DISCRIMINATIVE TRAINING OF UNSUPERVISED ACOUSTIC MODELS FOR NON-SPEECH AUDIO

Antti Eronen and Toni Heittola

Tampere University of Technology, Institute of Signal Processing
P.O.Box 553, FIN-33101 Tampere, Finland, antti.eronen@tut.fi, toni.heittola@tut.fi

ABSTRACT

This paper studies acoustic modeling of non-speech audio using hidden Markov models. Simulation results are presented in two different application areas: audio-based context awareness and music classification, the latter focusing on recognition of musical genres and instruments. Two training methods are evaluated: conventional maximum likelihood estimation using the Baum-Welch algorithm, and discriminative training, which is expected to improve the recognition accuracy on models with a small number of component densities in state distributions. Our approach is unsupervised in the sense that we do not know what are the underlying acoustic classes that are modeled with different HMM states. In addition to reporting the achieved recognition results, analyses are made to study what properties of sound signals are captured by the states.

1. INTRODUCTION

The aim of this paper is to evaluate the usefulness of hidden Markov models and discriminative training in non-speech audio classification tasks. We consider two different application areas: audio-based context awareness and music classification. In the former, the aim is to decide the context or environment based on audio information only. The potential applications in this field include e.g. intelligent wearable devices that adjust the mode of operation according to the context of the user, such as a meeting or a noisy street. In music classification, two subtasks are considered; recognition of the musical genre and musical instruments. Both components are needed for example in a content-based music indexing and retrieval system.

In automatic speech recognition, HMMs are the tool most commonly used to model speech characteristics. In other audio content analysis applications they are being used to an increasing degree. In speech recognition and text-dependent speaker recognition, supervised acoustic models are typically used [1], which means that we have e.g. phonetically labeled training sequences for each class, and the complete HMM for the class is constructed by concatenating the trained sub-word models.

In this paper, the focus is on unsupervised acoustic modeling, where the underlying acoustic classes are not known. It is not clear what the acoustic classes should be in music or in environmental audio. Thus, the unsupervised approach is attractive as the first attempt. This is analogous to the text-independent task in speaker identification. A similar approach has been taken in the MPEG-7 standard for the description of general sound similarity [2]. They model each class with a continuous density HMM trained with maximum *a posteriori* estimation.

Despite the diverse nature of our target applications, the use of HMMs to model the feature statistics has proved out to be a well-performing approach. Their use is motivated by at least two aspects. First, at least in theory there are no limitations to the class of probability distributions representable by HMMs, given enough hidden states and suitable state densities [3]. Second, in classification tasks we do not even need to accurately model the class-conditional densities; for classification purposes efficient modeling of class boundaries is sufficient [3]. Thus, even if our model is incapable of modeling all the variations in the sound, we can train it to focus on the differences between classes using discriminative training methods. Due to the highly varying material and unsupervised nature of these tasks, it is most likely that the acoustic models we are using are not able to sufficiently model the observation statistics. For these reasons, we propose using discriminative training of model parameters instead of conventional maximum-likelihood training.

2. DISCRIMINATIVE TRAINING OF HMMS

In each of our classification tasks, our acoustic data comprises a training set that consists of the recordings $\mathrm{O} = (\mathbf{O}^1,...,\mathbf{O}^R)$ and their associated class labels $L = (l^1,...,l^R)$. Depending of the application, L can express the context where the recording has been made, the musical genre, or the musical instrument playing on the musical excerpt *r*. To be more specific, $\mathbf{O}^r$ denotes the sequence of feature vectors measured from recording *r*. The purpose of the acoustic models is to represent the distribution of feature values in each class in this training set.

2.1. Description of the hidden Markov model

A continuous-density hidden Markov model (HMM) with N states consists of a set of parameters θ that

comprises the *N*-by-*N* transition matrix, the initial state distribution, and the weights, means and diagonal variances of Gaussian mixture model (GMM) state emission densities. The possibility to model sequences of states with different statistical properties and transition probabilities between them makes intuitively sense in our applications, since sounds are dynamic phenomena. For instance, one can imagine standing next to a road, where cars are passing by. When a car approaches, its sound changes in a certain manner, and after it has passed there is a clear change in its sound due to the Doppler effect. Naturally, when no cars are passing by the sound scene is rather quiet. Hopefully, the different states in the model are able to capture the different stages, and the statistical variation between different roads, cars, and recording times is modeled to some extent by the different components in the GMM state densities.

