STRANDED GAUSSIAN MIXTURE HIDDEN MARKOV MODELS FOR ROBUST SPEECH RECOGNITION

Yong Zhao and Biing-Hwang (Fred) Juang

Department of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, USA

ABSTRACT

Gaussian mixture (GMM)-HMMs, though being the predominant modeling technique for speech recognition, are often criticized as being inaccurate to model heterogeneous data sources. In this work, we propose the stranded Gaussian mixture (SGMM)-HMM, an extension of the GMM-HMM, to explicitly model the dependence among the mixture components, i.e., each mixture component is assumed to depend on the previous mixture component in addition to the state that generates it. In the evaluation over the Aurora 2 database, the proposed 20-mixture SGMM system obtains WER of 8.07%, 10% relative improvement over the baseline GMM system. The experiments demonstrate the discriminating power that would be possessed by the mixture weights in their advanced form.

Index Terms: Dynamic Bayesian network, Gaussian mixture model, hidden Markov model, robust speech recognition

1. INTRODUCTION

State-of-the-art speech recognition systems assume the availability of tremendous speech data to achieve accurate and robust recognition performance. Efficient modeling techniques that are highly scalable to the data volume consist of N-gram language models to maintain accurate word prediction, context-dependent phoneme models to represent pronunciation variations, and multiple mixtures of Gaussians to account for extraneous non-speech variabilities. Among them, the last approach is often criticized as being inaccurate to model heterogeneous data sources: the mixture components that are obtained in different acoustic conditions for one sound can be concatenated to match at a high probability with the speech observations from another sound, a problem referred to as trajectory folding [1].

One approach to improve the modeling accuracy is to relax the HMM conditional-independence assumption, and condition the distribution of each observation on the previous observations in addition to the state that generates it [2], [3]. This method is known as conditional Gaussian HMMs or autoregressive HMMs. However, it has been shown that the conditional Gaussian HMMs often do not provide a benefit if the dynamic features are used [4], [3]. Another class of methods explores the use of more complex HMM structures, such as multiple-path modeling [5], [6]. This model is composed of multiple parallel paths, each of which may account for the acoustic variability from a specific source. The multiple-path model may over-correct the trajectory folding problem associated with the GMM-HMM, as the allowable mixture paths are exponentially reduced. Most of such systems have been only evaluated on some simple recognition tasks using a small number of parallel paths. How to achieve a model that is intrinsically robust to speaker and environmental changes is still a challenging and interesting problem, though we have observed less efforts being attempted along this direction in recent years.

In this paper, we propose a stranded Gaussian mixture (SGMM)-HMM, an extension of the GMM-HMM, to explicitly model the dependence among the mixture components. In other words, each mixture component is assumed to depend on the previous mixture component in addition to the state that generates it. Another motivation for the SGMM model comes from the hope to make use of the discriminating power that would be possessed by the mixture weights, which have now evolved as the mixture transition probabilities. The SGMM model implicitly relaxes the HMM conditional-independence assumption for observations. Also, the model contains the multiple-path models as special cases by properly setting the mixture transition matrices. The effectiveness of the proposed model is evaluated on the Aurora 2 database.

The remainder of the paper is organized as follows. In Section 2, we describe the structure of the SGMM-HMM model and the associated learning and decoding algorithms. We present the experimental results and conclusions in Section 3 and Section 4, respectively.

2. STRANDED GAUSSIAN MIXTURE HMMS

As opposed to the regular GMM-HMM, the SGMM-HMM aims to explicitly model the relationships among the mixture components, that is, the distribution of the mixture component is assumed to depend on the previous mixture component in addition to the state that generates it. The model can be represented by a dynamic Bayesian network (DBN) [7] as shown in Fig. 1. Note that additional links between successive mixture variables are added in comparison with the GMM-HMM. Let $\boldsymbol{x}_1^T = \boldsymbol{x}_1, ..., \boldsymbol{x}_T$ be a sequence of observations of length T, and $s_1^T = s_1, ..., s_T$ and $m_1^T = m_1, ..., m_T$ are the hypothesized state and mixture sequences, respectively. The joint probability of the three sequences in the SGMM model is given by

$$p(\boldsymbol{x}_1^T, s_1^T, m_1^T) = \prod_{t=1}^{T} p(\boldsymbol{x}_t|s_t, m_t)p(s_t|s_{t-1})p(m_t|m_{t-1}, s_t) \tag{1}$$

