MULTIPLE MIXTURE SEGMENTAL HMM AND ITS APPLICATIONS

Bing Xiang, Toby Berger

School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, 14853
Email: {bxiang, berger}@ee.cornell.edu

ABSTRACT

In this paper a multiple mixture segmental hidden Markov model (MMSHMM) is presented. This model is extended from the linear probabilistic-trajectory segmental HMM [1]. Each segment is characterized by linear trajectory with slope and mid-point parameters, and also the residual error covariances around the trajectory, so that both extra-segmental and intra-segmental variation are represented. Instead of modeling single distribution for each model parameter as earlier work, we use multiple mixture components for model parameters to represent the variability due to the variation within each speaker and also the differences between speakers. This model is evaluated on two applications. One is a phonetic classification task with TIMIT corpus, which shows that MMSHMM has advantages over conventional HMM. Another one is a speaker-independent keyword spotting task with the Road Rally database. By rescoring putative events hypothesized by a primary HMM keyword spotter, the experiments show that the performance is improved through distinguishing true hits from false alarms.

1. INTRODUCTION

Whether or not being able to accurately modeling the dynamics in human speech is one of the important factors affecting the performance of a speech recognition system. In recent years, segmental modeling becomes an interesting and attractive area. A variety of methods were introduced in the past [2, 3]. It has been shown that segmental model has some advantages over conventional HMM, because HMM takes no account of the dynamic constraints of the speech production system. In segmental model, time-correlation between neighboring speech frames are modeled through different ways. Linear probabilistic-trajectory segmental HMM [1, 4] is one of these approaches. Each speech segment is represented by a set of statistics which includes a linear trajectory and residual error around the trajectory. The extra-segmental variability between different examples of a speech segment is modeled separately from the intra-segmental variability within the segment. Single distribution for each model parameter is used in the earlier work. However, to better represent the occasion-to-occasion variation for any one speaker and also differences between speakers, we believe that using multiple mixture components for model parameters in each segment can improve the performance of speech recognition system. It's analogous to using mixtures of Gaussians to represent output probability distributions for HMMs. Although some work already demonstrated advantages from including multiple mixture components in parametric trajectory segmental models [5, 6] and non-parametric trajectory segmental models [7], including multiple mixture components into probabilistic-trajectory segmental model has not been shown in literature. In this paper, a linear probabilistic-trajectory multiple mixture segmental HMM (MMSHMM) is presented and evaluated on a phonetic classification task with TIMIT and also on a speaker-independent keyword spotting task with Road Rally database. The experiments show that MMSHMM has advantages over conventional HMMs.

The rest of the paper is organized in this way. The representation and estimation of MMSHMM is presented in section 2. In Section 3, we introduce the applications of MMSHMM on phonetic classification and keyword spotting. Experiment results are presented in section 4. Section 5 contains conclusion.

2. MULTIPLE MIXTURE SEGMENTAL HMM

2.1. Model representation

A linear trajectory segmental HMM assumes that the underlying trajectory vector change linearly over time within each segment. Suppose the segmental model M has N states, i.e. segments, and $x = x_1, ..., x_N$ is a state sequence. Let the observation sequence $\vec{y_i} = y_{t_{x,i}}, ..., y_{t_{x,i+1}-1}$ correspond to state x_i, then the linear trajectory $f_{m,s}$ can be defined by the segment mid-point value m and slope s, such that $f_{m,s}(t) = m + s(t - \frac{t_{x,i}+t_{x,i+1}-1}{2})$. In a least-squared error sense, the mid-point and slope values which best fit to such particular sequence of observations $\vec{y_i}$ are given by

$$m'_{x,i}(\vec{y_i}) = \frac{\sum_{t=t_{x,i}}^{t_{x,i+1}-1} y_t}{T_{x,i}}, \qquad (1)$$

and

$$s'_{x,i}(\vec{y_i}) = \frac{\sum_{t=t_{x,i}}^{t_{x,i+1}-1}(t - \frac{t_{x,i}+t_{x,i+1}-1}{2})y_t}{\sum_{t=t_{x,i}}^{t_{x,i+1}-1}(t - \frac{t_{x,i}+t_{x,i+1}-1}{2})^2}, \qquad (2)$$

where $T_{x,i} = t_{x,i+1} - t_{x,i}$.

