Learning new acoustic events in an HMM-based system using MAP adaptation

Jürgen T. Geiger, Mohamed Anouar Lakhal, Björn Schuller, and Gerhard Rigoll

Institute for Human-Machine Communication
Technische Universität München
80290 Munich, Germany
{geiger, lakhal, schuller, rigoll}@mmk.ei.tum.de

Abstract

In this paper, we present a system for the recognition of acoustic events suited for a robotic application. HMMs are used to model different acoustic event classes. We are especially looking at the open-set case, where a class of acoustic events occurs that was not included in the training phase. It is evaluated how newly occuring classes can be learnt using MAP adaptation or conventional training methods. A small database of acoustic events was recorded with a robotic platform to perform the experiments.

Index Terms: Acoustic Event Classification, Hidden Markov Models, Model Adaptation

10.21437/Interspeech.2011-113

1. Introduction

Natural speech is a very intuitive communication channel for humans. Automatic speech recognition is a very prominent research field. But there are other ways of auditory communication. Other non-speech acoustic events can also carry important information. The detection and classification of such acoustic events is a less explored research area. In a typical office environment for example, many different sounds are produced either by a human or by objects handled by humans. Examples are human non-vocal sounds like coughing or clapping or other sounds like keyboard typing or closing a door. Being able to detect and identify these acoustic events can help to analyze the human activity that takes place. Acoustic event detection (AED) and classification (AEC) can also be used to enhance automatic speech recognition. AED and AEC both can be seen as disciplines in the area of computational auditory scene analysis [1].

A good overview over recent advances in AED/AEC technology for acoustic events in an office environment is given in [2]. Several international evaluation campaigns are described, where different approaches have been deployed, mainly using HMMs or SVMs.

Other domains for the application of AED/AEC techniques are the detection of key audio events in sports games [3] or affective video content analysis [4]. AED/AEC can also be part of a robot audition system as described in [5].

Whereas most of the listed work describes only closed-set recognition systems where the same classes occur during training and testing, open-set recognition is the challenge where previously unknown classes may occur in the test phase. In this case, the fact that an acoustic event belongs to a previously unknown class can be detected. This is known as novelty detection [6]. After detecting an acoustic event as being novel, the problem is then to add a new class to the classification system. This problem is analogous to the enrollment of a new speaker in a speaker recognition system. The standard approach utilized in speaker verification is to use a universal background model (UBM) which contains training data of many different speakers and to use maximum a posteriori (MAP) adaptation to derive a model for a new speaker based on limited amounts of enrollment data [7]. However, constructing a UBM from many different acoustic events, which can be of very diverse nature, might not be efficient.

In this work, we present a system for a robotic platform to detect and classify acoustic events. As a classifier, we use an HMM-based system with standard MFCC features, where each class of acoustic events is modeled by one HMM. We want to concentrate on the case where an acoustic event occurs that is not known to the system and is already detected as novel. Two different approaches to add new classes (with limited data) to the system are evaluated. The first approach is to simply perform a complete EM training cycle with the instances of the new class. The second approach uses MAP adaptation to derive the model of a new class from the model of one of the known classes. To evaluate the system, a small database of distinctive acoustic events from an office environment was recorded. A possible application scenario is a robotic platform which can learn typical sounds of its daily environment.

In Section 2, a general system overview and a description of the methods to learn new classes is given. The recorded database, experiments and results are described in Section 3. The paper closes with a conclusion in Section 4.

2. Acoustic event classification system

2.1. Overview

The acoustic events were segmented during the recording of the database, thus we only tackle the problem of classification, ignoring detection. As a baseline system, we extract MFCCs as features and use continuous HMMs for the classification task. The system is then improved by optimizing the number of HMM states using an approach based on the Bakis length modeling method. The main part of this work is the problem of adding a new class to the classifier. When an acoustic event is detected as previously unknown, it can be added to the system such that when it occurs the next time, it can be regarded as known to the system. In order to do this with an HMM-based system, a new model must be created for the new class. Two possibilities to achieve this are compared here. The first is to repeat the training phase as it was done with the other classes (using EM algorithm). As a second possibility, MAP adaptation can be used to create a model for the new class, leaving the other models unchanged.