The factorization of the probability specifies the model parameters Λ we need to define in the SGMM-HMM, including the transition probability $a_{ij} = p(s_t = j|s_{t-1} = i)$, and the observation probability given state j and mixture l distributed in a Gaussian manner

$$b_l^{(j)} = p(\boldsymbol{x}_t|s_t = j, m_t = l) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_l^{(j)}, \boldsymbol{\Sigma}_l^{(j)}) \tag{2}$$

Also, the mixture transition probabilities are defined as

$$p(m_t = l|m_{t-1} = k, s_t = j) = \begin{cases} c_{kl}^{(ij)} & \text{if } a_{ij} > 0 \\ 0 & \text{if } a_{ij} = 0 \end{cases} \tag{3}$$

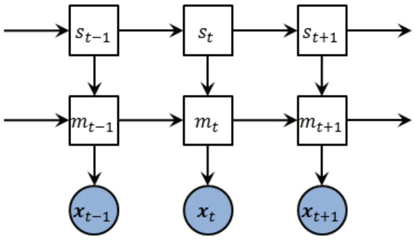

Fig. 1. A dynamic Bayesian network representation of the SGMM-HMM. Square nodes denote discrete variables, and shaded circles indicate continuous observed variables.

Here, the state s_{t-1} is excluded from the variable list on which the mixture m_t depends, as we assume that the state $s_{t-1} = i$ can be inferred from the mixture component $m_{t-1} = k$. The mixture transition probabilities for each state transition from i to j with a non-zero probability form a matrix $C^{(ij)} = \left[c_{kl}^{(ij)}\right]$. Each mixture transition matrix satisfies the following statistical constraint individually

$$\sum_l c_{kl}^{(ij)} = 1, \quad \text{for any feasible } i, j, k \tag{4}$$

We see that the mixture components in state i have multiple matrices of mixture transitions. Which transition matrix is activated at a particular time depends on the mastering state transitions. Also, we may refer to $C^{(ij)}, i = j$ as within-state mixture transitions, and $C^{(ij)}, i \neq j$ as cross-state transitions.

In this work, we present the SGMM model where the mixture components are Gaussian distributed as in (2). However, it is straightforward to extend the model such that each mixture component itself is a mixture of Gaussian distributions. Thus, the states in the extended SGMM model will contain mixtures of mixtures of Gaussians, and the interdependence of the top-level mixtures be accounted.

2.1. Properties of the SGMM-HMM

First, the HMM conditional independence assumption for observations is implicitly relaxed in the SGMM-HMM. This can be verified in Fig. 1 through the d-separation rule [3]: the observation variables are not d-separated by the state sequence due to the connection of the mixture variables.

The SGMM-HMM may also be portrayed in a state transition graph as in Fig. 2. The transitions between the states and the transitions between the corresponding mixture components (or substates) constitute a two-layer diagram, and are synchronized with each other. At first glance, it appears that the SGMM-HMM can be converted to an HMM by regarding each state/mixture pair as an augmented state. The resulting flat HMM has the same model topology as the lower-layer transition graph in Fig. 2, except that its transition probabilities are the product of the corresponding state and mixture transition probabilities in the SGMM model.

However, the SGMM-HMM is different and has several advantages over the flat HMM. First, unlike the flat HMM, the two-layer structure of the SGMM-HMM enforces the synchronization among different HMM paths. This extra constraint has great practical importance in modeling. When we learn the model parameters in the presence of numerous observation sequences, we hope that one observation sequence might be matched by one such HMM path. Through synchronization, other less likely paths have to go with the dominant path, and not to warp themselves to repeatedly match the current observation sequence. Thus, the synchronization prevents the path repetition problem, which might greatly discount the modeling power of the multiple-path model.

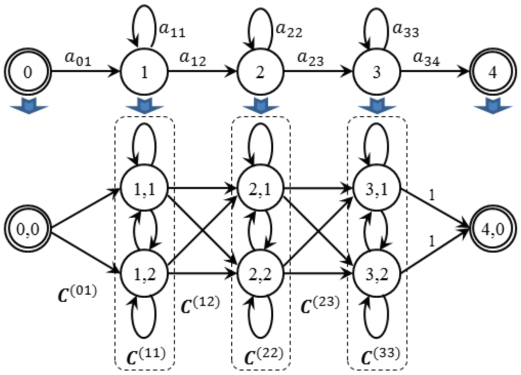

Fig. 2. Example of the two-layer state transition diagram for a 3-state 2-mixture left-to-right SGMM-HMM. The top layer consists of a Markov chain, in which each state corresponds to a column of the mixture components in the lower layer. The transitions between the mixture components are synchronized with the state transitions. The initial and final states are non-emitting, and represented by double circles.

