Kingdom. E-mail: gt108@ic.ac.uk, {s.zafeiriou, bjoern.schuller}@imperial.ac.uk.*
- *K. Bousmalis is with Google Research, Mountain View, CA 94043. E-mail: konstantinos@google.com.*

In order to extend the applicability of NMF in cases where our data matrix X is not strictly non-negative, [8] introduced Semi-NMF, an NMF variant that imposes non-negativity constraints only on the second factor H, but allows mixed signs in both the data matrix X and the first factor Z. This was motivated from a clustering perspective, where Z represents cluster centroids, and H represents soft membership indicators for every data point, allowing Semi-NMF to learn new lower-dimensional features from the data that have a convenient clustering interpretation.

It is possible that the mapping Z between this new representation H and our original data matrix X contains rather complex hierarchical and structural information. Such a complex dataset X is produced by a multi-modal data distribution which is a mixture of several distributions, where each of these constitutes an *attribute* of the dataset. Consider for example the problem of mapping images of faces to their identities: a face image also contains information about attributes like pose and expression that can help identify the person depicted. One could argue that by further factorizing this mapping Z, in a way that each factor adds an extra layer of abstraction, one could automatically learn such latent attributes and the intermediate hidden representations that are implied, allowing for a better higher-level feature representation H. In this work, we propose Deep Semi-NMF, a novel approach that is able to factorize a matrix into multiple factors in an unsupervised fashion—see Fig. 1, and it is therefore able to learn multiple hidden representations of the original data. As Semi-NMF has a close relation to k-means clustering, Deep Semi-NMF also has a clustering interpretation according to the different latent attributes of our dataset, as demonstrated in Fig. 2.

It might be the case that the different attributes of our data are not latent. If those are known and we actually have

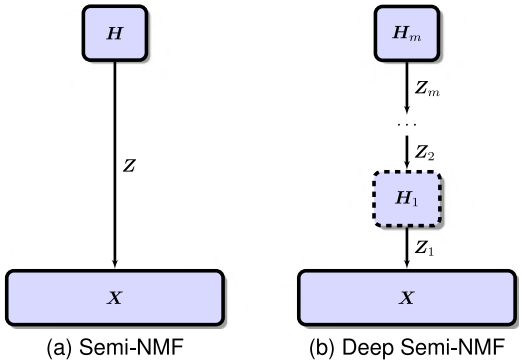


Fig. 1. (a) A Semi-NMF model results in a linear transformation of the initial input space. (b) Deep Semi-NMF learns a hierarchy of hidden representations that aid in uncovering the final lower-dimensional representation of the data.

some label information about some or all of our data, we would naturally want to leverage it and learn representations that would make the data more separable according to each of these attributes. To this effect, we also propose a weakly-supervised Deep Semi-NMF (Deep WSF), a technique that is able to learn, in a semi-supervised manner, a hierarchy of representations for a given dataset. Each level of this hierarchy corresponds to a specific attribute that is known a priori, and we show that by incorporating partial label information via graph regularization techniques we are able to perform better than with a fully unsupervised Deep Semi-NMF in the task of classifying our dataset of faces according to different attributes, when those are known. We also show that by initializing an unsupervised Deep Semi-NMF with the weights learned by a Deep WSF we are able to improve the clustering performance of the Deep Semi-NMF. This could be particularly useful if we have, as in our example, a small dataset of images of faces with partial attribute labels and a larger one with no attribute labels. By initializing a Deep Semi-NMF with the weights learned with Deep WSF from the small labelled dataset we can leverage all the information we have and allow our unsupervised model to uncover better representations for our initial data on the task of clustering faces.

Relevant to our proposal are hierarchical clustering algorithms [10], [11] which are popular in gene and document clustering applications. These algorithms typically abstract the initial data distribution as a form of tree called a *dendrogram*, which is useful for analysing the data and help identify genes that can be used as biomarkers or topics of a collection of documents. This makes it hard to incorporate out-of-sample data and prohibits the use of other techniques than clustering.

There are certain graphical models that share some similarities with our work as well. In particular, the work of Simon Prince et al. [12] presents a generative model for face recognition targeted for images of faces with a large pose variation. We would like to make clear that '*Tied Factor Analysis*' is a strictly supervised technique with a strong Bayesian flavour specifically aimed for face *recognition* where ours is a (weakly) semi-supervised matrix factorisation method with a more general application. In summary, '*Tied Factor Analysis*' *a)* requires the attribute information for all samples, *b)* it can only take into a account in the optimisation procedure a *single* attribute, and specifically is only formulated to only use the pose information, *c)* it creates multiple models; one for each pose combination (e.g., mapping profile to frontal pose,

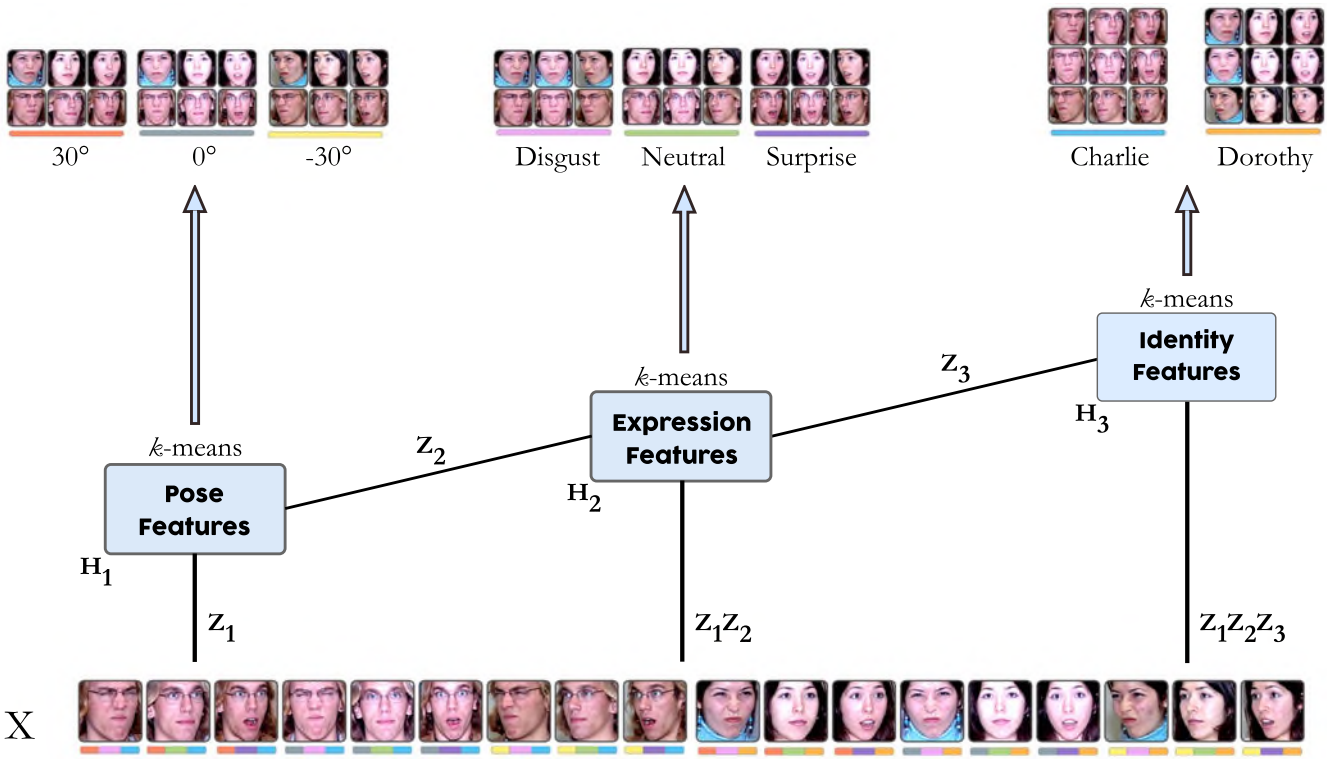


Fig. 2. A Deep Semi-NMF model learns a hierarchical structure of features, with each layer learning a representation suitable for clustering according to the different attributes of our data. In this simplified, for demonstration purposes, example from the CMU Multi-PIE database, a Deep Semi-NMF model is able to simultaneously learn features for pose clustering (H_1), for expression clustering (H_2), and for identity clustering (H_3). Each of the images in **X** has an associated colour coding that indicates its memberships according to each of these attributes (pose/expression/identity).

frontal to profile pose etc.) *d)* it is explicitly targeted for face recognition. In contrast our work *a)* can utilise only a partial prior information (for no or some samples) for each attribute, *b)* it can leverage information from an arbitrary number of attributes, *c)* we only have a single model which is jointly trained for all attributes and finally *d)* it is a matrix factorisation technique for finding the best low-dimensional representation for the task-at-hand (e.g., classification, regression etc.). Furthermore, our methodology can be used, as we show in the experiments, in a synergistic fashion with TFA in order to improve accuracy.

Another line of work which is related to ours is multi-label learning [13]. Multi-label learning techniques rely on the correlations [14] that exist between different attributes to extract better features. We are not interested in cases where there is complete knowledge about each of the attributes of the dataset but rather we propose a new paradigm of learning representations where have data with only partly annotated attributes. An example of this is a mixture of datasets where each one has label information about a different set of attributes. In this new paradigm we can *not* leverage the correlations between the attribute labels and we rather rely on the hierarchical structure of the data to uncover relations between the different dataset attributes. To the best of our knowledge this is the first piece of work that tries to automatically discover the representations for different (known and unknown) attributes of a dataset with an application to a multi-modal application such as face clustering.

The novelty of this work can be summarised as follows: *(1)* we outline a novel deep framework[1] for matrix factorization suitable for clustering of multimodally distributed objects such as faces, *(2)* we present a greedy algorithm to optimize the factors of the Semi-NMF problem, inspired by recent advances in deep learning [16], *(3)* we evaluate the representations learned by different NMF–variants in terms of clustering performance, *(4)* present the Deep WSF model that can use already known (partial) information for the attributes of our data distribution to extract better features for our model, and *(5)* demonstrate how to improve the performance of Deep Semi-NMF, by using the existing weights from a trained Deep WSF model.

