INTERPOLATING HIDDEN MARKOV MODEL AND ITS APPLICATION TO AUTOMATIC INSTRUMENT RECOGNITION

Tuomas Virtanen and Toni Heittola

Tampere University of Technology

ABSTRACT

This paper proposes an interpolating extension to hidden Markov models (HMMs), which allows more accurate modeling of natural sounds sources. The model is able to produce observations from distributions which are interpolated between discrete HMM states. The model uses Gaussian mixture state emission densities, and the interpolation is implemented by introducing interpolating states in which the mixture weights, means, and variances are interpolated from the discrete HMM state densities. We propose an algorithm extended from the Baum-Welch algorithm for estimating the parameters of the interpolating model. The model was evaluated in automatic instrument classification task, where it produced systematically better recognition accuracy than a baseline HMM recognition algorithm.

Index Terms— Hidden Markov models, acoustic signal processing, musical instruments, pattern classification

1. INTRODUCTION

State models are widely used in modeling, automatic recognition, and synthesis of audio signals since they allow modeling non-stationary sounds. Hidden Markov model (HMM) with continuous state emission functions is the most commonly used model in automatic speech recognition, since it provides a good framework for modeling the adverse acoustic characteristics of natural speech, simultaneously with high-level language modeling. A major advantage of HMMs is also that the parameters can be efficiently estimating using training data.

A drawback in HMMs is that each state produces observations which are independent from each other, whereas natural sounds sources have a strong correlation in time. This limitation is usually circumvented by using delta and acceleration features which model the temporal evolution of the signal. More advanced models, for example trajectory models [1] or switching linear dynamical systems [2] model the temporal evolution explicitly. An interpolating state model where the observations were modeled by a piece-wise linear function was found to be efficient in modeling musical sounds [3].

Section 2 in this paper proposes a interpolating state model which implements the piece-wise linear model in a probabilistic framework which is similar to HMMs. This allows modeling sequences where observations move gradually from a state to another state. Figure 1 shows an example of interpolated state parameters for a single Gaussian. With an equal number of model parameters, the modeling error of the proposed model is significantly smaller.

Section 3 proposes an algorithm which trains the parameters of the model while taking into account the interpolations. Section 4 presents the decoding algorithm and Section 5 presents simulations where the model is shown to outperform existing acoustic models in an automatic instrument recognition task.

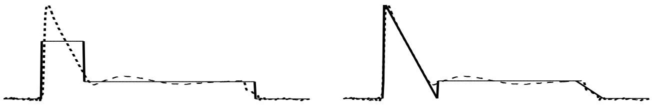

Fig. 1. *A feature of an example signal as a function of time (dashed lines) and the density means of the optimal state transition paths of a hidden Markov model (solid line, left plot) and the proposed model (solid line, right plot).*

2. THE MODEL

The model is defined by a set of main states and a set of interpolating states which parameters are interpolated from the main states. Let us denote the number of main states by K so that each main state is denoted by index $i = 1, \dots, K$. The main states alone form a state machine identical to an HMM. The state transition probability between main states i is j denoted by a_{ij} and the initial state distribution by π_i. The state emission probability density functions (pdfs) b_i are Gaussian mixture models (GMMs)

$$b_i(\mathbf{o}_t) = \sum_{n=1}^{N} w_i^n \mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_i^n, \boldsymbol{\Sigma}_i^n), \tag{1}$$

where N is the number of Gaussians and w_i^n, $\boldsymbol{\mu}_i^n$, $\boldsymbol{\Sigma}_i^n$, are the weight, mean vector, and diagonal covariance matrix of the n^{th} mixture of state i, respectively. $\mathbf{o}_t$ is the observation vector in frame t.

The interpolating part of the model consists of interpolating states which form fixed-length state transition paths from a main state to another main state. Possible lengths are beforehand determined by the set of allowed lengths $\mathcal{D}$. For a specific length $d \in \mathcal{D}$ and main state combination $i \neq j$, interpolating states (i, j, d, τ), $\tau = 1, \dots, d-1$ are generated. τ is a indicator variable which determines the order of the interpolating state in the particular interpolating path. The set is chosen manually to balance the performance and computational complexity of the algorithm.