Second, the two-layer decomposition of the SGMM-HMM retains the essential interpretation of the state transitions, and allows the manipulation on the mixture transitions with great flexibility. In particular, the type of the model, such as ergodic or left-to-right, is decided by the state transition matrix, regardless of the mixture transitions. This means that we can modify or prune the mixture transitions at ease, only if the statistical constraint (4) is satisfied. Such operations pose a challenge to the flat HMM, where arbitrary pruning of the transitions might, for example, cause some states trapped in a dead loop.

Finally, in many applications of HMMs, it is often of interest to find the most likely state sequence, excluding the mixture component sequence. In Section 2.3, we propose a modified Viterbi algorithm to find the best state sequence through the SGMM model by integrating out the mixture variables. This choice is infeasible for the flat HMM, which can only find the best sequence of the augmented states.

The SGMM-HMM contains the multiple-path models as special cases. If we set the within-state transition matrices to the identity matrix, it results in a model composed of parallel HMM paths with cross-coupled connections [8]. Further forcing the cross-state transition matrices to be a permutation matrix gives rise to a mixture of separate parallel paths [5], [6]. Since the SGMM-HMM still imposes the synchronization between the HMM paths, to be precise, we should say that the SGMM-HMM can represent parallel and synchronous HMM paths.

2.2. Training Procedure

The parameters of the SGMM-HMM can be learned in an expectation-maximization (EM) algorithm, similar to the regular HMM. As both the states and the mixture components are latent variables, we need to maximize the following EM auxiliary function

$$Q(\hat{\Lambda}|\Lambda) = \sum_{s_1^T} \sum_{m_1^T} p(s_1^T, m_1^T | \boldsymbol{x}_1^T, \Lambda) \log p(\boldsymbol{x}_1^T, s_1^T, m_1^T | \hat{\Lambda}) \tag{5}$$

where Λ and $\hat{\Lambda}$ denote the existing and new estimates of the model parameters, respectively. In the E step, the Q function requires finding the following sufficient statistics: the posterior probability of being in mixture k of state j at time t, $\gamma_t(j,l)$; the joint posterior probability of two successive state/mixture pairs, $\xi_t(i,k,j,l)$; and the joint posterior probability of two successive state variables, $\zeta_t(i,j)$, so that

$$\gamma_t(j,l) = p(s_t = j, m_t = l|\boldsymbol{x}_1^T, \Lambda) = \frac{\alpha_t(j,l)\beta_t(j,l)}{p(\boldsymbol{x}_1^T|\Lambda)} \quad (6)$$

$$\begin{aligned}\xi_t(i,k,j,l) &= p(s_{t-1} = i, m_{t-1} = k, s_t = j, m_t = l|\boldsymbol{x}_1^T, \Lambda) \\ &= \frac{\alpha_{t-1}(i,k)a_{ij}c_{kl}^{(ij)}b_l^{(j)}\beta_t(j,l)}{p(\boldsymbol{x}_1^T|\Lambda)} \quad (7)\end{aligned}$$

$$\zeta_t(i,j) = p(s_{t-1} = i, s_t = j|\boldsymbol{x}_1^T, \Lambda) = \sum_k \sum_l \xi_t(i,k,j,l) \quad (8)$$

where we have defined the forward and backward probabilities as

$$\alpha_t(j,l) = p(\boldsymbol{x}_1^t, s_t = j, m_t = l|\Lambda) \quad (9)$$

$$\beta_t(j,l) = p(\boldsymbol{x}_{t+1}^T|s_t = j, m_t = l, \Lambda) \quad (10)$$

The two quantities can be evaluated recursively in the forward-backward algorithm.

In the M step, differentiating the Q function with respect to the model parameters and equating them to zero yields

$$\hat{a}_{ij} = \frac{\sum_{t=1}^T \zeta_t(i,j)}{\sum_{t=1}^T \sum_j \zeta_t(i,j)} \quad (11)$$

$$\hat{c}_{kl}^{(ij)} = \frac{\sum_{t=1}^T \xi_t(i,k,j,l)}{\sum_{t=1}^T \sum_l \xi_t(i,k,j,l)} \quad (12)$$

The re-estimation formulae for the Gaussian means and variances are identical to those for the regular GMM-HMM, and omitted here.

One problem in learning the SGMM-HMM is how to gradually increase the mixture components to achieve a model with an optimal performance. In this work, the model is learned in two steps. First, a standard GMM-HMM is achieved by gradually increasing the number of the mixtures to the required number; then, the SGMM model is re-estimated by initializing the mixture transition probabilities to the weights of the Gaussian mixtures. It is admitted that there may exist more efficient methods to train the SGMM model. For example, we can run some sequential-data clustering algorithm to find a suitable initialization for multiple HMM paths, then establish connections among these paths for a complete SGMM re-estimation.