The previously introduced joint probability of $\vec{y_i}$ and trajectory parameters is

$$\begin{aligned} p(\vec{y_i}, m, s | M, x) &= N_{\mu_i,\eta_i}(m) N_{\nu_i,\xi_i}(s) \\ &\times \prod_{t=t_{x,i}}^{t_{x,i+1}-1} N_{f_{m,s}(t),\sigma_i}(y_t), \end{aligned} \qquad (3)$$

i.e., the distributions of m, s and y_t are defined by Gaussian distribution N_{μ_i,η_i}, N_{ν_i,ξ_i} and $N_{f_{m,s}(t),\sigma_i}$ respectively. All distributions are assumed to have diagonal covariance matrices. And for simplicity all observation sequences are assumed to be one-dimensional.

To better represent the variation within different examples of one speaker and also the differences between different speakers, we can extend Eq.(3) to multiple mixture components as

$$P(\vec{y_i}, m, s|M, x) = \sum_{k=1}^{K} w_{i,k} p_k(\vec{y_i}, m, s|M, x), \tag{4}$$

where $w_{i,k}$ are weights which satisfy $\sum_{k=1}^{K} w_{i,k} = 1$ for each $i = 1, ..., N$, and

$$\begin{aligned} p_k(\vec{y_i}, m, s|M, x) &= N_{\mu_{i,k},\eta_{i,k}}(m) N_{\nu_{i,k},\xi_{i,k}}(s) \\ \times & \prod_{t=t_{x,i}}^{t_{x,i+1}-1} N_{f_{m,s}(t),\sigma_{i,k}}(y_t). \end{aligned} \tag{5}$$

So $\mu_{i,k}$, $\eta_{i,k}$, $\nu_{i,k}$, $\xi_{i,k}$, $\sigma_{i,k}$ and $w_{i,k}$ are the model parameters corresponding to the kth mixture component in the ith state. The output probability given model parameters and state sequence can be calculated by integrating $P(\vec{y_i}, m, s|M, x)$ over the unknown trajectory parameters m and s, as

$$\begin{aligned} P(\vec{y_i}|M, x) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P(\vec{y_i}, m, s|M, x) dmds \\ &= \sum_{k=1}^{K} w_{i,k} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_k(\vec{y_i}, m, s|M, x) dmds. \end{aligned} \tag{6}$$

Similar with [1], we can get the trajectory-independent output probability as

$$P(\vec{y_i}|M, x) = \sum_{k=1}^{K} w_{i,k} p_k(\vec{y_i}|M, x), \tag{7}$$

where

$$\begin{aligned} p_k & \quad (\vec{y_i}|M, x) = (\frac{\sigma_{i,k}}{T_{x,i}\eta_{i,k} + \sigma_{i,k}})^{1/2} (\frac{\sigma_{i,k}}{q_{x,i}\xi_{i,k} + \sigma_{i,k}})^{1/2} \\ \times & \quad (\frac{1}{2\pi\sigma_{i,k}})^{\frac{T_{x,i}}{2}} exp(-\frac{1}{2}(\frac{T_{x,i}(\mu_{i,k} - m'_{x,i}(\vec{y_i}))^2}{T_{x,i}\eta_{i,k} + \sigma_{i,k}})) \\ \times & \quad exp(-\frac{1}{2}(\frac{q_{x,i}(\nu_{i,k} - s'_{x,i}(\vec{y_i}))^2}{q_{x,i}\xi_{i,k} + \sigma_{i,k}})) \\ \times & \quad exp(-\frac{1}{2\sigma_{i,k}}(\sum_{t=t_{x,i}}^{t_{x,i+1}-1} y_t^2 - T_{x,i}(m'_{x,i}(\vec{y_i}))^2 \\ - & \quad q_{x,i}(s'_{x,i}(\vec{y_i}))^2)) \end{aligned} \tag{8}$$

and $q_{x,i} = \sum_{t=t_{x,i}}^{t_{x,i+1}-1} (t - \frac{t_{x,i}+t_{x,i+1}-1}{2})^2$.