2 Background

In this work, we assume that our data is provided in a matrix form $\boldsymbol{X} \in \mathbb{R}^{p \times n}$, i.e., $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$ is a collection of n data vectors as columns, each with p features. Matrix factorization aims at finding factors of $\boldsymbol{X}$ that satisfy certain constraints. In Singular Value Decomposition (SVD) [17], the method that underlies Principal Component Analysis (PCA) [18], we factorize $\boldsymbol{X}$ into two factors: the loadings or bases $\boldsymbol{Z} \in \mathbb{R}^{p \times k}$ and the features or components $\boldsymbol{H} \in \mathbb{R}^{k \times n}$, without imposing any sign restrictions on either our data or the resulting factors. In Non-negative Matrix Factorization (NMF) [19] we assume that all matrices involved contain only non-negative elements[2], so we try to approximate a factorization $\boldsymbol{X}^{+} \approx \boldsymbol{Z}^{+}\boldsymbol{H}^{+}$.

1. A preliminary version of this work has appeared in [15].

2. When not clear from the context we will use the notation $\boldsymbol{A}^{+}$ to state that a matrix $\boldsymbol{A}$ contains only non-negative elements. Similarly, when not clear, we will use the notation $\boldsymbol{A}^{\pm}$ to state that $\boldsymbol{A}$ may contain any real number.

2.1 Semi-NMF

In turn, Semi-NMF [8] relaxes the non-negativity constrains of NMF and allows the data matrix $\boldsymbol{X}$ and the loadings matrix $\boldsymbol{Z}$ to have mixed signs, while restricting only the features matrix $\boldsymbol{H}$ to comprise of strictly non-negative components, thus approximating the following factorization:

$$\boldsymbol{X}^{\pm} \approx \boldsymbol{Z}^{\pm}\boldsymbol{H}^{+}. \tag{1}$$

This is motivated from a clustering perspective. If we view $\boldsymbol{Z} = [z_1, z_2, \ldots, z_k]$ as the cluster centroids, then $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n]$ can be viewed as the cluster indicators for each datapoint.

In fact, if we had a matrix $\boldsymbol{H}$ that was not only non-negative but also orthogonal, such that $\boldsymbol{H}\boldsymbol{H}^T = \boldsymbol{I}$ [8], then every column vector would have only one positive element, making Semi-NMF equivalent to k-means, with the following cost function:

$$C_{k\text{-means}} = \sum_{i=1}^{n}\sum_{j=1}^{k} h_{ki}\|\boldsymbol{x}_i - z_k\|^2 = \|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{H}\|_F^2, \tag{2}$$

where $\|\cdot\|$ denotes the L_2-norm of a vector and $\|\cdot\|_F$ the Frobenius norm of a matrix.

Thus Semi-NMF, which does not impose an orthogonality constraint on its features matrix, can be seen as a soft clustering method where the features matrix describes the compatibility of each component with a cluster centroid, a base in $\boldsymbol{Z}$. In fact, the cost function we optimize for approximating the Semi-NMF factors is indeed:

$$C_{\text{Semi-NMF}} = \|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{H}\|_F^2. \tag{3}$$

We optimize $C_{\text{Semi}-\text{NMF}}$ via an alternate optimization of $\boldsymbol{Z}^{\pm}$ and $\boldsymbol{H}^{+}$: we iteratively update each of the factors while fixing the other, imposing the non-negativity constrains only on the features matrix $\boldsymbol{H}$:

$$\boldsymbol{Z} \leftarrow \boldsymbol{X}\boldsymbol{H}^{\dagger}, \tag{4}$$

where $\boldsymbol{H}^{\dagger}$ is the Moore-Penrose pseudo-inverse of $\boldsymbol{H}$, and

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \odot \sqrt{\frac{[\boldsymbol{Z}^{\top}\boldsymbol{X}]^{\text{pos}} + [\boldsymbol{Z}^{\top}\boldsymbol{Z}]^{\text{neg}}\boldsymbol{H}}{[\boldsymbol{Z}^{\top}\boldsymbol{X}]^{\text{neg}} + [\boldsymbol{Z}^{\top}\boldsymbol{Z}]^{\text{pos}}\boldsymbol{H}}}, \tag{5}$$

where ϵ is a small number to avoid division by zero, $\boldsymbol{A}^{\text{pos}}$ is a matrix that has the negative elements of matrix $\boldsymbol{A}$ replaced with 0, and similarly $\boldsymbol{A}^{\text{neg}}$ is one that has the positive elements of $\boldsymbol{A}$ replaced with 0:

$$\forall i, j \;.\; \mathbf{A}_{ij}^{\text{pos}} = \frac{|\mathbf{A}_{ij}| + \mathbf{A}_{ij}}{2}, \quad \mathbf{A}_{ij}^{\text{neg}} = \frac{|\mathbf{A}_{ij}| - \mathbf{A}_{ij}}{2}. \tag{6}$$

2.2 State-of-the-Art for Learning Features for Clustering Based on NMF-Variants

In this work, we compare our method with, among others, the state-of-the-art NMF techniques for learning features for the purpose of clustering. Cai et al. [20] proposed a graph-regularized NMF (GNMF) which takes into account the intrinsic geometric and discriminating structure of the data space, which is essential to the real-world applications, especially in the area of clustering. To accomplish this,

GNMF constructs a nearest neighbor graph to model the manifold structure. By preserving the graph structure, it allows the learned features to have more discriminating power than the standard NMF algorithm, in cases that the data are sampled from a submanifold which lies in a higher dimensional ambient space.

Closest to our proposal is recent work that has presented NMF-variants that factorize X into more than two factors. Specifically, Ahn et al. [21] have demonstrated the concept of Multi-Layer NMF on a set of facial images and [22], [23], [24] have proposed similar NMF models that can be used for Blind Source Separation, classification of digit images (MNIST), and documents. The representations of the Multi-layer NMF however do not lend themselves to a clustering interpretation, as the representations learned from our model. Although the Multi-layer NMF is a promising technique for learning hierarchies of features from data, we show in this work that our proposed model, the Deep Semi-NMF outperforms the Multi-layer NMF and, in fact, all models we compared it with on the task of feature learning for clustering images of faces.

2.3 Semi-Supervised Matrix Factorization

For the case of the proposed Deep WSF algorithms, we also evaluate our method with previous semi-supervised non-negative matrix factorization techniques. These include the Constrained Nonnegative Matrix Factorization (CNMF) [25], and the Discriminant Nonnegative Matrix Factorization (DNMF) [26]. Although both take label information as additional constraints, the difference between these is that CNMF uses the label information as hard constrains on the resulting features H, whereas DNMF tries to use the Fisher Criterion in order to incorporate discriminant information in the decomposition [26]. Both approaches only work for cases where we want to encode the prior information of only one attribute, in contrast to the proposed Deep WSF model.

3 Deep Semi-NMF

In Semi-NMF the goal is to construct a low-dimensional representation H^+ of our original data $X^\pm$, with the bases matrix $Z^\pm$ serving as the mapping between our original data and its lower-dimensional representation (see Eq. (1)). In many cases the data we wish to analyze is often rather complex and has a collection of distinct, often unknown, attributes. In this work for example, we deal with datasets of human faces where the variability in the data does not only stem from the difference in the appearance of the subjects, but also from other attributes, such as the pose of the head in relation to the camera, or the facial expression of the subject. The multi-attribute nature of our data calls for a hierarchical framework that is better at representing it than a shallow Semi-NMF.

We therefore propose here the Deep Semi-NMF model, which factorizes a given data matrix X into $m+1$ factors, as follows:

$$X^\pm \approx Z_1^\pm Z_2^\pm \cdots Z_m^\pm H_m^+. \tag{7}$$

This formulation, as shown directly in Eq. (9) with respect to Figs. 1 and 2 allows for a hierarchy of m layers of implicit representations of our data that can be given by the following factorizations:

$$\begin{aligned} H_{m-1}^+ &\approx Z_m^\pm H_m^+ \\ &\vdots \\ H_2^+ &\approx Z_3^\pm \cdots Z_m^\pm H_m^+ \\ H_1^+ &\approx Z_2^\pm \cdots Z_m^\pm H_m^+, \end{aligned} \tag{8}$$

As one can see above, we further restrict these implicit representations $(H_1^+, \ldots, H_{m-1}^+)$ to also be non-negative. By doing so, every layer of this hierarchy of representations also lends itself to a clustering interpretation, which constitutes our method radically different to other multi-layer NMF approaches [22], [23], [24]. By examining Fig. 2, one can better understand the intuition of how that happens. In this case the input to the model, X, is a collection of face images from different subjects (identity), expressing a variety of facial expressions taken from many angles (pose). A Semi-NMF model would find a representation H of X, which would be useful for performing clustering according to the identity of the subjects, and Z the mapping between these identities and the face images. A Deep Semi-NMF model also finds a representation of our data that has a similar interpretation at the top layer, its last factor H_m. However, the mapping from identities to face images is now further analyzed as a product of three factors $Z = Z_1 Z_2 Z_3$, with Z_3 corresponding to the mapping of identities to expressions, $Z_2 Z_3$ corresponding to the mapping of identities to poses, and finally $Z_1 Z_2 Z_3$ corresponding to the mapping of identities to the face images. That means that, as shown in Fig. 2 we are able to decompose our data in 3 different ways according to our three different attributes:

$$\begin{aligned} X^\pm &\approx Z_1^\pm H_1^+ \\ X^\pm &\approx Z_1^\pm Z_2^\pm H_2^+ \\ X^\pm &\approx Z_1^\pm Z_2^\pm Z_3^\pm H_3^+. \end{aligned} \tag{9}$$

More over, due to the non-negativity constrains we enforce on the latent features $H_{(\cdot)}$, it should be noted that this model does not collapse to a Semi-NMF model. Our hypothesis is that by further factorizing Z we are ableAlgorithm construct a deep model that is able to *(1)* automatically learn what this latent hierarchy of attributes is; *(2)* find representations of the data that are most suitable for clustering according to the attribute that corresponds to each layer in the model; and *(3)* find a better high-level, final-layer representation for clustering according to the attribute with the lowest variability, in our case the identity of the face depicted. In our example in Fig. 2 we would expect to find better features for clustering according to identities H_3^+ by learning the hidden representations at each layer most suitable for each of the attributes in our data, in this example: $H_1^+ \approx Z_2^\pm Z_3^\pm H_3^+$ for clustering our original images in terms of poses and $H_2^+ \approx Z_3^\pm H_3^+$ for clustering the face images in terms of expressions.