In our baseline system, the HMM parameters are iteratively optimized with the Baum-Welch algorithm [4]. This algorithm iteratively finds a local maximum of the maximum likelihood (ML) objective function

$$F(\Theta) = \sum_{c=1}^{C} \sum_{r \in A_c} \log p(\mathbf{O}^r \mid c), \quad (1)$$

where Θ denotes the entire parameter set of all the classes $c \in \{1,...,C\}$, and A_c is the subset of $[1,R]$ that denotes the recordings from the class *c*. In the recognition phase, an unknown recording $\mathbf{O}$ is classified using the maximum *a posteriori* rule:

$$\hat{c} = \arg\max_{c} p(\mathbf{O} \mid c). \quad (2)$$

The needed likelihoods can be efficiently computed using the forward-backward algorithm, or approximated with the likelihood of the single most likely path given by the Viterbi-algorithm.

2.2. A discriminative training algorithm

Maximum Likelihood estimation is well justified if the observations are distributed according to the assumed statistical model. In our applications, it is very unlikely that a single HMM could capture all the statistical variation of the observations from an arbitrary environment or classical music, for instance. Moreover, the training databases are much smaller than for example the available speech databases, preventing the reliable estimation of parameters for complex models with high amounts of component densities. In applications where computational resources are limited such as context-awareness targeted for embedded applications, we are forced to use models with as few Gaussians as possible, since their evaluation poses the computationally most demanding step in the recognition phase. In these cases a model mismatch occurs and other approaches than ML may lead into better recognition results. Discriminative training methods such as the maximum mutual information (MMI) aim at maximizing the ability to distinguish between the observation sequences generated by the model of the correct class and those generated by models of other classes [4].

Different discriminative algorithms have been proposed in the literature. The algorithm used in this paper has been presented just recently, and one of its benefits is a straightforward implementation. The algorithm was proposed by Ben-Yishai & Burshtein, and is based on an approximation of the maximum mutual information criterion [5]. Their *approximated maximum mutual information* (AMMI) criterion is:

$$J(\Theta) = \sum_{c=1}^{C} \left\{ \sum_{r \in A_c} \log\left[p(c)p\left(\mathbf{O}^r \mid c\right)\right] - \lambda \sum_{r \in B_c} \log\left[p(c)p\left(\mathbf{O}^r \mid c\right)\right] \right\}$$

where B_c is the set of indices of training recordings that were recognized as class *c*. The set B_c is obtained by maximum a posteriori classification performed on the training set. The parameter $0 \le \lambda \le 1$ controls the "discrimination rate".

The prior probabilities $p(c)$ do not affect the maximization of $J(\Theta)$, thus the maximization is equivalent to maximizing the following objective functions:

$$J_c(\Theta) = \sum_{r \in A_c} \log p\left(\mathbf{O}^r \mid c\right) - \lambda \sum_{r \in B_c} \log p\left(\mathbf{O}^r \mid c\right), \quad (5)$$

for all the classes $1 \le c \le C$. Thus, the parameter set of each class can be estimated separately, which leads to a straightforward implementation. The authors give the re-estimation equations for HMM parameters [5]. Due to space restrictions, we present only the re-estimation equation for the transition probability from state *i* to state *j*:

$$\bar{a}_{ij} = \frac{\sum_{r \in A_c} \sum_{t=1}^{T_r - 1} \xi_t(i,j) - \lambda \sum_{r \in B_c} \sum_{t=1}^{T_r - 1} \xi_t(i,j)}{\sum_{r \in A_c} \sum_{t=1}^{T_r - 1} \gamma_t(i) - \lambda \sum_{r \in B_c} \sum_{t=1}^{T_r - 1} \gamma_t(i)}, \quad (6)$$

where $\xi_t(i,j) = p(q_t = i, q_{t+1} = j \mid \mathbf{O}^r, c)$ and $\gamma_t = \sum_{j=1}^{N} \xi_t(i,j)$. The state at time *t* is denoted by q_t, and the length of the observation sequence $\mathbf{O}^r$ is T_r. In a general form, for each parameter ν the re-estimation procedure is

$$\nu = \frac{N(\nu) - \lambda N_D(\nu)}{G(\nu) - \lambda G_D(\nu)}$$

where $N(\nu)$ and $G(\nu)$ are accumulators that are computed according to the set A_c, and $N_D(\nu)$ and $G_D(\nu)$ are the discriminative accumulators computed according to the set B_c, obtained by recognition on the training set. This discriminative re-estimation can be iterated. We used typically 5 iterations, since the improvement in recognition accuracy was only minor beyond that. In many cases, using just one iteration would be enough since it sometimes gave the greatest improvement. The recognition was done only at the first

iteration, after which the set B_c stayed fixed. The following iterations still increase the AMMI objective function and increase the accuracy at least in the training set. However, continuing iterations too long causes the algorithm to overfit the training data, leading into poor generalization on unseen test data. Maximum of 5 iterations with $\lambda = 0.3$ was observed to give an improvement in most cases without much danger of overfitting.