2.2. Model initialization

To train the model, we need to initialize the model parameters first. As pointed out in other literature, the initialization will largely affect the performance of segmental model. Based on some initial experiments, we choose the following way for model initialization. First, the segmentation can be obtained using Viterbi alignment with trained HMMs. Then the means and variances of the mid-points can be initialized from the distribution of the individual mid-points for each segment. Since we model multiple mixture components for each segment, the k-means clustering algorithm can be used here to get K mixture components distributions. Then the weights can be set to proportional to the number of examples belonging to each component. The variance of the observations around each individual trajectory can be obtained by calculating the residual errors. The means and variances of the slope are set to zero. In such way, the slope is constrained in the initialization.

2.3. Model re-estimation

Following the approach of [8], the re-estimation formulae for MMSHMM parameters can be derived by introducing an auxiliary function and finding new values for the model parameters which can maximize that auxiliary function. Such auxiliary function can be created as

$$Q(M, \bar{M}) = \sum_{x \in S} P(\vec{y}, x|M) log P(\vec{y}, x|\bar{M}), \tag{9}$$

where S is the set of possible state sequences. To simplify notation, we assume the state transition probability matrix A is strictly left-to-right, such that $a_{ij} = 1$ if $j = i + 1$ and all other transition probabilities are equal to zero. The auxiliary function can be derived as

$$\begin{aligned} Q & \quad (M, \bar{M}) = \sum_{x \in S} P(\vec{y}, x|M) log(p(x|\bar{M}) \prod_{i=1}^{N} P(\vec{y_i}|\bar{M}, x)) \\ = & \quad \sum_{i=1}^{N} \sum_{x \in S} P(\vec{y}, x|M) log(\sum_{k=1}^{K} \bar{w}_{i,k} p_k(\vec{y_i}|\bar{M}, x)) \\ + & \quad \sum_{x \in S} P(\vec{y}, x|M) log p(x|\bar{M}). \end{aligned} \tag{10}$$

Then by differentiating $Q(M, \bar{M})$ with respect to each model parameter $\bar{\mu}_{i,k}$, $\bar{\eta}_{i,k}$, $\bar{\nu}_{i,k}$, $\bar{\xi}_{i,k}$, and $\bar{\sigma}_{i,k}$, and make the derivatives equal to zero, we can get the re-estimation formulae for these model parameters as follows:

$$\bar{\mu}_{i,k} = \frac{\sum_{x \in S} R_{x,i,k} \lambda_{x,i,k} T_{x,i} m'_{x,i}(\vec{y_i})}{\sum_{x \in S} R_{x,i,k} \lambda_{x,i,k} T_{x,i}}, \tag{11}$$

$$\bar{\eta}_{i,k} = \frac{\sum_{x \in S} R_{x,i,k} \lambda_{x,i,k}^2 T_{x,i}^2 \bar{\eta}_{i,k} (\bar{\mu}_{i,k} - m'_{x,i}(\vec{y_i}))^2}{\sum_{x \in S} R_{x,i,k} \lambda_{x,i,k} T_{x,i}}, \tag{12}$$

$$\bar{\nu}_{i,k} = \frac{\sum_{x \in S} R_{x,i,k} \kappa_{x,i,k} q_{x,i} s'_{x,i}(\vec{y_i})}{\sum_{x \in S} R_{x,i,k} \kappa_{x,i,k} q_{x,i}}, \tag{13}$$

$$\bar{\xi}_{i,k} = \frac{\sum_{x \in S} R_{x,i,k} \kappa_{x,i,k}^2 q_{x,i}^2 \bar{\xi}_{i,k} (\bar{\nu}_{i,k} - s'_{x,i}(\vec{y_i}))^2}{\sum_{x \in S} R_{x,i,k} \kappa_{x,i,k} q_{x,i}}, \tag{14}$$