2.2. Feature Extraction

As acoustic features, we use standard MFCCs (+ Energy) with delta and acceleration coefficients. Whereas the baseline system uses 12 MFCC coefficients, the best results could be achieved using 8 MFCC coefficients, which, together with delta and acceleration coefficients, leads to a total number of 27 extracted features. The features are calculated for overlapping windows of 25 ms size using 60 % overlap (which corresponds to a frame shift of 10 ms). During feature extraction, the stereo signal is converted to mono by averaging over both channels.

2.3. Classification

Hidden Markov Models (HMMs), implemented in HTK [8], are used as a classifier to recognize the acoustic events. An HMM is a generative statistical classifier which represents a sequence of feature vectors using two statistical processes, an internal (hidden) Markov Chain and an observable state sequence omitting the observations (feature vectors).

We use continuous HMMs with left-to-right topology; the observations are modeled by a mixture of Gaussians, defined by a mean vector, covariance matrix and mixture weight for each Gaussian. For each class of acoustic events, one model is created in the training phase of the classifier.

2.4. Optimizing the number of HMM states

In order to optimize the number of HMM states, we use an approach based on Bakis length modeling [9] as presented in [10]. The baseline system uses the same fixed number of states for every model (also referred to as fixed length modeling). The Bakis length modeling method proposes to set the number of states of each model to a fraction of the average length of the instances of the class. Here the length would be the length of the recording of the acoustic event in seconds, which can be derived from the recorded database. In our variant of the Bakis length modeling method, the number of states $n(s)$ for a model s is set to

$$n(s) = c + f \cdot \bar{t}(s), \tag{1}$$

where $\bar{t}(s)$ is the average length of the recordings corresponding to class s while c and f are an additive constant and a factor, respectively, which have to be optimized heuristically. Using this method, the number of HMM states can be modeled to better fit the class statistics.

2.5. Learning new acoustic events

Once an acoustic event has been detected as being novel (novelty detection is not covered in this work), it can be added to the database of known acoustic events. We compare two ways to learn a new class of acoustic events. The first, conventional way (which is referred to as *train* method in the following,) is to perform a normal training cycle using expectation maximization (EM); a complete retraining of all models is done, which can be very time consuming, depending on the amount of training data. The training must be performed with all classes to ensure proper model initialization.

Another possibility to add a new class to the classifier is to use MAP adaptation. This method will be called *adapt* method in the following. The model for the new class is not built up from scratch, but it is derived from another model using MAP adaptation. First of all, the new, unknown acoustic event is classified as one of the known classes, then the class it is classified as is used as a starting point for MAP adaptation. The new model is created by copying and adapting the model of the most similar class. As adaptation data, the recorded (one or more) instances of the new class are used.

We use MAP adaptation to adapt the means, mixture weights and variances of the output probabilities of the HMMs. The mean of mixture component m is adapted using Eq. (2):

$$\hat{\boldsymbol{\mu}}_m = \frac{N_m}{N_m + \tau}\bar{\boldsymbol{\mu}}_m + \frac{\tau}{N_m + \tau}\boldsymbol{\mu}_m, \tag{2}$$

where $\hat{\boldsymbol{\mu}}_m$ is the adapted mean, $\bar{\boldsymbol{\mu}}_m$ is the mean of the observed adaptation data, $\boldsymbol{\mu}_m$ ist the old mean of the Gaussian, τ is a weighting factor and N_m is the occupation likelihood of the adaptation data for mixture component m.

Eq. (3) is used to adapt the variances of the output probability distributions of the HMMs:

$$\hat{\boldsymbol{\sigma}}_m^2 = \frac{N_m}{N_m + \tau}\mathrm{E}_m(\mathbf{x}^2) + \frac{\tau}{N_m + \tau}(\boldsymbol{\sigma}_m^2 + \boldsymbol{\mu}_m^2) - \hat{\boldsymbol{\mu}}_m^2. \tag{3}$$

Here, $\hat{\boldsymbol{\sigma}}_m^2$ is the adapted variance, $\boldsymbol{\sigma}_m^2$ is the old variance and $\mathrm{E}_m(\mathbf{x}^2)$ is the expected value of the squared observation vector $\mathbf{x}^2$.