The initial probabilities of interpolating states are defined to be zero. The state transition probability from main state i to first interpolating state $(i, j, d, 1)$ of each duration d equals a_{ij}. In practise, this means that all interpolation lengths are modeled as equally probable. From interpolating state (i, j, d, τ) which is not the last state of interpolating state sequence $(\tau < d-1)$, we always move to the interpolating state $(i, j, d, \tau+1)$ with probability 1. From the last interpolating state $(i, j, d, d-1)$ in an interpolating state sequence we always move to main state j, thus terminating the interpolation state transition path. Figure 2 illustrates the interpolating states between two main states.

The emission probability of the interpolating state (i, j, d, τ) is also a GMM, the parameters of which are obtained by linear inter-

This work has been supported by the Academy of Finland.

polation of the parameters of the states i and j as

$$w^n_{(i,j,d,\tau)} = w^n_i \frac{\tau}{d} + w^n_j \frac{d-\tau}{d} \tag{2}$$

$$\boldsymbol{\mu}^n_{(i,j,d,\tau)} = \boldsymbol{\mu}^n_i \frac{\tau}{d} + \boldsymbol{\mu}^n_j \frac{d-\tau}{d} \tag{3}$$

$$\boldsymbol{\Sigma}^n_{(i,j,d,\tau)} = \boldsymbol{\Sigma}^n_i \frac{\tau}{d} + \boldsymbol{\Sigma}^n_j \frac{d-\tau}{d}. \tag{4}$$

The above implements linear trajectories of pdfs from main state i to main state j, having a length of $d-1$ states. Interpolations are not allowed from a main state to itself, so that the total number of interpolating states equals $K_{\text{int}} = K(K-1)\sum_{d\in\mathcal{D}}(d-1)$.

3. ESTIMATING THE MODEL PARAMETERS

With the interpolating states the proposed model is still a HMM, but it has a specific interpolating topology, and the parameters are shared between the states in a specific way. Therefore, it is advantageous to estimate the parameters while taking the topology into account. Given an observation sequence $\mathbf{O} = \mathbf{o}_0, \mathbf{o}_1, \ldots, \mathbf{o}_T$, we estimate the parameters of the model by the generalized expectation maximization algorithm. Similarly to the Baum-Welch algorithm, we calculate the state occupation probabilities and then estimate the model parameters by maximizing and auxiliary function derived from the state occupation probabilities.

The hidden variables in the training algorithm are q_t, the state at time t. We first use the forward-backward procedure of Baum [4, pp. 335-337] to calculate the forward variables $\alpha_t(i) = p(\mathbf{o}_0, \mathbf{o}_1, \ldots, \mathbf{o}_t, q_t = i|\lambda)$ and backward variables $\beta_t(i) = p(\mathbf{o}_{t+1}, \ldots, \mathbf{o}_T | q_t = i, \lambda)$. Here λ denotes the current parameters of the model. To simplify the notation, i can denote either a main state or an interpolating state.

The state occupation variable $\gamma_t(i) = p(q_t = i|\mathbf{O}, \lambda)$ is then calculated for the main states and the interpolating states as

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_i \alpha_t(i)\beta_t(i)}, \tag{5}$$

where the summation is done over all the main states and interpolating states. We also introduce variable $\xi_t(i,j) = p(q_t = i, q_{t+1} = j|\mathbf{O}, \lambda)$

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^K \sum_{j=1}^K \alpha_t(i) a_{ij} b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}. \tag{6}$$

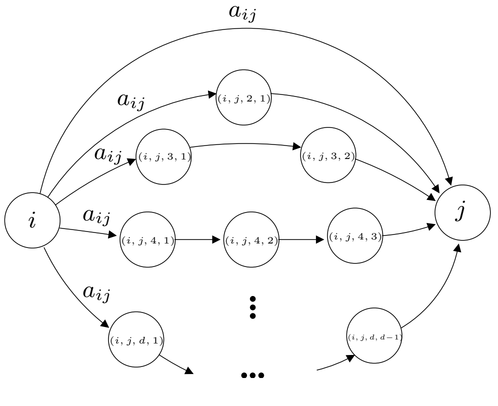

Fig. 2. *An illustration of two main states i and j, state transition path between them, and interpolating states (i,j,d,τ) generated between them. All the interpolation durations have an equal probability a_{ij}*

In Eqs. (5) and (6), i and j can denote either a main state or an interpolating state. It can be noticed that $\xi_t(i,(i,j,d,\tau))$ is non-zero only when $\tau = 1$. The auxiliary function

$$Q(\lambda', \lambda) = \sum_{\mathbf{q}} p(\mathbf{O}, \mathbf{q}|\lambda') \log p(\mathbf{O}, \mathbf{q}|\lambda) \tag{7}$$

used in maximization of the likelihood of the model parameters is identical to the one used in HMM training. λ' denotes the current parameters of the model, and the auxiliary function maximized with respect to new parameters λ. $\mathbf{q}$ denotes a state transition path, and the summation above is done over all possible state transition paths. Even though the auxiliary function is identical to HMM training, the parameter estimation is different since the parameters of the interpolating states are dependent on the main state parameters.