$$\bar{\sigma}_{i,k} = \frac{\sum_{x \in S} R_{x,i,k} \alpha_{x,i,k}}{\sum_{x \in S} R_{x,i,k} (T_{x,i}(1 - \lambda_{x,i,k} \bar{\eta}_{i,k}) - \kappa_{x,i,k} q_{x,i} \bar{\xi}_{i,k})}, \tag{15}$$

where

$$R_{x,i,k} = \frac{P(\vec{y}, x|M) p_k(\vec{y_i}|\bar{M}, x)}{\sum_{k=1}^{K} \bar{w}_{i,k} p_k(\vec{y_i}|\bar{M}, x)}, \tag{16}$$

$$\lambda_{x,i,k} = \frac{1}{T_{x,i} \bar{\eta}_{i,k} + \bar{\sigma}_{i,k}}, \tag{17}$$

$$\kappa_{x,i,k} = \frac{1}{q_{x,i} \bar{\xi}_{i,k} + \bar{\sigma}_{i,k}}, \tag{18}$$

$$\begin{aligned} \alpha_{x,i,k} &= \lambda_{x,i,k}^2 T_{x,i} \bar{\sigma}_{i,k}^2 (\bar{\mu}_{i,k} - m'_{x,i}(\vec{y_i}))^2 \\ + & \quad \kappa_{x,i,k}^2 q_{x,i} \bar{\sigma}_{i,k}^2 (\bar{\nu}_{i,k} - s'_{x,i}(\vec{y_i}))^2 \\ + & \quad \sum_{t=t_{x,i}}^{t_{x,i+1}-1} y_t^2 - T_{x,i}(m'_{x,i}(\vec{y_i}))^2 - q_{x,i}(s'_{x,i}(\vec{y_i}))^2, \end{aligned} \tag{19}$$

and $i = 1, ..., N,\ k = 1, ..., K$.

For the mixture component weights w, since there is an extra constraint that $\sum_{k=1}^{K} w_{i,k} = 1$ for each $i = 1, ..., N$, we can use Lagrange multiplier to get its re-estimation formula as:

$$\bar{w}_{i,k} = \frac{\sum_{x \in S} R_{x,i,k} \bar{w}_{i,k}}{\sum_{x \in S} P(\vec{y}, x|M)}, \tag{20}$$

where $i = 1, ..., N,\ k = 1, ..., K$.

We can see that the right hand sides of the above formulae include re-estimated parameters. In practice, we can replace the re-estimated values of $\bar{\mu}_{i,k},\ \bar{\eta}_{i,k},\ \bar{\nu}_{i,k},\ \bar{\xi}_{i,k},\ \bar{\sigma}_{i,k}$ and $\bar{w}_{i,k}$ on the right hand sides with original values respectively. Experiments in section 4.1 show that such changes won't prevent the increase of $P(\vec{y}|M)$. Usually five iterations of re-estimation are run during the training session of MMSHMM.

3. APPLICATIONS OF MMSHMM

In this paper two different applications are used to evaluate this segmental model. The first is the classification of phones in American English. Another one is to rescore the putative events hypothesized by a primary HMM keyword spotter to discriminate true hits from false alarms.

3.1. Phonetic classification

The phonetic classification is to determine the identity of speech segments with specified phonetic boundaries. Classification has computational advantages over recognition. So it can give quick feedback about the performance of MMSHMM.

Each segmental model corresponds to a phone and has three states. And each state has one or three mixture components. First, the MMSHMM parameters μ, η and σ can be initialized based on conventional HMMs with three states each model. If each segment has K mixture components, through the clustering algorithm, these initial parameters can be generated for each component. The initial parameters of ν and ξ are set to be zero. Then five iterations of re-estimation are run to re-estimate these parameters. After training, these models can be used to classify the tokens from all phones in the testing set.

3.2. Keyword spotting

MMSHMM can also be applied to keyword spotting to serve as second processing such as that in [9]. In keyword spotting, the keywords that are of interest to the system are to be spotted and the irrelevant sounds are to be rejected. There have been a variety of approaches taken to solve this problem during last two decades [10].