The adaptation of mixture weight w_m for mixture m follows Eq. (4):

$$\hat{w}_m = (\frac{N_m}{N_m + \tau}\frac{N_m}{T} + \frac{\tau}{N_m + \tau}w_m)\gamma, \tag{4}$$

where $\hat{w}_m$ is the adapted mixture weight, w_m is the original mixture weight, T is the length of the adaptation data, and γ is a normalizing factor, which is needed to ensure that all mixture weights sum to 1.

The weighting factor τ is optimised heuristically. Smaller values of τ lead to higher adaptation, which means (in case for the mean) that the new mean is nearer at the mean of the adaptation data than at the old mean. If τ is set to zero, the new model corresponds to a model trained only with the adaptation data. Conversely, higher values of τ lead to less adaptation. In the case of $\tau \to \infty$, the old model is copied in order to get the new model, while neglecting the adaptation data.

3. Experiments

3.1. Database

For testing purposes we recorded a database of non overlapping acoustic events with the microphones of our robotic platform, called ALIAS (ambiant living assistant). ALIAS is a mobile robot system that interacts with elderly users and monitors and provides cognitive assistance in daily life. In order to orient himelf in his environment, ALIAS should be able to recognize the sounds that surround him. It could also be possible to control the robot with (non-verbal) acoustic commands.

The robotic platform ALIAS is depicted in Figure 1.

With the pair of AKG C-520 L microphones of the depicted robotic platform, acoustic events were recorded in a silent office environment.

During the recording process, an automatic energy-based voice activity detection (VAD) algorithm was used to detect and segment the acoustic events. Thus, the database contains segmented recordings of acoustic events.

15 different classes of acoustic events occuring in an office environment were chosen to be included in the database. These classes include ambient sound events and command-oriented social signals and gestures that are intended to provide a home service robot with better understanding of its environment. Some special acoustic events that were included in

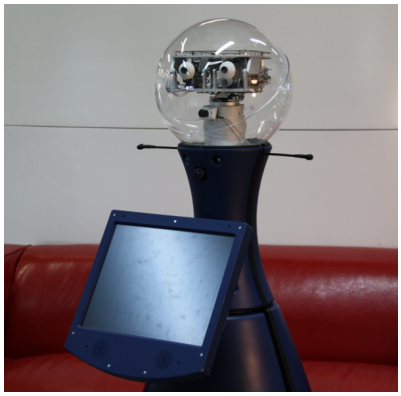

Figure 1: Robotic platform ALIAS with two AKG C-520 L microphones

the database are *speech* for normal speech and *garbage* for all sounds that occured during the recording but are not included in any of the other classes. Table 1 shows the 15 different classes, their frequency in the database, and the average length of the recordings (which is needed for the optimization of HMM state numbers). The average recording length includes a short phase of silence at the beginning and at the end of each recording. The recorded database is not intended to be used for a very generalized recognition system. For example, it will not be able to recognize all kinds of closing doors. The database is rather intended to be a small sample of very specific sounds from a robot's environment. In total, the database is made up of 506 single acoustic events.

Class	number of files	avg. length (in s)
chair rolling	22	1.68
chair squeak	24	1.16
clap	36	1.27
cough	51	1.56
door closing	21	1.42
finger snap	30	1.13
garbage	25	1.96
glass placement	51	1.26
key laydown	16	1.25
key rattle	41	2.43
keyboard	39	1.43
paper rustle	42	2.30
paper tear	44	1.38
speech	36	2.69
steps	28	2.03

Table 1: Overview over the 15 classes of different acoustic events in the database, their frequency and the average length of the recordings in s. The total number of acoustic events in the database is 506, with an average length of 1.69s.

3.2. Baseline system results

For the baseline system, 12 MFCC coefficients are calculated during feature extraction. Together with the energy coefficient and delta and acceleration coefficients, this sums up to a total of 39 features. The baseline system uses fixed length modeling (with 6 HMM states for each class) to determine the number of HMM states.