In order to expedite the approximation of the factors in our model, we pretrain each of the layers to have an initial approximation of the matrices Z_i, H_i as this greatly improves the training time of the model. This is a tactic that has been employed successfully before [16] on deep

autoencoder networks. To perform the pre-training, we first decompose the initial data matrix $\boldsymbol{X} \approx \boldsymbol{Z}_1\boldsymbol{H}_1$, where $\boldsymbol{Z}_1 \in \mathbb{R}^{p\times k_1}$ and $\boldsymbol{H}_1 \in \mathbb{R}_0^{+\,k_1\times n}$. Following this, we decompose the features matrix $\boldsymbol{H}_1 \approx \boldsymbol{Z}_2\boldsymbol{H}_2$, where $\boldsymbol{Z}_2 \in \mathbb{R}^{k_1\times k_2}$ and $\boldsymbol{H}_1 \in \mathbb{R}_0^{+k_2\times n}$, continuing to do so until we have pre-trained all of the layers. Afterwards, we can fine-tune the weights of each layer, by employing alternating minimization (with respect to the objective function in Eq. (10)) of the two factors in each layer, in order to reduce the total reconstruction error of the model, according to the cost function in Eq. (10).

$$\begin{aligned} C_{\text{deep}} &= \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{Z}_1\boldsymbol{Z}_2\cdots\boldsymbol{Z}_m\boldsymbol{H}_m\|_F^2 \\ &= tr[\boldsymbol{X}^\top\boldsymbol{X} - 2\boldsymbol{X}^\top\boldsymbol{Z}_1\boldsymbol{Z}_2\cdots\boldsymbol{Z}_m\boldsymbol{H}_m \\ &\quad + \boldsymbol{H}_m^\top\boldsymbol{Z}_m^\top\boldsymbol{Z}_{m-1}^\top\cdots\boldsymbol{Z}_1^\top\boldsymbol{Z}_1\boldsymbol{Z}_2\cdots\boldsymbol{Z}_m\boldsymbol{H}_m]. \end{aligned} \tag{10}$$

Update rule for the weights matrix $\boldsymbol{Z}$. We fix the rest of the weights for the ith layer and we minimize the cost function with respect to $\boldsymbol{Z}_i$. That is, we set $\frac{\partial C_{\text{deep}}}{\partial \boldsymbol{Z}_i} = 0$, which gives us the updates:

$$\begin{aligned} \boldsymbol{Z}_i &= (\boldsymbol{\Psi}^\top\boldsymbol{\Psi})^{-1}\boldsymbol{\Psi}^\top\boldsymbol{X}\tilde{\boldsymbol{H}}_i^\top(\tilde{\boldsymbol{H}}_i\tilde{\boldsymbol{H}}_i^\top)^{-1} \\ \boldsymbol{Z}_i &= \boldsymbol{\Psi}^\dagger\boldsymbol{X}\tilde{\boldsymbol{H}}_i^\dagger, \end{aligned} \tag{11}$$

where $\boldsymbol{\Psi} = \boldsymbol{Z}_1\cdots\boldsymbol{Z}_{i-1}$, $\dagger$ denotes the Moore-Penrose pseudo-inverse and $\tilde{\boldsymbol{H}}_i$ is the reconstruction of the ith layer's feature matrix.

Update rule for features matrix $\boldsymbol{H}$. Utilizing a similar proof to [8], we can formulate the update rule for $\boldsymbol{H}_i$ which enforces the non-negativity of $\boldsymbol{H}_i$:

$$\boldsymbol{H}_i = \boldsymbol{H}_i \odot \sqrt{\frac{[\boldsymbol{\Psi}^\top\boldsymbol{X}]^{\text{pos}} + [\boldsymbol{\Psi}^\top\boldsymbol{\Psi}]^{\text{neg}}\boldsymbol{H}_i}{[\boldsymbol{\Psi}^\top\boldsymbol{X}]^{\text{neg}} + [\boldsymbol{\Psi}^\top\boldsymbol{\Psi}]^{\text{pos}}\boldsymbol{H}_i}}. \tag{12}$$

Complexity. The computational complexity for the pre-training stage of Deep Semi-NMF is of order $\mathcal{O}(mt(pnk + nk^2 + kp^2 + kn^2))$, where m is the number of layers, t the number of iterations until convergence and k is the maximum number of components out of all the layers. The complexity for the fine-tuning stage is $\mathcal{O}\big(mt_f(pnk + (p+n)k^2)\big)$ where t_f is the number of additional iterations needed.

3.1 Non-Linear Representations

By having a linear decomposition of the initial data distribution we may fail to describe efficiently the non-linearities that exist in between the latent attributes of the model. Introducing non-linear functions between the layers, can enable us to extract features for each of the latent attributes of the model that are non-linearly separable in the initial input space.

This is motivated further from neurophysiology paradigms, as the theoretical and experimental evidence suggests that the human visual system has a hierarchical and rather non-linear approach [27] in processing image structure, in which neurons become selective to process progressively more complex features of the image structure. As argued by Malo et al. [28], employing an adaptive non-linear image representation algorithm results in a reduction of the statistical and the perceptual redundancy amongst the representation elements.

Algorithm 1. Suggested Algorithm for Training a Deep Semi-NMF Model. Initially We Approximate the Factors Greedily Using the Semi-NMF Algorithm [8] and We Fine-Tune the Factors Until We Reach the Convergence Criterion

Input: $\boldsymbol{X} \in \mathbb{R}^{p\times n}$, list of layer sizes
Output: weight matrices $\boldsymbol{Z}_i$ and feature matrices $\boldsymbol{H}_i$ for each of the layers

Initialize Layers
for all layers **do**
  $\boldsymbol{Z}_i, \boldsymbol{H}_i$ SemiNmf($\boldsymbol{H}_{i-1}$, layers(i))
end for
repeat
  for all layers **do**
    $\tilde{\boldsymbol{H}}_i$ $\begin{cases} \boldsymbol{H}_i & \text{if } i = k \\ \boldsymbol{Z}_{i+1}\tilde{\boldsymbol{H}}_{i+1} & \text{otherwise} \end{cases}$
    $\boldsymbol{\Psi} \leftarrow \prod_{k=1}^{i-1}\boldsymbol{Z}_k$
    $\boldsymbol{Z}_i$ $\boldsymbol{\Psi}^\dagger\boldsymbol{X}\tilde{\boldsymbol{H}}_i^\dagger$
    $\boldsymbol{H}_i$ $\boldsymbol{H}_i \odot \left[\frac{[\boldsymbol{\Psi}^\top\boldsymbol{X}]^{\text{pos}} + [\boldsymbol{\Psi}^\top\boldsymbol{\Psi}]^{\text{neg}}\boldsymbol{H}_i}{[\boldsymbol{\Psi}^\top\boldsymbol{X}]^{\text{neg}} + [\boldsymbol{\Psi}^\top\boldsymbol{\Psi}]^{\text{pos}}\boldsymbol{H}_i}\right]^\eta$
  end for
until Stopping criterion is reached

From a mathematical point of view, one can use a non-linear function $g(\cdot)$, between each of the implicit representations $(\boldsymbol{H}_1^+, \ldots, \boldsymbol{H}_{m-1}^+)$, in order to better approximate the non-linear manifolds which the given data matrix $\boldsymbol{X}$ originally lies on. In other words by using a non-linear squashing function we enhance the expressibility of our model and allow for a better reconstruction of the initial data. This has been proved in [29] by the use of the Stone-Weierstrass theorem, in the case of multilayer feedforward network structures, which Semi-NMF is an instance of, that arbitrary squashing functions can approximate virtually any function of interest to any desired degree of accuracy, provided sufficiently many hidden units are available.

To introduce non-linearities in our model we modify the ith feature matrix $\boldsymbol{H}_i$, by setting

$$\boldsymbol{H}_i \approx g(\boldsymbol{Z}_{i+1}\boldsymbol{H}_{i+1}). \tag{13}$$

which in turns changes the objective function of the model to be:

$$C^* = \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{Z}_1 g(\boldsymbol{Z}_2\, g(\cdots g(\boldsymbol{Z}_m\boldsymbol{H}_m)))\|_F^2. \tag{14}$$

In order to compute the derivative for the ith feature layer, we make use of the chain rule and get:

$$\begin{aligned} \frac{\partial \boldsymbol{C}^*}{\partial \boldsymbol{H}_i} &= \boldsymbol{Z}_i^\top\frac{\partial \boldsymbol{C}^*}{\partial \boldsymbol{Z}_i\boldsymbol{H}_i} \\ &= \boldsymbol{Z}_i^\top\left[\frac{\partial \boldsymbol{C}^*}{\partial g(\boldsymbol{Z}_i\boldsymbol{H}_i)} \odot \nabla g(\boldsymbol{Z}_i\boldsymbol{H}_i)\right] \\ &= \boldsymbol{Z}_i^\top\left[\frac{\partial \boldsymbol{C}^*}{\partial \boldsymbol{H}_{i-1}} \odot \nabla g(\boldsymbol{Z}_i\boldsymbol{H}_i)\right]. \end{aligned}$$

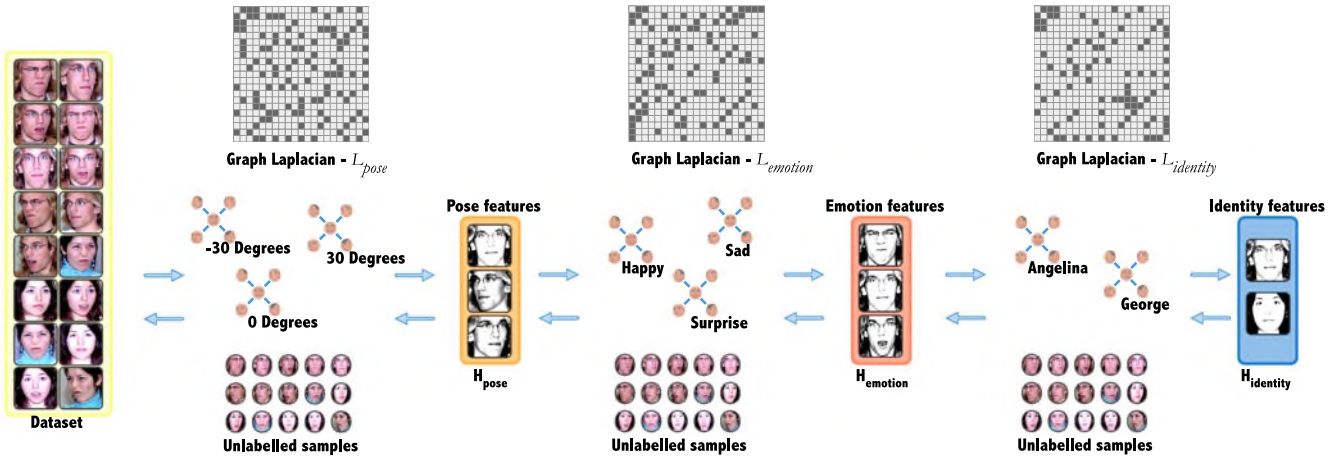


Fig. 3. A weakly-supervised Deep Semi-NMF model uses prior knowledge we have about the attributes of our model to improve the final representation of our data. In this illustration we incorporate information from pose, expression, and identity attributes into the three feature layers of our model H_{pose}, $H_{\text{expression}}$, and H_{identity} respectively.