2.3. Decoding Algorithm

A direct method to infer the SGMM-HMM is to simultaneously find the most likely state/mixture pair sequence $\{s_1^T, m_1^T\}$ for a given observations sequence. This method may not be optimal, as in most cases we are only interested in the governing state sequence. Here, we propose a modified Viterbi algorithm to find the best state sequence through the SGMM model. The proposed algorithm embeds the forward algorithm in the dynamic programming procedure to integrate out the latent mixture variables. Let $\delta_t(j)$ be the highest probability of observing the partial sequence $\boldsymbol{x}_1^t$ and being in state j at time t

$$\delta_t(j) = \max_{s_1^{t-1}} p(\boldsymbol{x}_1^t, s_t = j) \quad (13)$$

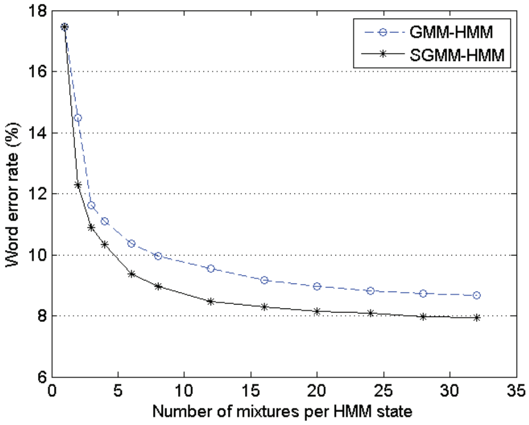

Fig. 3. WER (%) as a function of the number of mixtures per state using the SGMM-HMMs on the Aurora 2 test set.

and $\delta_t(j,l) = p(\boldsymbol{x}_1^t, s_t = j, m_t = l)$, its associated portion on each mixture component l. Obviously, we have $\delta_t(j) = \sum_l \delta_t(j,l)$. The best state sequence for $\delta_t(j)$ can be found using the dynamic programming

$$\delta_t(j) = \max_i a_{ij} \sum_l \sum_k \delta_{t-1}(i,k)c_{kl}^{(ij)}b_l^{(j)}(\boldsymbol{x}_t) \quad (14)$$

It should be noted that the above recursive procedure is an approximate inference, because the maximized quantity $\delta_{t-1}(i)$ may not warrant to be the highest after a re-weighted sum of its portions, as $\sum_k \delta_{t-1}(i,k)c_{kl}^{(ij)}$ in (14). The approximation will be accurate enough when, as is often the case, the probability $\delta_t(j)$ is dominated by one or a few of its mixtures.

3. EXPERIMENTS AND RESULTS

The proposed algorithm is evaluated on the Aurora 2 database [9] of connected digits. The test set consists of three different parts. Test Set A and Set B each contain 4 types of additive noises, and the data in Set C are contaminated with 2 types of additive noises as well as channel distortion. For each noise type, a subset of the clean speech utterances is contaminated at SNRs ranging from 20 to -5 dB at a 5 dB step size, which, including the clean condition, constitute 7 different SNR levels.

The multi-style training set is used to learn the baseline GMM-HMM and the SGMM-HMM systems. Following the standard Aurora 2 recipe for acoustic model training, each digit is modeled by a 16-state left-to-right HMM, and the silence and the short pause are modeled by three and one states, respectively. The number of mixtures per state for the silence model is roughly 1.5 times the size for the digit models. Each feature vector consists of 12 mel-cepstral coefficients and log energy, and their delta and delta-delta coefficients, to which cepstral mean subtraction (CMS) are applied in an utterance level. The 20-mixture GMM baseline yields word error rate (WER) of 8.96% by averaging over SNRs between 20 and 0 dB of three test sets.

Fig. 3 compares the recognition accuracy of the proposed SGMM system with the GMM system in different numbers of mixture components per state. The significant improvements over the GMM system are observed at all levels of the model complexities. With more than 6 mixtures, the SGMM system can reduce the WER by from 8% to 11%.

Table 1. WER (%) of the 20-mixture SGMM system with various configurations.