We use a primary HMM word spotter to get the putative events first. This HMM word spotter is composed of a parallel network of both keyword and non-keyword(filler) models. Using a null-grammar, frame-synchronous network search algorithm [11], we can get the sequence of keywords and fillers as putative events. Each HMM in the network is a continuous Gaussian mixture model. The confidence measure used in primary word spotter is the log likelihood ratio between the probability of the obervations that came from the keyword model, and the probability that came from the filler network.

In the second processing, two MMSHMMs can be generated for each keyword using labeled putative events. One is for the segments from the true keywords, and the other is for segments from false alarms. The initialization of MMSHMM is the same as that in phonetic classification. After initialization, five iterations of re-estimation are run for each model. Once these models for each particular keyword are trained, they are used to rescore new putative events of that keyword. The secondary score is calculated as the log likelihood ratio between the probability from the truth model and the probability from the false alarm model. To obtain the final score to reorder the putative events for each keyword, the new secondary scores and original primary HMM scores are normalized separately and then summed together.

4. EXPERIMENTS

The feature used for both of the experiments is a feature vector with 26 dimension for each frame, which includes log energy, 12 mel cepstral coefficients and their delta coefficients.

4.1. Phonetic classification

The data used for phonetic classification task is the TIMIT acoustic-phonetic continuous speech database of American English. The training set includes 3260 utterances from 326 male speakers with 10 sentences for each speaker. The testing set includes 1120 utterances of 112 male speakers. During testing, a 39-category scoring set is used as [12]. Language model and duration are not considered in this experiment.

As introduced in section 2.3, during re-estimation, the model parameters on the right hands of those formulae are replaced by original values in practice. To test its reasonability, we monitored the changing of the log probabilities of the training set during the five re-estimation iterations. As shown in Fig. 1, the probabilities kept increasing during the training for both the single mixture component MMSHMM ($K = 1$) and three mixture components MMSHMM ($K = 3$). This ensured the reliability of the following experiment results.

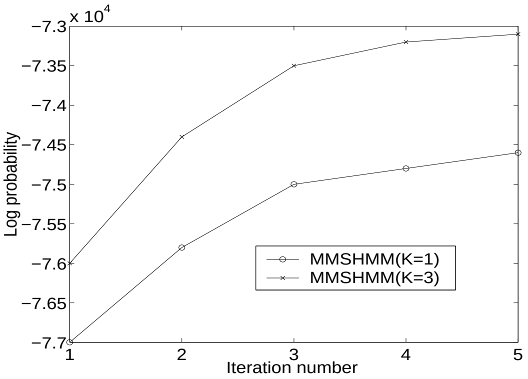

Figure 1: Log probability of phone training set vs. number of training iterations

Experiment results for phonetic classification are shown in Table 1. The result of continuous HMM with three states is also given. In addition to phoneme classification, a 16 vowels classification task is also tried. And different performances from HMM, MMSHMM with $K = 1$ and MMSHMM with $K = 3$ are compared.

	HMM	MMSHMM(K=1)	MMSHMM(K=3)
phoneme	61.4%	62.3%	63.8%
vowel	57.5%	58.0%	59.1%

Table 1: Phone classification and vowel classification results on TIMIT (K is the number of mixture components)

From the results we can see that MMSHMM has advantages over the conventional HMM. It proves the importance of representing dynamics between neighboring speech frames. With one mixture component, MMSHMM is just simplified to the normal linear trajectory segmental HMM (SHMM). The results show that with three mixture components for each segment, MMSHMM is better than the linear trajectory SHMM for both phoneme classification and vowel classification.

4.2. Keyword spotting

The keyword spotting task is evaluated on the Road Rally database, which has twenty keywords designated. This database includes two data corpus, the Waterloo corpus and Stonehenge corpus. The marked keyword occurrences from the read paragraph speech of the 28 male speakers in Waterloo corpus are used to train the 20 keyword HMMs. Each HMM has 10 states, and 9 mixtures for each state. For each keyword there are 84 to 258 tokens for training. And the non-keyword speech in the Waterloo are used to train 20 filler models. These 40 HMMs compose the network in the word spotter. The conversational speech from 10 male speakers in the Stonehenge corpus is used for training 40 MMSHMMs, with two models corresponding to one keyword. The number of states in MMSHMM is also chosen as 10. The speech of 10 other male speakers are used for testing and comparing HMM and MMSHMM.