3. EVALUATION DATABASES

3.1. Database of environmental recordings

The database for evaluating the context-awareness system consisted of 225 real-world recordings from a variety of different contexts, or environments. Typical environments include a street, lecture, meeting, family home, restaurant, and so on. Most recordings have been made with AKG C460B microphones and stored using a Sony (TCD-D10) digital audio tape recorder in 16-bit and 48 kHz sampling rate format. Some recordings have been made using a head and torso simulator. However, in our simulations no directional information is used but the system is trained and tested using monophonic material only. The training set consisted of 155 recordings of 24 contexts and 70 recordings of 16 contexts were tested. 160 s was used from each training recording, and testing was done with 60 s excerpts from the test recordings. The division between training and test sets was done randomly. A recording from a certain location was always used either for training or testing, but never for both, aiming at realistic testing of the generalization ability of the system.

3.2. Musical instrument sound database

The database for instrument recognition comprises 359 different musical solo pieces from 15 different instruments, which are: trombone, trumpet, bassoon, oboe, clarinet, flute, saxophone, church organ, cello, double bass, viola, violin, electric guitar, acoustic guitar, and piano. Most of the recordings have been collected from commercial CD recordings. The material also includes recordings made at Tampere University of Technology, and MIT Media Lab [6]. The recordings were randomly assigned into a training set comprising 294 recordings and a test set of 65 recordings. When there were several musical excerpts from the same CD recording, or from the same player with the same instrument, all these excerpts were included into either set. In this way, care was taken not to allow the system to train on any recordings of a certain instrument instance recorded in certain conditions which were included in the test set. This is important since the generalization across recording conditions, players, and instrument instances poses the greatest challenge in instrument recognition with monophonic material.

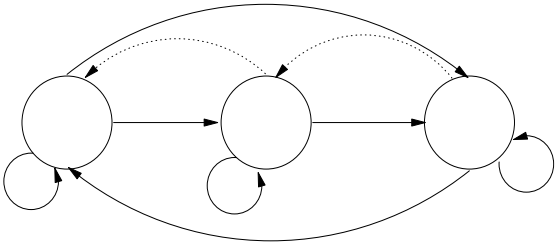

Figure 1. The two HMM topologies tested in this study. The dotted transitions have zero-probability in the left-right model.

3.3. Music database for genre recognition

Six musical genres are considered in this study: classical, electronic/dance, hip hop/rap, jazz/blues, rock/pop, and soul/rhythm&blues/funk. The database comprises a total of 493 different musical pieces taken from commercial CDs, the number of pieces from each genre varying between 37 and 125. Approximately 70 % of the recordings were randomly allocated into a training set, and testing was performed with the remaining recordings. A representative one-minute excerpt was used from each recording in both training and testing.

3.4. Feature extraction

The databases consist of audio recordings sampled either at 44.1 or 48 kHz. The feature extraction stage involves transforming the raw input into a representation with a lower dimensionality. Mel-frequency cepstral coefficients (MFCC) have been found to be a well performing feature set in these applications [7][8][9], and are used as the front-end parameters in our system. The input signal is first pre-emphasized with the FIR filter $1-0.97z^{-1}$ to flatten the spectrum. MFCC analysis is performed in 20 or 30 ms windowed frames advanced every 15 ms. The number of triangular filters spaced evenly on the mel-frequency scale was 40. The lowest frequency taken into consideration varied between 30-80Hz depending on the application, and the highest frequency was equal to half the sampling rate. The number of cepstral coefficients was between 11 and 16 after the zeroth coefficient was discarded. The features can be augmented by appending the time derivatives describing the dynamic properties of the cepstrum. For derivative approximation we used a 3-point first-order polynomial fit. The resulting features were both mean and variance normalized.