In order to get reliable results, a 5-fold stratified cross validation is applied. Therefore, the instances of each class in the database are randomly divided into five subsets and the experiments are conducted five times, where each time another one of the subsets is used for testing. Each time, the remaining four subsets are used to train the models. The final classification result is obtained by averaging over the results of the five single experiments. Table 2 shows the confusion matrix for the baseline system.

	a chair rolling	b chair squeak	c clap	d cough	e door closing	f finger snap	g garbage	h glass	i key laydown	j key rattle	k keyboard	l paper rustle	m paper tear	n speech	o steps	total error in %
a	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b	0	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	0	0	34	0	0	2	0	0	0	0	0	0	0	0	0	5.6
d	0	0	0	51	0	0	0	0	0	0	0	0	0	0	0	0
e	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0
f	0	0	0	1	0	28	0	0	0	0	0	0	1	0	0	6.7
g	0	1	0	0	0	0	20	0	0	0	1	0	0	1	2	20
h	0	1	1	0	0	0	0	49	0	0	0	0	0	0	0	4
i	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0
j	0	0	0	0	0	0	0	0	0	41	0	0	0	0	0	0
k	0	0	2	0	0	1	0	0	0	0	35	1	0	0	0	10.3
l	0	0	0	0	0	0	0	0	0	0	0	42	0	0	0	0
m	0	0	0	0	0	1	0	0	0	0	1	2	40	0	0	9.1
n	0	0	0	0	0	0	0	0	0	0	0	0	0	36	0	0
o	0	0	0	0	2	0	0	0	0	0	0	0	0	0	26	7.2

Table 2: Confusion matrix for the baseline recognition system. Most of the classes are recognized without errors.

Eight out of the 15 total classes are recognized with 100 % accuracy. The total accuracy is 95.9 %, which corresponds to an error rate of 4.1 %.

3.3. Results for optimization of HMM state numbers

Using Eq. 1, the number of HMM states is optimized. The optimal parameters are $f = 2.5$ and $c = 4$. The application of these parameters (which corresponds to state numbers between 7 and 11) leads to an error rate of 3.2 %, which is a relative improvement of 23.7 % compared to the baseline system.

Finally, the number of MFCC coefficients is surveyed and the best result is achieved with eight coefficients (27 features in total) with an error rate of 2.8 %.

The results of the baseline system including the optimizations are summarized in Table 3.3.

system setup	error rate (in %)
baseline	4.1
improved state numbers	3.2
improved state numbers + MFCC	2.8

Table 3: Summary of the recognition results of the baseline system and its improvements. Using a separate number of states for each HMM and adjusting the number of MFCC components reduced the error rate by roughly one third.

3.4. Results for learning new classes

To evaluate how good a new class can be learnt by the system, the experimental setup as described in the following is used. The described experiment is repeated 15 times, where each time, one of the acoustic event classes is used as the "new" class.

Using the structure of the improved baseline system, the system is trained using all but one of the classes, where the same amount of training data is used as for the baseline system. Then, the new class is added to the system using one of the two methods described in Section 2.5 (complete retraining or adaptation). One, two or three instances of the unknown class are used to create the new model. The remaining data of the new class is used for testing. A 5-fold cross validation is used to have each instance (of the "old" classes) in the test set once. In addition, this experiment is repeated several times, where each time different instances of the new class are used to create the new model. The final average results are shown in Table 3.4

	1 instance		2 instances		3 instances	
	new	rest	new	rest	new	rest
train	85.4	3.0	57.7	3.0	35.6	3.1
adapt	33.9	4.6	25.6	4.4	21.4	4.3

Table 4: Error rates (in %) for learning a new class using either full training (*train*) or MAP adaptation (*adapt*) to create the model of the new class. Using one, two, or three instances of the new class was evaluated, whereby each time, error rates are reported for the new class (*new*) as well as for the other classes (*rest*).