The derivation of the first feature layer H_1 is then identical to the version of the model with one layer

$$\begin{aligned}\frac{\partial C^*}{\partial H_1} &= \frac{1}{2}\frac{\partial Tr[-2X^\top Z_1 H_1 + (Z_1 H_1)^\top Z_1 H_1]}{\partial H_1} \\ &= Z_1^\top Z_1 H_1 - Z_1^\top X \\ &= Z_1^\top (Z_1 H_1 - X).\end{aligned}$$

Similarly we can compute the derivative for the weight matrices Z_i,

$$\begin{aligned}\frac{\partial C^*}{\partial Z_i} &= \frac{\partial C^*}{\partial Z_i H_i} H_i^\top \\ &= \left[\frac{\partial C^*}{\partial g(Z_i H_i)} \odot \nabla g(Z_i H_i)\right] H_i^\top \\ &= \left[\frac{\partial C^*}{\partial H_{i-1}} \odot \nabla g(Z_i H_i)\right] H_i^\top,\end{aligned}$$

and

$$\begin{aligned}\frac{\partial C^*}{\partial Z_1} &= \frac{1}{2}\frac{\partial Tr[-2X^\top Z_1 \tilde{H}_1 + (Z_1 \tilde{H}_1)^\top Z_1 \tilde{H}_1]}{\partial Z_1} \\ &= \left(Z_1 \tilde{H}_1 - X\right)\tilde{H}_1^\top.\end{aligned}$$

Using these derivatives we can make use of gradient descent optimizations such as Nesterov's optimal gradient [30], to minimize the cost function with respect to each of the weights of our model.

3.2 Stochastic Optimisation

In recent years we have witnessed an exponential growth in data both in variety but also volume. Unfortunately it is computationally intractable to take advantage of this sheer amount of data as (semi) non-negative factorisation algorithms scale quadratically in time with respect to the number of observations n (cf. Section 3) but also require the whole training set to reside in main memory. Fortunately, stochastic optimisation techniques [31], [32] combat both of these issues by processing only a small portion of the dataset on every iteration, known as a minibatch. For simplicity of notation and without loss of generality we assume that n is divisible by the number of mini-batches q and in Eq. (15) we formulate the stochastic version of Deep Semi-NMF cost function

$$\tilde{C} = \frac{1}{2}\sum_{i=0}^{q-1}\left\|X^{[i]} - Z_1 g\Big(Z_2\, g\Big(\cdots g\Big(Z_m H_m^{[i]}\Big)\Big)\Big)\right\|_F^2, \qquad (15)$$

subject to $\forall i.\ H_i \geq 0$, and where $X^{[i]}$ is the subset of the training set (minibatch) containing $b = \frac{n}{q}$ examples. To compute the parameter updates for all the parameters of our model we can use stochastic optimisation techniques such as SGD [31] or Adam [32]. Although this is an approximation of the objective function over the whole training set Eq. (4) we have found that in practise this works well even for small batch sizes (32 samples).

4 Weakly-Supervised Attribute Learning

As before, consider a dataset of faces X as in Fig. 2. In this dataset, we have a collection of subjects, where each one has a number of images expressing different expressions, taken by different angles (pose information). A three layer Deep Semi-NMF model could be used here to automatically learn representations in an unsupervised manner (H_{pose}, $H_{\text{expression}}$, H_{identity}) that conform to this latent hierarchy of attributes. Of course, the features are extracted without accounting (partially) available information that may exist for each of the these attributes of the dataset.

To this effect we propose a Deep Semi-NMF approach that can incorporate partial attribute information that we named *Weakly-Supervised Deep Semi-Nonnegative Matrix Factorization* (Deep WSF). Deep WSF is able to learn, in a semi-supervised manner, a hierarchy of representations; each level of this hierarchy corresponding to a specific attribute for which we may have only partial labels for. As depicted in Fig. 3, we show that by incorporating some label information via graph regularization techniques we are able to do better than the Deep Semi-NMF for classifying faces according to pose, expression, and identity. We also show that by initializing a Deep Semi-NMF with the weights learned by a Deep WSF we are able to improve the performance of the Deep Semi-NMF for the task of clustering faces according to identity.

4.1 Incorporating Known Attribute Information

Consider that we have an undirected graph G with N nodes, where each of the nodes corresponds to one data point in our initial dataset. A node i is connected to another node j iff we have a priori knowledge that those samples share the same label, and this edge has a weight $\boldsymbol{w}_{ij}$.

In the simplest case scenario, we use a binary weight matrix $\boldsymbol{W}$ defined as:

$$\boldsymbol{W}_{ij} = \begin{cases} 1 & \text{if } y_i = y_j \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Instead one can also choose a *radial basis function kernel*

$$\boldsymbol{W}_{ij} = \begin{cases} \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right) & \text{if } y_i = y_j \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

or a *dot-product weighting*, where

$$\boldsymbol{W}_{ij} = \begin{cases} \boldsymbol{x}_i^\top \boldsymbol{x}_j & \text{if } y_i = y_j \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

Using the graph weight matrix $\boldsymbol{W}$, we formulate $\boldsymbol{L}$, which denotes the Graph Laplacian [33] that stores our prior knowledge about the relationship of our samples and is defined as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{D}$ is a diagonal matrix whose entries are column (or row, since $\boldsymbol{W}$ is symmetric) sums of $\boldsymbol{W}$, $\boldsymbol{D}_{jj} = \sum_k \boldsymbol{W}_{jk}$. In order to control the amount of embedded information in the graph we introduce as in [34], [35], [36], a term $\mathcal{R}$ which controls the smoothness of the low dimensional representation

$$\begin{aligned} \mathcal{R} &= \sum_{j,l=1}^{N} \|\boldsymbol{h}_j - \boldsymbol{h}_l\|^2 \boldsymbol{W}_{jl} \\ &= \sum_{j=1}^{N} \boldsymbol{h}_j^\top \boldsymbol{h}_j \boldsymbol{D}_{jj} - \sum_{j,l=1}^{N} \boldsymbol{h}_j^\top \boldsymbol{h}_j \boldsymbol{W}_{jl} \\ &= \mathrm{Tr}\big(\boldsymbol{H}^\top \boldsymbol{D} \boldsymbol{H}\big) - \mathrm{Tr}\big(\boldsymbol{H}^\top \boldsymbol{W} \boldsymbol{H}\big) \\ &= \mathrm{Tr}\big(\boldsymbol{H}^\top \boldsymbol{L} \boldsymbol{H}\big), \end{aligned} \tag{19}$$

where h_i is the low-dimensional features for sample i, that we obtain from the decomposed model.

Minimizing this term $\mathcal{R}$, we ensure that the euclidean difference between the final level representations of any two data points $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ is low when we have prior knowledge that those samples have a relationship, producing similar features $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$. On the other hand, when we do not have any expert information about some or even all the class information about an attribute, the term has no influence on the rest of the optimization.

Before deriving the update rules and the algorithm for the multi-layer Deep WSF model, we first show the simpler case of the one layer version, which will come into use for pre-training the model, as Semi-NMF can be used to pre-train the purely unsupervised Deep Semi-NMF. We call this model *Weakly Supervised Semi-NMF* WSF .

By combining the term $\mathcal{R}$ introduced in Eq. (19), with the cost function of Semi-NMF we obtain the cost function for Weakly-Supervised Factorization (WSF)

$$\begin{aligned} C_{\mathrm{WSF}} = \|\boldsymbol{X} - \boldsymbol{Z}^{\pm}\boldsymbol{H}^{+}\|_F^2 + \lambda \mathrm{Tr}(\boldsymbol{H}^\top \boldsymbol{L} \boldsymbol{H}) \\ \text{s.t. } \boldsymbol{H} \geq 0. \end{aligned} \tag{20}$$

The update rules, but also the algorithm for training a WSF model can be found in the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2016.2554555.