4. RESULTS

4.1. Model initialization

The Baum-Welch algorithm was used to train the baseline HMMs. The number of states (NS) and component densities per state (NC) was varied. Increasing the number of components in each state was obtained by gradually increasing the model order by splitting the component with the largest weight until the desired order NC was obtained. During training, a

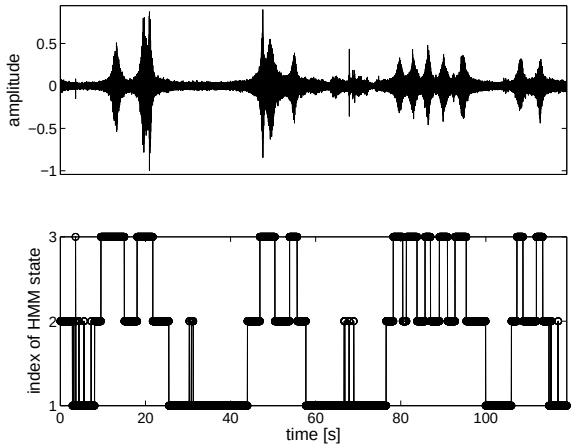

Figure 2. The top panel shows the amplitude of a recording made next to a road with passing cars. The bottom panel shows the Viterbi segmentation through a three-state HMM trained using the recording.

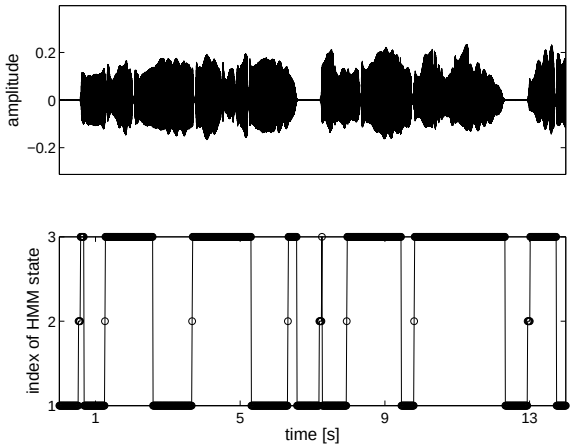

Figure 3. The top panel shows the amplitude of a solo melody played with a trumpet. The bottom panel shows the Viterbi segmentation through a three-state HMM trained for the trumpet class.

straightforward form of regularization was applied by adding a small constant to the variance elements falling below a predetermined threshold. The state means and variances were initialized by k-means clustering the training data into as many clusters as there were states in the model, and the state means and variances were initialized with the estimates computed from the different cluster segments. A heuristic minimum duration constraint was placed on the initial clustering segmentation to encourage longer continuous regions than just of a couple of frames. The model topologies tested were a fully-connected model, and a left-right model with skips. Figure 1 shows these two topologies. The dotted lines indicate the zero-probability transitions in the left-right model. Note that with two states, these two topologies are identical.

4.2. Studying the segmentation

To gain insight into the properties of sounds modeled by different HMM states it is useful to visually study the Viterbi segmentations after training, or in the test stage. In Figure 2, a three-state HMM has been trained using a recording of the sound next to a road. The top panel shows the amplitude of the signal as a function of time. The high amplitude peaks correspond to passing cars. The bottom panel shows the resulting Viterbi segmentation through the three states. The state number one models the silent periods when there are no cars passing; the second state the transition periods when a car is either approaching or getting farther, and the third state the period when the car is just passing or is very close to the recording place. A similar example with a musical sound is depicted in Figure 3. A three-state HMM was trained on trumpet recordings, and the segmentation is shown for a melody phrase of 15 seconds in duration. By listening it was found that state one represents high-pitched notes and pauses between notes, low-pitched notes are modeled with state three. Most interestingly, state two models the initial transients.

4.3. Context awareness

The simulation results for context awareness have been presented in [7] and are summarized in Table 1. The baseline system with a fully-connected HMM with two states and one component per state gave a recognition accuracy of 74.7%. Using discriminative training improved the accuracy to 77.3%. Increasing the number of component densities in state GMMs above one did not improve the baseline performance. Also, the improvement obtained with discriminative training was smaller when the number of components was increased. When the topology was left-right with skips, the recognition accuracy was 74.5% with a three-state model. Discriminative training improved only slightly the result which was 75.0%.

4.4. Instrument recognition

In instrument recognition, testing was done in adjacent 200-frame segments with no overlap, which corresponds to approximately 3 seconds in time. Table 2 shows the results for both training methods and varying model-topologies and number of states and mixture densities. The recognition accuracy gradually increases as the number of mixture densities is increased, converging to about 70% correct with 6 components and beyond. The conditions where discriminative training improved the accuracy over the baseline are shown in italics in Table 2. Only with very low order models having 2 states and 1 or 2 component densities per state a significant improvement is observed using discriminative training.

Table 1. Percentage correct in context awareness using both training methods and different model topologies. All models have single-Gaussian state densities.