Similarly to the Baum-Welch algorithm, the auxiliary function can be split into three terms accounting for the initial state probabilities, the state transition probabilities, and state emission probabilities. Because the initial probabilities of interpolating states are zero and transitions from them are fixed, the factorization is

$$Q(\lambda', \lambda) = Q_\pi(\lambda', \lambda) + \sum_{i=1}^K Q_{a_i}(\lambda', \lambda) + Q_b(\lambda', \lambda), \tag{8}$$

where the terms Q_π, Q_a, and Q_b account for the the initial state probabilities, the state transition probabilities, and state emission probabilities, respectively, and are defined in the following sections.

3.1. Initial state probabilities and state transition probabilities

Since the initial state probabilities of the interpolating states are zero, the auxiliary term corresponding to the initial state probabilities is identical to the one in the Baum-Welch algorithm:

$$Q_\pi(\lambda', \lambda) = \sum_{i=1}^K \gamma_0(i) \log \pi_i.$$

With constraint $\sum_{i=1}^K \pi_i = 1$ the above is maximized with

$$\pi_i = \gamma_0(i). \tag{9}$$

Since transitions from a main state are allowed to main states but as well to first states of interpolation, the auxiliary term corresponding to transitions from state i is given as

$$Q_{a_i}(\lambda', \lambda) = \sum_{t=1}^T \sum_{j=1}^K [\xi_{t-1}(i,j) + \sum_{d\in\mathcal{D}} \xi_{t-1}(i,(i,j,d,1))] \log a_{ij}.$$

With constraint $\sum_{j=1}^K a_{ij} = 1$ the above is maximized with

$$a_{ij} = \frac{\sum_{t=1}^T [\xi_{t-1}(i,j) + \sum_{d\in\mathcal{D}} \xi_{t-1}(i,(i,j,d,1))]}{\sum_{t=1}^T \gamma_{t-1}(i)}. \tag{10}$$

3.2. Mixture weights

Similarly to HMMs with Gaussian mixture densities, we can model the densities by individual Gaussians by generating artificial states with state transition probabilities from the original state to the generated states being equal to the mixture weights [5]. We calculate the probability of being in state i at time t and the n^{th} mixture accounting for $\mathbf{o}_t$ as

$$\gamma^n_t(i) = \gamma_t(i) \frac{w^n_i \mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}^n_i, \boldsymbol{\Sigma}^n_i)}{\sum_{n'=1}^N w^{n'}_i \mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}^{n'}_i, \boldsymbol{\Sigma}^{n'}_i)}, \tag{11}$$

where i can denote either a main state or an interpolating state. The auxiliary function corresponding to the mixture weights is

$$Q(\lambda', \lambda)_w = \sum_{t=1}^{T} \sum_{i=1}^{K} \sum_{n=1}^{N} [\gamma_t^n(i) \log(w_i^n) +$$

$$\sum_{j \neq i} \sum_{d \in \mathcal{D}} \sum_{\tau=1}^{d-1} \gamma_t^n(i,j,d,\tau) \log(w_i^n \frac{\tau}{d} + w_j^n \frac{d-\tau}{d})], \quad (12)$$

where the interpolated weights (2) are explicitly written in the equation. To simplify the notation, let us write $\zeta_n(i) = \sum_{t=1}^{T} \gamma_t^n(i)$ and $\zeta_n(i,j,d,\tau) = \sum_{t=1}^{T} \gamma_t^n(i,j,d,\tau)$. With constraint $\sum_i w_i^n = 1$ the above is maximized with the solution

$$w'^n_i = \zeta^n(i) + \sum_{j \neq i} \sum_{d \in \mathcal{D}} \sum_{\tau=1}^{d-1} [\zeta^n(i,j,d,d-\tau) \frac{\tau}{d} + \zeta^n(i,j,d,\tau) \frac{d-\tau}{d}],$$

$$w_i^n = \frac{w'^n_i}{\sum_n w'^n_i}. \quad (13)$$

The solution can be verified with Lagrange multipliers, but it is here omitted because of space limitation constraints.