We incorporate the available partial labelled information for the pose, expression, and identity by forming a graph Laplacian for pose for the first layer ($\boldsymbol{L}_{\mathrm{pose}}$), expression for the second layer ($\boldsymbol{L}_{\mathrm{expression}}$), and identity for the third layer ($\boldsymbol{L}_{\mathrm{identity}}$) of the model. We can then tune the regularization parameters λ_i accordingly for each of the layers to express the importance of each of these parameters to the Deep WSF model.[3] Using the modified version of our objective function Eq. (21), we can derive the Algorithm 2.

$$\begin{aligned} C_{\mathrm{DeepWSF}} = &\frac{1}{2}\|\boldsymbol{X} - \boldsymbol{Z}_1 g(\ldots g(\boldsymbol{Z}_m \boldsymbol{H}_m))\|_F^2 \\ &+ \frac{1}{2}\sum_{i=1}^{m} \lambda_i \mathrm{Tr}(\boldsymbol{H}_i^\top \boldsymbol{L}_i \boldsymbol{H}_i). \end{aligned} \tag{21}$$

In order to compute the derivative for the ith feature layer, we make use of the chain rule and get:

$$\begin{aligned} \frac{\partial \mathcal{C}_{\mathrm{dwsf}}}{\partial \boldsymbol{H}_i} &= \boldsymbol{Z}_i^\top \frac{\partial \mathcal{C}_{\mathrm{deep}}}{\partial \boldsymbol{Z}_i \boldsymbol{H}_i} + \frac{1}{2}\frac{\lambda_i \mathrm{Tr}(\boldsymbol{H}_i^\top \boldsymbol{L}_i \boldsymbol{H}_i)}{\partial \boldsymbol{H}_i} \\ &= \boldsymbol{Z}_i^\top \left[\frac{\partial \mathcal{C}_{\mathrm{deep}}}{\partial \boldsymbol{H}_{i-1}} \odot \nabla g(\boldsymbol{Z}_i \boldsymbol{H}_i)\right] \\ &\quad + \lambda_i \boldsymbol{L}_i \boldsymbol{H}_i, \end{aligned}$$

and the derivation of the first feature layer $\boldsymbol{H}_1$ is then:

$$\begin{aligned} \frac{\partial \mathcal{C}_{\mathrm{dwsf}}}{\partial \boldsymbol{H}_1} &= \frac{\partial \mathcal{C}_{\mathrm{deep}}}{\partial \boldsymbol{Z}_i \boldsymbol{H}_i} + \frac{1}{2}\frac{\lambda_1 \mathrm{Tr}(\boldsymbol{H}_1^\top \boldsymbol{L}_1 \boldsymbol{H}_1)}{\partial \boldsymbol{H}_1} \\ &= \boldsymbol{Z}_1^\top (\boldsymbol{Z}_1 \boldsymbol{H}_1 - \boldsymbol{X}) + \lambda_1 \boldsymbol{L}_1 \boldsymbol{H}_1. \end{aligned}$$

Similarly we can compute the derivative for the weight matrices $\boldsymbol{Z}_i$,

$$\begin{aligned} \frac{\partial \mathcal{C}_{\mathrm{dwsf}}}{\partial \boldsymbol{Z}_i} &= \frac{\partial \mathcal{C}_{\mathrm{deep}}}{\partial \boldsymbol{Z}_i \boldsymbol{H}_i} \boldsymbol{H}_i^\top \\ &= \left[\frac{\partial \mathcal{C}_{\mathrm{deep}}}{\partial \boldsymbol{H}_{i-1}} \odot \nabla g(\boldsymbol{Z}_i \boldsymbol{H}_i)\right] \boldsymbol{H}_i^\top, \end{aligned}$$

and

$$\begin{aligned} \frac{\partial \mathcal{C}_{\mathrm{dwsf}}}{\partial \boldsymbol{Z}_1} &= \frac{\partial \mathcal{C}_{\mathrm{deep}}}{\partial \boldsymbol{Z}_1} \\ &= \big(\boldsymbol{Z}_1 \tilde{\boldsymbol{H}}_1 - \boldsymbol{X}\big) \tilde{\boldsymbol{H}}_1^\top. \end{aligned}$$

3. In general, we do believe that the knowledge that each layer captures depends on the gradually refined level of variability. Of course, knowing the correspondence of the attributes and the layers is problem, as well as database dependent. An empirical indicator to choose which layer is optimal for each attribute is the clustering accuracy (e.g., select the attribute of the dataset which yields the highest degree of mutual information for each of the layers). We have chosen the proposed setting because it was intuitive, as well as because it was supported by empirical evidence (supplementary material, available online).

Using these derivatives we can make use of gradient descent optimizations as with the non-linear Deep Semi-NMF model, to minimize the cost function with respect to each of the factors of our model. If instead use the linear version of the algorithm where g is the identity function, then we can derive a multiplicative update algorithm version of Deep WSF, as described in Algorithm 2.

Algorithm 2. Proposed Algorithm for Training a Deep WSF Model. Initially We Approximate the Factors Greedily Using WSF or Semi-NMF and We Fine-Tune the Factors Until We Reach the Convergence Criterion

Input: $\boldsymbol{X} \in \mathbb{R}^{p \times n}$, list of layer sizes *layers*
Output: weight matrices $\boldsymbol{Z}_i$ and feature matrices $\boldsymbol{H}_i$ for each of the layers
Initialize Layers
for all layers **do**
  $\boldsymbol{Z}_i, \boldsymbol{H}_i$ WSF($\boldsymbol{H}_{i-1}$, layers(i), λ_i)
end for
repeat
  for all layers **do**
    $\tilde{\boldsymbol{H}}_i \quad \begin{cases} \boldsymbol{H}_i & \text{if} \quad i = k \\ \boldsymbol{Z}_{i+1}\tilde{\boldsymbol{H}}_{i+1} & \text{otherwise} \end{cases}$
    $\boldsymbol{\Psi} \leftarrow \prod_{k=1}^{i-1} \boldsymbol{Z}_k$
    $\boldsymbol{Z}_i \quad \boldsymbol{\Psi}^{\dagger}\boldsymbol{X}\tilde{\boldsymbol{H}}_i^{\dagger}$
    $\boldsymbol{F} \leftarrow \frac{[\boldsymbol{\Psi}^{\top}\boldsymbol{X}]^{\text{pos}} + [\boldsymbol{\Psi}^{\top}\boldsymbol{\Psi}]^{\text{neg}}\boldsymbol{H}_i + \lambda_i \boldsymbol{H}_i \boldsymbol{W}_i}{[\boldsymbol{\Psi}^{\top}\boldsymbol{X}]^{\text{neg}} + [\boldsymbol{\Psi}^{\top}\boldsymbol{\Psi}]^{\text{pos}}\boldsymbol{H}_i + \lambda_i \boldsymbol{H}_i \boldsymbol{D}_i}$
    $\boldsymbol{H}_i \quad \boldsymbol{H}_i \odot \boldsymbol{F}^{\eta}$
  end for
until Stopping criterion is reached

4.2 Weakly Supervised Factorization with Multiple Label Constraints

Another approach we propose within this framework is a single–layer WSF model that learns only a single representation based on information from multiple attributes. This Multiple-Attribute extension of the WSF, the WSF-MA, accounts for the case of having multiple number of attributes ξ for our data matrix $\boldsymbol{X}$, by having a regularization term $\lambda_i \text{Tr}(\boldsymbol{H}\boldsymbol{L}_i\boldsymbol{H}^{\top})$. This term uses the prior information from all the available attributes to construct ξ Laplacian graphs where each of them has a different regularization factor λ_i.

This constitutes WSF-MA, whose cost function is

$$C_{\text{mawsf}} = \|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{H}\|_F^2 + \sum_{i=1}^{\xi} \lambda_i \text{Tr}(\boldsymbol{H}^{\top}\boldsymbol{L}_i\boldsymbol{H}) \quad \text{s.t. } \boldsymbol{H} \geq 0. \tag{22}$$

The update rules used, and the algorithm can be found in the supplementary material, available online.

5 Out-of-Sample Projection

After learning an internal model of the data, either using the purely unsupervised Deep Semi-NMF or to perform semi-supervised learning using the Deep WSF model with learned weights $\boldsymbol{Z}$, and features $\boldsymbol{H}$ we can project an out-of-sample data point $\boldsymbol{x}^*$ to the new lower-dimensional embedding $\boldsymbol{h}^*$. We can accomplish this using one of the two presented methods.

Method 1: Basis matrix reconstruction.

Each testing sample $\boldsymbol{x}^*$ is projected into the linear space defined by the weights matrix $\boldsymbol{Z}$. Although this method has been used by various previous works [37], [38] using the NMF model, it does *not* guarantee the non-negativity of $\boldsymbol{h}^*$.

For the linear case of Deep WSF, this would lead to

$$\boldsymbol{h}^* \approx [\boldsymbol{Z}_1\boldsymbol{Z}_2 \dots \boldsymbol{Z}_l]^{\dagger}\boldsymbol{x}^*. \tag{23}$$

and for the non-linear case

$$\boldsymbol{h}^* \approx g^{-1}\Big(\boldsymbol{Z}_l^{\dagger}\Big(\cdots\Big(\boldsymbol{Z}_2^{\dagger}g^{-1}\Big(\boldsymbol{Z}_1^{\dagger}\boldsymbol{x}^*\Big)\Big)\Big)\Big). \tag{24}$$

Method 2: Using non-negativity update rules.

Using the same process as in Deep Semi-NMF, we can intuitively learn the new features $\boldsymbol{h}^*$, by assuming that the weight matrices $\forall i.\boldsymbol{Z}_i$ remain fixed.

$$\forall l.\boldsymbol{h}_l^* = \text{argmin}_{\boldsymbol{h}}\|\boldsymbol{x}^* - \prod_{i=1}^{l}\boldsymbol{Z}_i\boldsymbol{h}_l\| \quad \text{such that } \boldsymbol{h}_l \geq 0. \tag{25}$$

and for the non-linear case

$$\forall l.\boldsymbol{h}_l^* = \text{argmin}_{\boldsymbol{h}}\|\boldsymbol{x}^* - \boldsymbol{Z}_1 g(\boldsymbol{Z}_2 \cdots g(\boldsymbol{Z}_l\boldsymbol{h}_l))\| \quad \text{such that } \boldsymbol{h}_l \geq 0. \tag{26}$$

where $\boldsymbol{h}_l$, corresponds to the lth feature layer for the out-of-sample data point $\boldsymbol{x}^*$. This problem is then solved by using Algorithm 1 as Deep Semi-NMF, but without updating the weight matrices $\boldsymbol{Z}_i$.

6 Experiments

In order to assess the performance of the proposed unsupervised, as well as semi-supervised methodologies and compare with state-of-the-art we have conducted clustering, as well as classification experiments.

For clustering we have used the following datasets:

- *CMU PIE:* We used a freely available version of CMU Pie [39], which comprises of 2,856 grayscale 32×32 face images of 68 subjects. Each person has 42 facial images under different illumination conditions. In this database we only know the identity of the face in each image.
- *XM2VTS: The Extended Multi Modal Verification for Teleservices and Security applications* (XM2VTS) [40] contains 2,360 frontal images of 295 different subjects. Each subject has two available images for each of the four different laboratory sessions, for a total of eight images. The images were eye-aligned and resized to 42×30.
- *CASIA WebFace:* This is the largest and most challenging of the three datasets. CASIA WebFace [41] is comprised of 494,414 images of 10,575 people in-the-wild conditions. We use the aligned and rescaled version of the dataset as is provided by the creators (100×100 images). As the large size of the dataset makes it hard to do extensive experimental analysis on it, we use a

subset of 10,000 images of 500 subjects (20 images/subject). Some emperical clustering results are shown in supplementary material, available online.