These results show that when using the *train* method, the average error rate for the newly added class is very high when the model is created from only one instance of the class, but, not surprisingly, it decreases very strongly when more instances are used. For the *adapt* method, using only one instance of the new class already delivers results that are much better than for the *train* method. Here, too, the performance increases when more instances of the new class are used, but not so strongly as with the first method.

The classification results for the other classes must also be looked at. With the *train* method, the effect of adding a new class on the classification performance of the previously known classes is almost neglectable. This is not the case when MAP adaptation is used. In this case, the error rate for the previously known classes increases. Looking at the detailed results shows that this is due to confusion of the class that was taken as a base for MAP adaptation with the new class. However, in a one-tailed test, the downgrade is not significant.

4. Conclusion and Future Work

We implemented a system for acoustic event classification based on HMMs and tested it with our own database of acoustic events recorded in an office environment. Then, the case of open-set recognition was regarded. When an acoustic event is detected as being previously unknown, how can it be added to the system as a new class. Two approaches were compared, using full training or MAP adaptation. The results showed that when using full training, the accuracy for the new class was not acceptable when only one instance of this class was used to train the model of this new class. However, using two or three instances improved the results. The classification performance of the other classes was not affected by adding a new class. As another approach to add a new class, MAP adaptation was evaluated. With only very few instances of a new class, MAP adaptation outperformed full training regarding the error rate for the new class. However, the error rate for the other classes was affected negatively by adding a new class.

It is also notable that using MAP adaptation instead of full training requires a lot less computational time. This fact, together with the good results of the *adapt* method compared to the *train* method when using only one single instance of a new class leads to the conclusion, that the method of MAP adaptation should be prefered when implementing a real-time system which can detect and learn new classes of acoustic events online.

In our ongoing research, we want to integrate a novelty detection system into the system to perform the whole process of detecting unknown acoustic events and learning a new class for this event. We also want to implement the described system on a robotic platform to test it in a real environment.

5. Acknowledgements

This work was partially supported by the project AAL-2009-2-049 "Adaptable Ambient Living Assistant" (ALIAS) co-funded by the European Commission and the German Federal Ministry of Education (BMBF) in the Ambient Assisted Living (AAL) programme and by the DFG excellence initiative research cluster *Cognition for Technical Systems - CoTeSys*, see also `www.cotesys.org`.

6. References

[1] D.L. Wang and G.J. Brown, *Computational auditory scene analysis: Principles, algorithms, and applications*, IEEE Press, 2006.

[2] A. Temko, C. Nadeu, D. Macho, R. Malkin, C. Zieger, and M. Omologo, "Acoustic event detection and classification," in *Computers in the Human Interaction Loop*, pp. 61–73. Springer, 2009.

[3] Q. Huang and S. Cox, "Using High-level Information to Detect Key Audio Events in a Tennis Game," in *Proc. Interspeech*, 2010, pp. 1409–1412.

[4] M. Xu, L.T. Chia, and J. Jin, "Affective content analysis in comedy and horror videos by audio emotional event detection," in *2005 IEEE International Conference on Multimedia and Expo*. IEEE, 2005.

[5] H.G. Okuno, T. Ogata, K. Komatani, and K. Nakadai, "Computational Auditory Scene Analysis and Its Application to Robot Audition," in *Proc. International Conference on Informatics Research for Development of Knowledge Society Infrastructure*. IEEE Computer Society, 2004, pp. 73–80.

[6] J.H. Bach and J. Anemüller, "Detecting novel objects in acoustic scenes through classifier incongruence," in *Proc. Interspeech*, 2010, pp. 2206–2209.

[7] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[8] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, et al., "The HTK book (for HTK version 3.4)," *Cambridge University Engineering Department*, 2006.

[9] R. Bakis, "Continuous speech recognition via centisecond acoustic states," *The Journal of the Acoustical Society of America*, vol. 59, pp. S97, 1976.

[10] J. Geiger, J. Schenk, F. Wallhoff, and G. Rigoll, "Optimizing the Number of States for HMM-Based On-line Handwritten Whiteboard Recognition," in *2010 12th International Conference on Frontiers in Handwriting Recognition*. IEEE, 2010, pp. 107–112.