3.3. Variances

The auxiliary function corresponding to the emission pdfs of the n^{th} mixture is

$$Q(\lambda', \lambda)_{b_n} = -\frac{1}{2} \sum_{i=1}^{K} \sum_{t=1}^{T} [\gamma_t^n(i) \log(b_t^n(i))$$

$$+ \sum_{j \neq i} \sum_{d \in \mathcal{D}} \sum_{\tau=1}^{d-1} \gamma_t^n(i,j,d,\tau) \log(b_t^n(i,j,d,\tau))], \quad (14)$$

where $b_t^n(i) = \mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_i^n, \boldsymbol{\Sigma}_i^n)$ is the Gaussian distribution of the n^{th} mixture. Since we use diagonal covariance matrices, the means and variances of each feature can be updated independently. To simplify the notation, the equations in Sections 3.3 and 3.4 present the updates of an individual feature, but the feature index is omitted. Thus, let us denote an observed feature in frame t by o_t, the mean of a feature of state i and mixture n by μ_i^n and its variance by $\sigma_{i,n}^2$.

When the terms independent of the variances and means in Eq. (14) are denoted by L, we obtain

$$Q(\lambda', \lambda) = L - \frac{1}{2} \sum_{t=1}^{T} \sum_{n=1}^{N} \Big\{ \gamma_t^n(i) [\log(\sigma_{i,n}^2) + \frac{(o_t - \mu_i^n)^2}{\sigma_{i,n}^2}]$$

$$+ \sum_{j \neq i} \sum_{d \in \mathcal{D}} \sum_{\tau=1}^{d-1} \gamma_t^n(i,j,d,\tau) [\log(\sigma_{(i,j,d,\tau),n}^2) + \frac{(o_t - \mu_{(i,j,d,\tau)}^n)^2}{\sigma_{(i,j,d,\tau),n}^2}] \Big\}. \quad (15)$$

Since the variances of the interpolating states are weighted sums of main state variances, we do not have a method for direct maximization of the above with respect to the variances, but instead use an update which just increases the auxiliary function.

Each term in the above sum equals the Bregman divergence [6] with $\phi(x) = -\log(x)$, or the Itakura Saito distance between the terms $(o_t - \mu_i^n)^2$ and the corresponding variance $\sigma_{i,n}^2$, up to additive terms independent of the variances, weighted by the state occupation probability. For the maximization of the divergences there exists a multiplicative gradient descent update [7, 6], which has been found to decrease the divergence. The resulting update is

$$\sigma_{i,n}^2 \leftarrow \sigma_{i,n}^2 \frac{\rho_i^n + \sum_{d \in \mathcal{D}} \sum_{\tau=1}^{d-1} [\frac{\tau}{d} \rho_{(i,d,\tau)}^n + \frac{d-\tau}{d} \rho_{(i,d,d-\tau)}^n]}{\zeta_i^n \sigma_{i,n}^{-2} + \sum_{d \in \mathcal{D}} \sum_{\tau=1}^{d-1} [\frac{\tau}{d} \nu_{(i,d,\tau)}^n + \frac{d-\tau}{d} \nu_{(i,d,d-\tau)}^n]}, \quad (16)$$

where

$$\rho_i^n = \sum_{t=1}^{T} \gamma_i^n (o_t - \mu_i^n)^2 \sigma_{i,n}^{-4},$$

$$\rho_{(i,d,\tau)}^n = \sum_{j \neq i} \sum_{t=1}^{T} \gamma_{(i,j,d,\tau)}^n (o_t - \mu_{(i,j,d,\tau)}^n)^2 \sigma_{(i,j,d,\tau),n}^{-4},$$

$$\nu_{(i,d,\tau)}^n = \sum_{j \neq i} \zeta^n(i,j,d,\tau) \sigma_{(i,j,d,\tau),n}^{-2}.$$

3.4. Means

When the weighted means in Eq. (3) are substituted to (15) and terms independent on the state mean vectors are omitted, we obtain auxiliary function corresponding to the means of the n^{th} mixture as

$$Q(\lambda', \lambda)_{\mu_n} = -\frac{1}{2} \sum_{t=1}^{T} \sum_{i=1}^{K} [\frac{\gamma_t^n(i)}{\sigma_{i,n}^2} (o_t - \mu_i^n)^2$$

$$+ \sum_{j \neq i} \sum_{d \in \mathcal{D}} \sum_{\tau=1}^{d-1} \frac{\gamma_t^n(i,j,d,\tau)}{\sigma_{(i,j,d,\tau),n}^2} (o_t - \frac{\tau}{d} \mu_i^n - \frac{d-\tau}{d} \mu_i^n)^2].$$