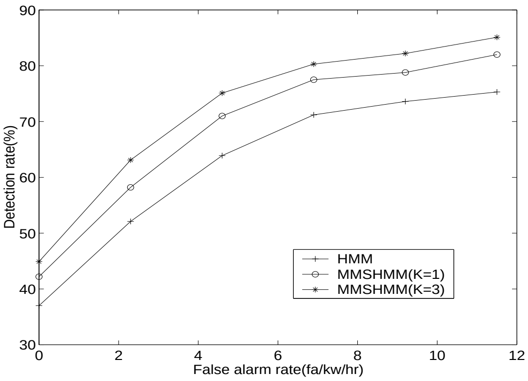

Figure 2: ROC performance curve of HMM and MMSHMM

The ROC performance curve of the primary HMM keyword spotter and MMSHMM is shown in Fig. 2. According to the NIST standard, composite detection rate for all twenty keywords is computed and plotted against the false alarm rate in false alarm per keyword per hour (fa/kw/hr). We can see that with MMSHMM as secondary processor to rescore putative events, the performance of keyword spotter is improved. The Figure of Merit(FOM), which is the average detection rate from 0 to 10 fa/kw/hr, is shown in Table 2. The FOM improves from 60.4% (HMM) to 66.2% (MMSHMM with 1 mixture), then to 70.2% (MMSHMM with 3 mixtures).

HMM	MMSHMM(K=1)	MMSHMM(K=3)
60.4%	66.2%	70.2%

Table 2: FOM of HMM and MMSHMM

5. CONCLUSION

In this paper we presented a linear trajectory multiple mixture segmental HMM. This model is evaluated on a TIMIT phonetic classification task and a keyword spotting task. Under both circumstances, it shows advantages over the conventional HMM and gives performance comparable to those reported by other groups. Of cause, it has its own drawback, such as relatively expensive computation. How to reduce this and still keep good performance would be one of the future work. Moreover, modeling the dynamics across the segments may also improve the performance of current model.

6. REFERENCES

[1] W.J.Holmes and M.J.Russell, "Probabilistic-trajectory segmental HMMs", Computer Speech and Language, vol.13, pp.3-37, 1999.

[2] M.Ostendorf, V.V.Digalakis and O.A.Kimball, "From HMM's to segmental models: a unified view of stochastic modeling for speech recognition", IEEE Trans. Speech and Audio Processing, vol.4, no.5, pp.360-378, 1996.

[3] M.J.F.Gales and S.J.Young, "Segmental hidden Markov models", Proc. of European Conf. on speech Comm. and Tech., pp.1579-1582, 1993.

[4] W.J.Holmes and M.J.Russell, "Linear dynamic segmental HMMs: variability representation and training procedure", ICASSP, pp.1399-1402, 1997.

[5] H.Gish and K.Ng, "Parametric trajectory models for speech recognition", ICSLP, pp.466-469, 1996.

[6] L.Deng and M.Aksmanovic, "Speaker-independent phonetic classification using hidden Markov models with mixtures of trend functions", IEEE Trans. Speech and Audio Processing, vol.5, pp.319-324, 1997.

[7] Y.Gong and J.P.Haton, "Stochastic trajectory modelling for speech recognition", ICASSP, pp.57-60, 1994.

[8] L.A.Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources", IEEE Trans. Information Theory, vol.28, pp.729-734, 1982.

[9] H.Gish and K.Ng, "A segmental speech model with applications to word spotting", ICASSP, pp.447-450, 1993.

[10] R.C.Rose, "Keyword detection in conversational speech utterances using hidden Markov model based continuous speech recognition", Computer Speech and Language, vol.9, pp.309-333, 1995.

[11] C.H.Lee and L.R.Rabiner, "A frame-synchronous network search algorithm for connected word recognition", IEEE Trans. ASSP, vol.37, no.11, pp.1649-1658, 1989.

[12] K.F.Lee and H.W.Hon, "Speaker-independent phone recognition using hidden Markov models", IEEE Trans. ASSP, vol.37, pp.1641-1648, 1989.