In order to evaluate the performance of our Deep Semi-NMF model, we compared it against not only Semi-NMF [8], but also against other NMF variants that could be useful in learning such representations. By using only the pixel intensities of the images in each of our datasets, which of course give us a strictly non-negative input data matrix X, we compare the reconstruction error and the clustering performance of our Deep Semi-NMF method against the Semi-NMF, NMF with multiplicative update rules [19], Multi-Layer NMF [24], GNMF [20], and NeNMF [42].

In Section 6.4, having demonstrated the effectiveness of the purely unsupervised Deep Semi-NMF model we show next how pretraining a Deep WSF model on an auxiliary dataset and using the learned weights to perform unsupervised Deep Semi-NMF can lead to significant improvements in terms of the clustering accuracy.

Classification experiments are reported in two datasets:

- *CMU Multi-PIE:* CMU Multi-PIE [43] contains around $750{,}000$ images of 337 subjects, captured under laboratory conditions in four different sessions. In this work, we used a subset of $7{,}905$ images of 147 subjects in 5 different poses and expressing six different emotions, which is the amount of samples that we had annotations and were imposed to the same illumination conditions. In this experiment (in Section 6.5) we examine the classification abilities of the proposed models for each of the three attributes of the CMU Multi-PIE dataset (pose/expression/identity) and use this to test more on our secondary hypothesis, i.e., that every representation in each layer is in fact most suited for learning according to the attributes that corresponds to the layer of interest.

6.1 Implementation Details

To initiate the matrix factorization process, NMF and Semi-NMF algorithms start from some initial point (Z^0, H^0), where usually Z^0 and H^0 are randomly initialized matrices.

A problem with this approach, is not only the initialization point is far from the final convergence point, but also makes the optimization non deterministic.

The proposed initialization of Semi-NMF by its authors is instead by using the k-means algorithm [44]. Nonetheless, k-means is computationally heavy when the number of components k is fairly high ($k > 100$). As an alternative we implemented the approach by [45] which suggests exact and heuristic algorithms which solve Semi-NMF decompositions using an SVD based initialization. We have found that using this method for Semi-NMF, Deep Semi-NMF, and WSF helps the algorithms to converge a lot sooner.

Similarly, to speed up the convergence rate of NMF we use the Non-negative Double Singular Value decomposition (NNDSVD) suggested by Boutsidis et al. [46]. NNDSVD is a method based on two SVD processes, one to approximate the initial data matrix X and the other to approximate the positive sections of the resulting partial SVD factors.

For the GNMF experimental setup, we chose a suitable number of neighbours to create the regularizing graph, by visualizing our datasets using Laplacian Eigenmaps [47], such that we had visually distinct clusters (in our case 5).

TABLE 1
The Reconstruction Error ($\|X - \tilde{X}\|_F^2$) for Each of the Algorithms on the CMU PIE Dataset, for a Variable Number of Components

	# Components					
Model	**20**	**30**	**40**	**50**	**60**	**70**
Deep Semi-NMF	9.18	7.61	6.50	5.67	4.99	4.39
GNMF	10.56	9.35	8.73	8.18	7.81	7.48
Multi-layer NMF	11.11	10.16	9.28	8.49	7.63	6.98
NMF (MUL)	10.53	9.36	8.51	7.91	7.42	7.00
NeNMF	9.83	8.39	7.39	6.60	5.94	5.36
Semi-NMF	9.14	7.57	6.43	5.53	4.76	4.13

6.2 Reconstruction Error Results

Our first experiment was to evaluate whether the extra layers, which naturally introduce more factors and are therefore more difficult to optimize, result in a lower quality local optimum. We evaluated how well the matrix decomposition is performed by calculating the reconstruction error, the Frobenius norm of the difference between the original data and the reconstruction for all the methods we compared Table 1. Note that, in order to have comparable results, all of the methods have the same stopping criterion rules. We have set the maximum amount of iterations to 1,000 (usually $\sim$100 iterations are enough) and we use the convergence rule $E_{i-1} - E_i \leq \kappa \max(1, E_{i-1})$ in order to stop the process when the reconstruction error (E_i) between the current and previous update is small enough. In our experiments we set $\kappa = 10^{-6}$. Section 6.2 shows the change in reconstruction error with respect to the selected number of features in H_2 for all the methods we used on the Multi-PIE dataset.

The results show that Semi-NMF manages to reach a much lower reconstruction error than the other methods consistently, which would match our expectations as it does not constrain the weights Z to be non-negative. What is important to note here is that the Deep Semi-NMF models do not have a significantly lower reconstruction error compared to the equivalent Semi-NMF models, even though the approximation involves more factors. Multi-layer NMF and GNMF have a larger reconstruction error, in return for uncovering more meaningful features than their NMF counterpart.

6.3 Clustering Results

After achieving satisfactory reconstruction error for our method, we proceeded to evaluate the features learned at the final representation layer, by using k-means clustering, as in [20]. To assess the clustering quality of the representations produced by each of the algorithms we compared, we take advantage of the fact that the datasets are already labelled.

As learning is unsupervised, we do not have a correspondence between the clusters found by k-means and the ground truth annotations, making the evaluation of the results not completely trivial. Two popular metrics that are used in literature to assess the clustering results are the clustering accuracy (AC) and the normalized mutual

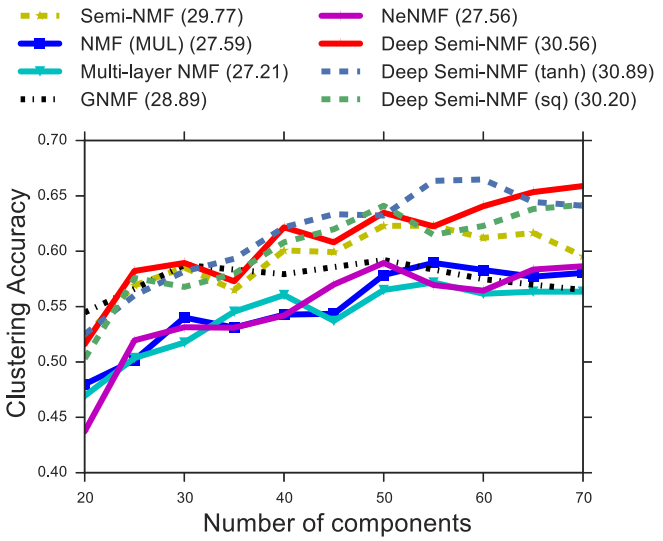


Fig. 4. *XM2VTS-Pixel Intensities:* Accuracy for clustering based on the representations learned by each model with respect to identities. The deep architectures are comprised of two representation layers (1260-625-a) and the representations used were from the top layer. In parentheses we show the AUC scores.

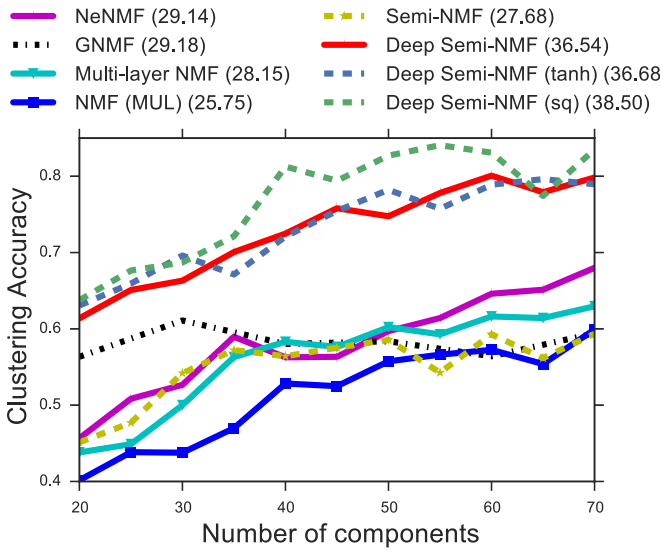


Fig. 6. *CASIA WebFace–Pixel Intensities:* Accuracy for clustering based on the representations learned by each model with respect to identities. The deep architectures are comprised of two representation layers (10000-625-a) and the representations used were from the top layer. In parentheses we show the AUC scores.

information (NMI) metric [20], [48]. Given a set of predictions $\hat{\boldsymbol{y}}$ and ground truth annotations $\boldsymbol{y}$, the clustering accuracy is defined to be

$$\mathrm{AC}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{\sum_{i=1}^{N} \delta(y_i, \mathrm{map}(\hat{y}_i))}{N},$$

where map is a permutation function that maps each predicted cluster id to each corresponding ground truth. The problem is formulated as a weighted bipartite matching problem and is solved using the Kuhn-Munkres algorithm [49].

By employing information theory to measure the agreement between the two clustering partitions one arrives at the Mutual Information (MI) score. MI quantifies the amount of information between the two random variables and is defined as

$$\mathrm{MI}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_{y_i \in \boldsymbol{y}, \hat{y}_i \in \hat{\boldsymbol{y}}} p(y_i, \hat{y}_i) \log \left(\frac{p(y_i, \hat{y}_i)}{p(y_i)\, p(\hat{y}_i)} \right),$$

where $p(y_i)$ and $p(\hat{y}_i)$ are the probabilities that an image of a face selected from the dataset belongs to clusters y_i and $\hat{y}_i$ respectively, and $p(y_i, \hat{y}_i)$ is the joint probability that an arbitrary selected image belongs to clusters y_i and $\hat{y}_i$ at the same time. To force the score to have an explicit upper bound we use the Normalized Mutual Information (NMI) score which is then defined to be

$$\mathrm{NMI}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{\mathrm{MI}(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\sqrt{H(\boldsymbol{y}) H(\hat{\boldsymbol{y}})}}.$$

It is easy to check that NMI score ranges from 0 to 1. In the case that the two clusters are identical then it will be exactly to 1, whereas if two clusterings are independent then it will be equal to 0. The AUC score which we report for each method in our results, is simply the area under each curve approximated using the trapezoid rule.