	# states	Baum-Welch	Discriminative
Fully-connected	NS = 2	74.7	*77.3*
	NS = 3	73.7	73.7
Left-right	NS = 3	74.5	*75.0*

Table 2. Percentage correct in instrument recognition for both training methods and varying the model topology and complexity.

# components		NC=1	NC=2	NC=3	NC=6
Fully-connected	# states	Baum-Welch			
	NS = 2	61	65	67	70
	NS = 3	66	68	68	70
		Discriminative			
	NS = 2	*66*	*68*	67	*71*
	NS = 3	*67*	67	68	70
Left-right		Baum-Welch			
	NS = 3	67	67	68	69
		Discriminative			
	NS = 3	65	*68*	68	*70*

Table 3. Percentage correct in genre recognition for both training methods and varying the model topology and complexity.

# components		NC=1	NC=2	NC=3	NC=4
Fully-connected	# states	Baum-Welch			
	NS = 3	52.9	53.8	53.8	55.7
	NS = 4	54.3	54.6	55.0	56.8
		Discriminative			
	NS = 3	*53.4*	*56.9*	*56.1*	*58.5*
	NS = 4	*56.7*	54.4	*57.6*	*59.5*
Left-right		Baum-Welch			
	NS = 3	52.4	53.7	54.7	55.2
	NS = 4	54.0	55.0	56.2	56.1
		Discriminative			
	NS = 3	*54.7*	*55.5*	*58.1*	*58.2*
	NS = 4	*55.7*	*55.9*	*58.0*	*57.9*

4.5. Musical genre recognition

Table 3 shows the recognition accuracies in recognition of musical genre with varying model topologies and training methods. The large enough size of the music database makes it possible to show the numbers with one-decimal accuracy. It can be seen that discriminative training gives an improvement of only a few percentage points. However, improvement is observed almost consistently across the model orders and topologies tested. On the average, the recognizer is not very successful, even the best configuration gives less than 60% correct. However, the acoustic model classifier presented here is only a subpart of a more general genre recognizer that could utilize for example rhythmic information [7]. Thus, the relatively low accuracy provided by these models may be sufficient when combined with another classifier using different information sources.

5. CONCLUSION

Discriminative training improved the accuracy obtained with hidden Markov models having small number of states and component densities in states. With models having more complex state densities no improvement was observed. This is due to overfitting to the training data causing poor generalization to unseen test data. Using a low-complexity HMMs as classifier is interesting as such in context-awareness systems having limited computational resources. For genre recognition, HMMs are a useful basic building block for modeling acoustic information, and may be augmented with other information sources.

Future work should also consider the supervised approach and attempt to model certain phenomena in sounds with HMMs, such as notes from musical instruments with left-right models. In our unsupervised approach there is much variation in the quality of trained models due to the clustering initialization, and supervision should be used to find better initial estimates for model parameters, and to guide the selection of model topology.

6. REFERENCES

[1] G.R. Doddington, M.A. Przybocki, A.F. Martin, D.A. Reynolds, "The NIST speaker recognition evaluation – Overview, methodology, systems, results, perspective". *Speech Communication* 31, pp. 225-254, 2000.

[2] M. Casey, "General Sound Classification and Similarity in MPEG-7". In *Organized sound*, 6:2, 2002.

[3] J. Bilmes, "What HMMs Can Do", University of Washington, Department of Electrical Engineering, Technical Report UWEETR-2002-0003, Jan. 2002. Available at https://www.ee.washington.edu/techsite/papers/documents/UWEETR-2002-0003.pdf.

[4] R. Rabiner, B.-H. Juang. *Fundamentals of Speech Recognitio*n, PTR Prentice-Hall Inc., New Jersey, 1993.

[5] A. Ben-Yishai, D. Burshtein, "A Discriminative Training Algorithm for Hidden Markov Models". Submitted to *IEEE Transactions on Speech and Audio Processing.*

[6] K. D. Martin, Sound-Source Recognition: *A Theory and Computational Model*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 1999. Available at http://sound.media.mit.edu/Papers/kdm-phdthesis.pdf.

[7] A. Eronen, J. Tuomi, A. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, J. Huopaniemi, "Audio based context awareness – acoustic modeling and perceptual evaluation". In Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP 2003.

[8] G. Tzanetakis, P. Cook, "Musical Genre Classification of Audio Signals". *IEEE Transactions on Speech and Audio Processing*, Vol. 10, No. 5, pp. 293-302, Jul 2002.

[9] A. Eronen, "Comparison of features for musical instrument recognition". In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 19-22, Oct. 2001.

[10] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, T. Sorsa. "Computational auditory scene recognition". In Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP 2002.