For each feature and mixture combination, the above can be viewed as a weighted linear least-squares problem with $KT(1 + (1-K)\sum_{d \in \mathcal{D}}(d-1))$ equations and K unknown variables. The global minimum can be solved by setting the derivative with respect to the means to zero which leads to the normal equations, but it is impractical to write the coefficients and weights explicitly using matrices. However, the resulting solution can be summarized as follows. For each mixture n, let us write the weighted coefficient matrix multiplied by its transpose by K x K matrix $\mathbf{H}$ where the diagonal entries are defined as

$$\mathbf{H}_{i,i}^n = \frac{\zeta^n(i)}{\sigma_{i,n}^2} + \sum_{j \neq i} \sum_{d \in \mathcal{D}} \sum_{\tau=1}^{d-1} \Big[\frac{(d-\tau)^2 \zeta^n(i,j,d,\tau)}{d^2 \sigma_{(i,j,d,\tau),n}^2} + \frac{\tau^2 \zeta^n(i,j,d,d-\tau)}{d^2 \sigma_{(i,j,d,d-\tau),n}^2} \Big]$$

and the non-diagonal entries as

$$\mathbf{H}_{i,j}^n = \sum_{j \neq i} \sum_{d \in \mathcal{D}} \sum_{\tau=1}^{d-1} \frac{\tau(d-\tau)\zeta^n(i,j,d,\tau)}{d^2 \sigma_{(i,j,d,\tau),n}^2}$$

Furthermore, let us write the weighted coefficient matrix multiplied by the observation vector as a vector $\mathbf{g}^n$ having entries

$$\mathbf{g}_i^n = \sum_{t=1}^{T} o_t \Big[\frac{\gamma_t(i)}{\sigma_{i,n}^2} + \sum_{j \neq i} \sum_{d \in \mathcal{D}} \sum_{\tau=1}^{d-1} \frac{(d-\tau)\gamma_t(i,j,d,\tau) + \tau\gamma_t(i,j,d,d-\tau)}{d \sigma_{(i,j,d,\tau),n}^2} \Big].$$

The optimal mean vector can be solved as

$$\boldsymbol{\mu}^n = (\mathbf{H}^n)^{-1} \mathbf{g}^n, \quad (17)$$

where the i^{th} entry of $\boldsymbol{\mu}^n$ is the mean of i^{th} state and n^{th} mixture.

3.5. Implementation issues

Like HMMs, the algorithm is sensitive to the initial parameters. We used to following initial parameters. All the state transition probabilities are $1/(K|\mathcal{D}|)$, where $|\mathcal{D}|$ is the number of possible interpolation lengths. All the initial state probabilities are set to $1/K$. The means and variances are obtained by k-means clustering the observation vectors.

In each iteration, the occupation probabilities are calculated according to Eqs. (5), (6), and (11). While keeping them fixed, the parameters are updated according to Eqs. (9), (10), (13), (16), and (17). Even though simultaneous update of means and variance is not guaranteed to increase the joint auxiliary function, in practise it was found to produce good results. It is advantageous to restrict the variances above a minimum threshold after each iteration. In our implementation we first train the model using a single Gaussian and then use mixture splitting to train GMMs. It can be noticed that when interpolation is not allowed ($\mathcal{D} = \emptyset$), the model reduces to a basic HMM and the training algorithm to the Baum-Welch algorithm for continuous mixture densities. The computational complexity of the training algorithm is approximately $Z = K \sum_{d \in \mathcal{D}} (d-1)$ times higher than training a HMM with an equal number states using the Baum-Welch algorithm, because the probabilities has to be accumulated over Z times higher number of states.

When multiple observation sequences are used to train the model, the reestimation formulas are first modified by normalizing the forward and backward coefficients as presented in [4, pp. 369-370]. The dividends and divisors in Equations (9), (10), (13), and (16) are calculated for each sequence, and the update is the quotient of the summed terms. In the case of means, the matrix $\mathbf{H}^n$ and vector $\mathbf{g}^n$ in (17) is calculated for each observation sequence and then summed, after which the sum $\mathbf{H}^n$ is inverted and multiplied by the sum $\mathbf{g}^n$ to obtain the means.