For a cleaner presentation we have included all the experiments that use NMI in the supplement, available online.

We made use of two main non-linearities for our experiments, the scaled hyperbolic tangent $stanh(x) = \alpha \tanh(\beta x)$ with $\alpha = 1.7159, \beta = \frac{2}{3}$ [50], and a square auxiliary function $sq(x) = x^2$.

Figs. 4, 5, and 6 show the comparison in clustering accuracy when using k-means on the feature representations produced by each of the techniques we compared, when our input matrix contained only the pixel intensities of each image. Our method significantly outperforms every method we compared it with on all the datasets, in terms of clustering accuracy.

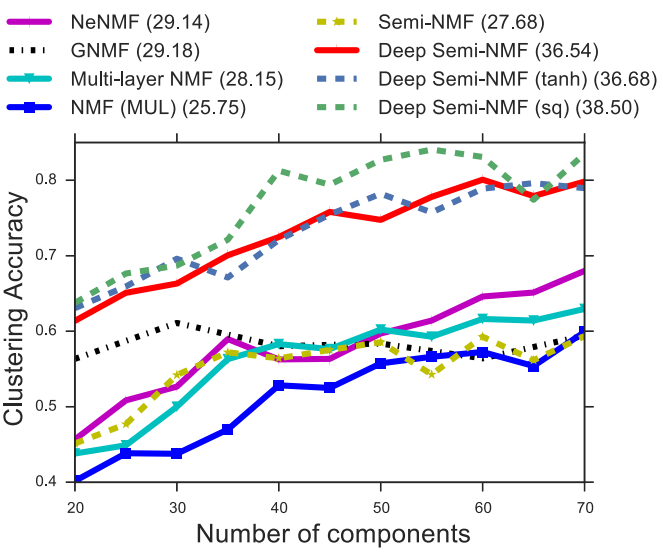


Fig. 5. *CMU PIE–Pixel Intensities:* Accuracy for clustering based on the representations learned by each model with respect to identities. The deep architectures are comprised of two representation layers (1024-625-a) and the representations used were from the top layer. In parenthesese we show the AUC scores.

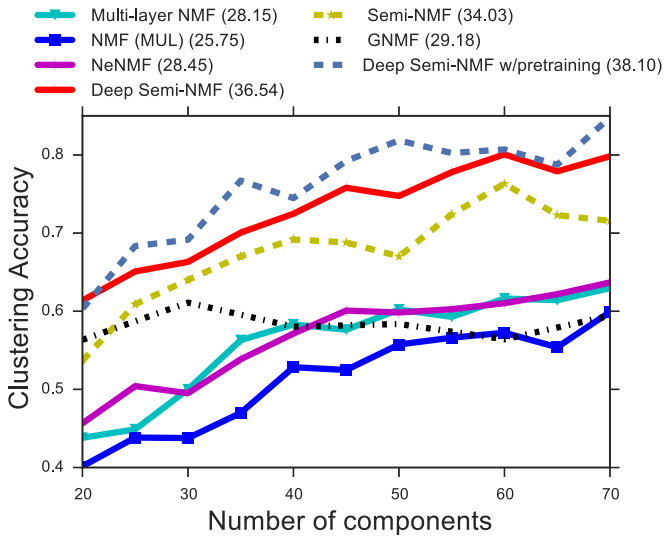


Fig. 7. *Supervised pre-training:* Clustering accuracy on the CMU PIE dataset, after supervised training on the XM2VTS dataset using a priori Deep Semi-NMF. In parentheses we show the AUC scores.

6.4 Supervised Pre-Training

As the optimization process of deep architectures is highly non-convex, the initialization point of the process is an important factor for obtaining good final representation for the initial dataset. Following trends in deep learning [51], we show that supervised pretraining of our model on a auxiliary dataset and using the learned weights as initialization points for the unsupervised Deep Semi-NMF algorithm can lead to significant performance improvements in regards to clustering accuracy.

As an auxiliary dataset we use XM2VTS where we resize all the images to a 32×32 resolution to match the CMU PIE image resolution, which is our primary dataset. Splitting the XM2VTS dataset to training/validation sets, we learn weights $Z_{1,2}^{\text{xm2vts}}$ using a Deep WSF model with $(625\text{–}a)$ layers, and regularization parameters $\lambda = \{0, 0.01\}$.

We then use the obtained weights $Z_{1,2}^{\text{xm2vts}}$ from the supervised task as an initialization point and perform unsupervised fine-tuning on the CMU PIE dataset. To evaluate the resulting features, we once again perform clustering using the k-means algorithm.

In our experiments all the models with supervised pretraining outperformed the ones without, as shown in Fig. 7, in terms of clustering accuracy. Additionally this validates our claim of how pretraining can be exploited to get better representations out of unlabelled data.

6.5 Learning with Respect to Different Attributes

Finally, we conducted experiments for classification using each of the three representations learned by our three-layered Deep WSF models when the input was the raw pixel intensities of our images of a larger subset of the CMU Multi-PIE dataset. Using the annotations from Sagonas et al. [52], [53], we aligned these images based on a common frame. After that, we resized them to a smaller resolution of 40×30. The database comes with labels for each of the attributes mentioned above: identity, illumination, pose, expression. We only used CMU Multi-PIE for this experiment since we only had identity labels for our other datasets. We split this subset into a training and validation set of 2,025 images, and the rest for testing.

TABLE 2
The Performance in Classification Accuracy on the CMU Multi-PIE Dataset Using an SVM Classifier on Top of the Learned Features

	Model	Pose	Expression	Identity
Unsupervised	Semi-NMF	99.73	81.50	36.46
	NMF	100.00	80.68	49.12
	Deep Semi-NMF	99.86	80.54	61.22
Semi	CNMF	89.21	33.88	28.30
	DNMF	100.00	82.22	55.78
Proposed	WSF	100.00	81.50	63.81
	WSF-MA	100.00	81.50	64.08
	Deep WSF	**100.00**	**82.90**	**65.17**

For the multi-layer models we used three layers corresponding to each of the attributes, and performed classification using the features learned for the corresponding attribute. For the one-layer models, we learned three distinct models.

We compare the classification performance of an SVM classifier (with a penalty parameter $\gamma = 1$) using the data representations of the NMF, Semi-NMF, and Deep Semi-NMF models that have no attribute information. The CNMF [25], DNMF [26], and our WSF models that have attribute labels only for the attribute we were classifying for, and our WSF-MA and Deep WSF that learned data representations based on all attribute information available. In Table 2, we demonstrate the performance in accuracy of each of the methods. In all of the methods, each feature layer has 100 components, and in the case of the Deep WSF model, we have used $\forall i.\lambda_i = 10^{-3}$. Detailed results for Deep WSF.

We also compared the performance of our Deep WSF with that of WSF and WSF-MA to see whether the different levels of representation amount to better performance in classification tasks for each of the attributes represented. In both cases, but also in comparison with the rest state-of-the-art unsupervised and semi-supervised matrix factorization techniques, our proposed solution manages to extract better features for the task at hand as seen in Table 2 for classification are also shown in Fig. 8.

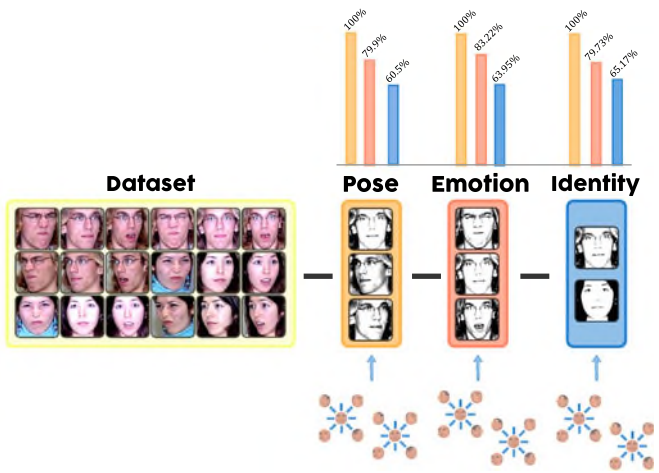


Fig. 8. A three layer Deep WSF model trained on CMU MultiPIE with only frontal illumination (camera 5). The bars depict the accuracy levels for the pose (●), emotion (●), and identity (●) respectively, for each layer, with a linear SVM classifier.

7 Conclusion

We have introduced a novel deep architecture for semi-nonnegative matrix factorization, the Deep Semi-NMF, that is able to automatically learn a hierarchy of attributes of a given dataset, as well as representations suited for clustering according to these attributes. Furthermore we have presented an algorithm for optimizing the factors of our Deep Semi-NMF, and we evaluate its performance compared to the single-layered Semi-NMF and other related work, on the problem of clustering faces with respect to their identities. We have shown that our technique is able to learn a high-level, final-layer representation for clustering with respect to the attribute with the lowest variability in the case of two popular datasets of face images, outperforming the considered range of typical powerful NMF-based techniques.

We further proposed Deep WSF , which incorporates knowledge from the known attributes of a dataset that might be available. Deep WSF can be used for datasets that have (partially) annotated attributes or even are a combination of different data sources with each one providing different attribute information. We have demonstrated the abilities of this model on the CMU Multi-PIE dataset, where using additional information provided to us during training about the pose, emotion, and identity information of the subject we were able to uncover better features for each of the attributes, by having the model learning from all the available attributes simultaneously. Moreover, we have shown that Deep WSF could be used to pretrain models on auxiliary datasets, not only to speed up the learning process, but also uncover better representations for the attribute of interest.

Future avenues include experimenting with other applications, e.g., in the area of speech recognition, especially for multi-source speech recognition and we will investigate multilinear extensions of the proposed framework [54], [55].

Acknowledgments

George Trigeorgis is a recipient of the fellowship of the Department of Computing, Imperial College London, and this work was partially funded by it. The work of Konstantinos Bousmalis was funded partially from the Google Europe Fellowship in Social Signal Processing. The work of Stefanos Zafeiriou was partially funded by the EPSRC project EP/J017787/1 (4D-FAB). The work of Björn W. Schuller was partially funded by the European Community's Horizon 2020 Framework Programme under grant agreement No. 645378 (ARIA-VALUSPA). The responsibility lies with the authors.