4. DECODING

Once the models have been trained, the model can be viewed as a normal HMM, so that the most likely state transition path and its likelihood can be calculated using the Viterbi algorithm. The state emission probabilities has to be calculated for the K main states and the K_{int} interpolating states. Per each observation, there are in total K^2 transitions from main states to main states, and $K(K-1)|\mathcal{D}|$ transitions from main states to interpolating states. The total number of transitions from interpolating states equals K_{int}. The computational complexity of the Viterbi decoding of the proposed model is approximately K_{int} times higher than basic HMM Viterbi decoding.

5. APPLICATION TO AUTOMATIC INSTRUMENT RECOGNITION

The proposed method was validated in automatic instrument recognition task, in which HMMs have been found to produce good results [8]. The acoustic data was isolated musical instrument note samples selected from the McGill University Master Samples collection, University of Iowa sample collection, IRCAM's Studio Online, and Real World Computing database. Eight instrument classes were selected to be used in the evaluations (piano, electric piano, acoustic guitar, electric guitar, electric bass, saxophone, oboe, flute). The instrument instances were randomized evenly into training and testing. From these instrument instances, twelve scales played with varying dynamics and style were randomly selected for both sets. In order to keep the training time reasonable, the training set was reduced by selecting only every fifth note from the scales. A total of 1108 individual note samples were used in training and 5278 in testing. Mel-frequency cepstral coefficients and their first time derivatives were used as features. Features were projected linearly to a base with maximal statistical independence using independent component analysis.

In addition to the proposed system, we used a baseline HMM classifier in the evaluation. A model was trained for each instrument class and in the classification stage, the Viterbi algorithm was applied to find the most likely state sequence and instrument class.

The baseline HMM system and the proposed IHMM system both utilized a fully connected topology between the main states. The training algorithm was initialized identically in both systems and the maximum number of iterations in the Baum-Welch training algorithm was limited to ten. The IHMM system was tested with a duration length set $\mathcal{D} = \{5, 10, 15\}$ which was found to produce good results in initial experiments.

	$K = 3$	$K = 4$	$K = 6$	$K = 12$	$K = 16$
method	$N = 4$	$N = 3$	$N = 2$	$N = 1$	$N = 1$
HMM	73.2%	75.4%	76.8%	78.3%	74.4%
IHMM	79.1%	**80.1%**	78.5%	79.1%	79.3%

Table 1. Recognition accuracy of the evaluated methods as a function of the number of states K and the number of mixtures N.

The recognition accuracies as a function of the number of states and Gaussians are given in Table 1. The IHMM gives systematically better results than the HMM. Increasing the number of states up to 12 states improves the performance of the HMM recognizer, but a larger number states decreases its accuracy. The IHMM algorithm is able to produce a good performance with a significantly smaller number of states. Analysis of effects of the duration length set and other parameters of the algorithm are topics for a future research.

6. CONCLUSIONS

We have proposed and interpolating extension to HMMs, which overcomes the limitation of HMM state independence by interpolating the emission pdfs linearly between states. We have presented an algorithm for training the parameters of the proposed model. In automatic instrument recognition study the proposed method is shown to outperform the baseline HMM classifier.

7. REFERENCES

[1] K.C. Sim and M.J.F. Gales, "Discriminative semi-parametric trajectory model for speech recognition," *Computer Speech and Language*, vol. 21, no. 4, 2007.

[2] A.-V. I. Rosti, *Linear Gaussian Models for Speech Recognition*, Ph.D. thesis, University of Cambridge, 2004.

[3] A. Klapuri, T. Virtanen, and M. Helén, "Modeling musical sounds with an interpolating state model," in *Proceedings of European Signal Processing Conference*, Istanbul, Turkey, 2005.

[4] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, New Jersey, 1993.

[5] B.H. Juang, S.E. Levinson, and M.M. Sondhi, "Maximum likelihood estimation for multivariate observations of Markov chains," *IEEE Transactions on Information Theory*, vol. 32, no. 2, 1986.

[6] I.S. Dhillon and S. Sra, "Generalized nonnegative matrix approximations with Bregman divergences," in *Proc. of Neural Information Processing Systems*, Vancouver, Canada, 2005.

[7] N. Bertin C. Févotte and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence. with application to music analysis," *Neural Computation*, vol. 21, no. 3, 2009.

[8] A. Eronen, "Musical instrument recognition using ICA-based transform of features and discriminatively trained HMMs," in *Proceedings of Seventh International Symposium on Signal Processing and Its Applications*, 2003.