References

[1] P. Paatero and U. Tapper, "Positive matrix factorization: A nonnegative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[2] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization," *Proc. Nat. Academy Sciences*, vol. 101, pp. 4164–4149, 2004.

[3] K. Devarajan, "Nonnegative matrix factorization: An analytical and interpretive tool in computational biology," *PLoS Comput. Biol.*, vol. 4, 2008, Art. no. e1000029.

[4] M. W. Berry and M. Browne, "Email surveillance using non-negative matrix factorization," *Comput. Math. Org. Theory*, vol. 11, pp. 249–264, 2005.

[5] S. Zafeiriou, A. Tefas, I. Buciu, and I. Pitas, "Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 683–695, May 2006.

[6] I. Kotsia, S. Zafeiriou, and I. Pitas, "A novel discriminant non-negative matrix factorization algorithm with applications to facial image characterization problems," *IEEE Trans. Inform. Forensics Security*, vol. 2, no. 3, pp. 588–595, Sep. 2007.

[7] F. Weninger and B. Schuller, "Optimization and parallelization of monaural source separation algorithms in the openBliSSART toolkit," *J. Signal Process. Syst.*, vol. 69, pp. 267–277, 2012.

[8] C. H. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 45–55, Jan. 2010.

[9] C. Cing, X. He, and H. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proc. SIAM Data Mining Conf.*, 2005, pp. 606–610.

[10] J. Herrero, A. Valencia, and J. Dopazo, "A hierarchical unsupervised growing neural network for clustering gene expression patterns," *Bioinformatics*, vol. 17, pp. 126–136, 2001.

[11] Y. Zhao and G. Karypis, "Hierarchical clustering algorithms for document datasets," *Data Mining Knowl. Discovery*, vol. 10, pp. 141–168, 2005.

[12] S. J. Prince, J. Warrell, J. H. Elder, and F. M. Felisberti, "Tied factor analysis for face recognition across large pose differences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 970–984, Jun. 2008.

[13] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Int. J. Data Warehousing Mining*, vol. 3, pp. 1–13, 2007.

[14] Y. Zhang and Z.-H. Zhou, "Multilabel dimensionality reduction via dependence maximization," *ACM Trans. Knowl. Discovery Data*, vol. 4, 2010, Art. no. 14.

[15] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and W. B. Schuller, "A Deep semi-NMF model for learning hidden representations," in *Proc. Int. Conf. Mach. Learn.*, 2014, vol. 32, pp. 1692–1700.

[16] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, pp. 504–507, 2006.

[17] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische Mathematik*, vol. 14, pp. 403–420, 1970.

[18] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Laboratory Syst.*, vol. 2, pp. 37–52, 1987.

[19] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 556–562.

[20] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[21] J.-H. Ahn, S. Choi, and J.-H. Oh, "A multiplicative up-propagation algorithm," in *Proc. Int. Conf. Mach. Learn.*, 2004, p. 3.

[22] S. Lyu and X. Wang, "On algorithms for sparse multi-factor NMF," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 602–610.

[23] A. Cichocki and R. Zdunek, "Multilayer nonnegative matrix factorization," *Electron. Lett.*, vol. 42, pp. 947–948, 2006.

[24] H. A. Song and S.-Y. Lee, "Hierarchical data representation model - multi-layer NMF," in *Proc. Int. Conf. Learn. Representations*, 2013, Art. no. 1301.6316.

[25] H. Liu, Z. Wu, X. Li, D. Cai, and T. S. Huang, "Constrained nonnegative matrix factorization for image representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1299–1311, Jul. 2012.

[26] I. Kotsia, S. Zafeiriou, and I. Pitas, "Novel discriminant non-negative matrix factorization algorithm with applications to facial image characterization problems," *IEEE Trans. Inform. Forensics Security*, vol. 2, no. 3, pp. 588–595, Sep. 2007.

[27] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neuroscience*, vol. 2, pp. 1019–1025, 1999.

[28] J. Malo, I. Epifanio, R. Navarro, and E. P. Simoncelli, "Nonlinear image representation for efficient perceptual coding," *IEEE Trans. Image Process.*, vol. 15, no. 1, pp. 68–80, Jan. 2006.

[29] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, pp. 359–366, 1989.

[30] Y. Nesterov, "Gradient methods for minimizing composite objective function," *Math. Program.*, Springer, vol. 140, no. 1, pp. 125–161, 2013.

[31] M. Li, T. Zhang, Y. Chen, and A. Smola, "Efficient mini-batch training for stochastic optimization," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 661–670.

[32] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," presented at the Int. Conf. Learning Representations, San Diego, CA, USA, 2015.

[33] D. M. Cvetkovic, M. Doob, and H. Sachs, *Spectra of Graphs: Theory and Application*. New York, NY, USA: Academic, 1980, vol. 413.

[34] M. Belkin and P. Niyogi, "Using manifold structure for partially labelled classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 328–340.

[35] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, 2006.

[36] Y. Hao, C. Han, G. Shao, and T. Guo, "Generalized graph regularized non-negative matrix factorization for data representation," in *Proc. Int. Conf. Inform. Technol. Softw. Eng.*, 2013, pp. 1–12.

[37] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neurosci.*, vol. 3, pp. 71–86, 1991.

[38] S. Li, X. W. H. X. W. Hou, H. J. Z. H. J. Zhang, and Q. S. C. Q. S. Cheng, "Learning spatially localized, parts-based representation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recog.*, 2001, vol. 1, pp. I-207–I-212.

[39] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression (PIE) database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, pp. 1615–1618, 2003.

[40] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, "XM2VTSDB: The extended M2VTS database," in *Proc. 2nd Int. Conf. Audio Video-Based Biometric Person Authentication*, 1999, pp. 965–966.

[41] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[42] N. Guan, D. Tao, Z. Luo, and B. Yuan, "NeNMF: An optimal gradient method for nonnegative matrix factorization," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2882–2898, Jun. 2012.

[43] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image Vision Comput.*, vol. 28, pp. 807–813, 2010.

[44] C. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 45–55, Jan. 2010.

[45] N. Gillis and A. Kumar, "Exact and heuristic algorithms for semi-nonnegative matrix factorization," *SIAM J. Matrix Anal. Appl.*, vol. 36, pp. 1404–1424, 2014.

[46] C. Boutsidis and E. Gallopoulos, "SVD based initialization: A head start for nonnegative matrix factorization," *Pattern Recog.*, vol. 41, pp. 1350–1362, 2008.

[47] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 585–591.

[48] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2003, pp. 267–273.

[49] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistics Quart.*, vol. 2, pp. 1–2, 1955.

[50] Y. A. LeCun, L. Bottou, G. B. Orr, and K. R. Müller, "Efficient backProp," *Neural Netw., Tricks Trade*, vol. 7700, pp. 9–48, 2012.

[51] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 153–160.

[52] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark Localization Challenge," in *Proc. Comput. Vision Pattern Recog.*, 2013, pp. 397–403.

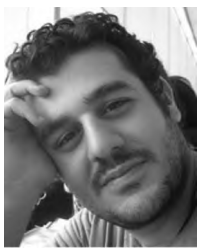
[53] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "A semi-automatic methodology for facial landmark annotation," in *Proc. IEEE Conf. Comput. Vision Pattern Recog. Workshops*, 2013, pp. 896–903.

[54] S. Zafeiriou, "Discriminant nonnegative tensor factorization algorithms," *IEEE Trans Neural Netw.*, vol. 20, no. 2, pp. 217-235, Feb. 2009.

[55] S. Zafeiriou, "Algorithms for nonnegative tensor factorization," *Tensors Image Process. Comput. Vision*, vol. 1, pp. 105–124, 2009.

George Trigeorgis received the MEng in artificial intelligence from the Department of Computing, Imperial College London, in 2013, where he is currently working toward the PhD degree. His current research interests include deep learning, computer vision, and matrix factorization.

Konstantinos Bousmalis received the BS degree in computer science from Lafayette College, Easton, PA, in 2005, the MSc degree in artificial intelligence from the University of Edinburgh, Edinburgh, United Kingdom, in 2007, and the PhD degree from Imperial College London in 2014. He is currently with the Brain team at Google Research in Mountain View, California, USA. During his PhD he was awarded the Google European Doctoral Fellowship. His research interests include Bayesian nonparametrics, Deep Learning, and Computer Vision for Robotics. He is a member of the IEEE.



Stefanos Zafeiriou is a senior lecturer in Pattern Recognition/Statistical Machine Learning for Computer Vision in the Department of Computing, Imperial College London. He has been awarded one of the prestigious Junior Research Fellowships from Imperial College London in 2011 to start his own independent research group. He has participated in more than 10 EU, British and Greek research projects. He currently serves as an associate editor in *IEEE Transactions on Cybernetics* and *Image and Vision Computing Journal.* He has been guest editor in more than four special issues and co-organized more than five workshops/special sessions in top venues such as CVPR/FG/ICCV/ECCV. His students have received very prestigious and highly competitive fellowships such as the Google Fellowship, the Intel Fellowship and the Qualcomm fellowship. He has more than 2,000 citations to his work. He is a member of the IEEE.

Björn W. Schuller received his diploma, doctoral degree, habilitation, and Adjunct Teaching Professorship all in EE/IT from TUM in Munich/Germany. He is currently a reader (associate professor) in Machine Learning at Imperial College London, United Kingdom, full professor and chair of Complex & Intelligent Systems at the University of Passau, Germany, and the co-founding CEO of audEERING UG. Previously, he headed the Machine Intelligence and Signal Processing Group at TUM from 2006 to 2014. In 2013 he was also invited as a permanent visiting professor at the Harbin Institute of Technology, P.R. China and the University of Geneva, Switzerland. In 2012, he was with Joanneum Research in Graz, Austria, remaining an expert consultant. In 2011 he was guest lecturer in Ancona, Italy and visiting researcher in the Machine Learning Research Group of NICTA in Sydney, Australia. From 2009 to 2010 he was with the CNRS-LIMSI in Orsay, France, and a visiting scientist at Imperial College. He co-authored more than 500 technical contributions (more than 10,000 citations, h-index = 49) in the field. He is a senior member of the IEEE.