System	WER (%)
GMM-HMM	8.96
SGMM-HMM	8.07
With state transitions fixed	8.07
With Gaussian means & variances fixed	8.48
Diagonalizing within-state mixture transitions	8.57
Diagonalizing cross-state mixture transitions	9.70
Diagonalizing both within-state & cross-state	10.92

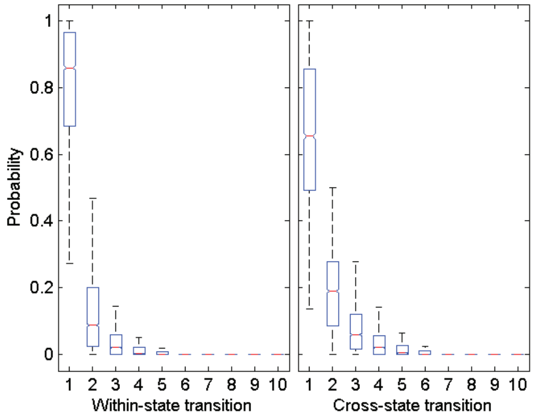

Fig. 4. Box plots of the ordered outgoing transition probabilities for the 20-mixture SGMM system. Top 10 transitions for within-state and cross-state are enclosed, respectively.

The second experiment investigates the 20-mixture SGMM system with different configurations, as shown in Table 1. First, the SGMM yield WER of 8.07%, 10% relative improvement over the GMM system. To quantify the incremental contribution of different model parameters in the course of refining SGMM system based on the GMM system, we produce two systems by fixing some parameters to the baseline GMM system, as shown in the third part of Table 1. We can see that the further refinement of the Gaussian means and variances is helpful in achieving a good performance for the SGMM system, whereas refining the state transition probabilities has not effect. This is not unexpected as the Gaussian parameters possess much more discriminating power than others in a GMM system.

The last part of Table 1 shows the performance of the SGMM system configured as several multiple-path models. We modify the mixture transition matrices of the well-trained SGMM system, such that each row has 1 in one entry and 0 everywhere else. This operation is loosely referred to as diagonalizing. For the within-state transition matrices, ones are assigned to the main diagonals. For the cross-state transition matrices, ones are assigned to those entries with probabilities as high as possible, provided the resulting matrix is a permutation matrix. After diagonalizing the mixture transition matrices, the re-estimation is then repeated for several times until convergence. Hence, the first row of the last part of Table 1 represents the system of cross-coupled parallel HMM paths [8], and the third row for a mixture of separate parallel paths [5], [6]. It is shown that these multiple-path models do not produce higher recognition accuracy than the SGMM system, and the models whose cross-state transition matrices are diagonalized perform even worse than the regular GMM system. This observation may indicate that the multiple-path models, in a simplistic setup as described in this paper, over-correct the trajectory folding problem associated with the GMM-HMM.

Finally, we analyze the distributions of the mixture transition probabilities for the 20-mixture SGMM system. Usually, the more peaked the transition probabilities, the more discriminability they may hold. The outgoing transition probabilities of each mixture are sorted in a descending order. Then the within-state and cross-state transitions of all mixtures at separate order levels are pooled, respectively, and their statistics are illustrated with the box plots in Fig. 4. It is observed that the ordered probabilities decay dramatically along with the level of orders. In fact, if we place a threshold of 10^{-5} on the effective outgoing transitions, the average fan-out will be 4.0 for within-state, and 5.1 for cross-state, respectively. Moreover, the first order bar of the within-state transitions, which mainly consists of the self-loop transitions, is more prominent than the first order bar of the cross-state transitions.

4. CONCLUSION

We have proposed the SGMM-HMM to explicitly model the relationship among the mixture components, and achieve more accurate representation of heterogeneous data. Due to its close similarity to the GMM-HMM, the SGMM system can be learned starting from an existing GMM system to achieve an incremental improvement. Another advantage is that the SGMM system is less expensive, and can work readily with many acoustic modeling techniques established in the literature, like MLLR and HLDA. Our initial experiments on the Aurora 2 database have showed the significant gain over the standard GMM system, encouraging further investigation on more challenging tasks. In the future, we plan to explore more effective methods to produce the mixture transition matrices with greater sparsity, such that less computational load and more discriminating power could both be achieved.

5. REFERENCES

[1] Y. Gong, "Stochastic trajectory modeling and sentence searching for continuous speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 1, pp. 33–44, 1997.

[2] C. J. Wellekens, "Explicit time correlation in hidden Markov models for speech recognition," in *Proc. ICASSP*, 1987, pp. 384–386.

[3] J. A. Bilmes, "Graphical models and automatic speech recognition," in *Mathematical foundations of speech and language processing*. Springer-Verlag, New York, 2003.

[4] P. Kenny, M. Lennig, and P. Mermelstein, "A linear predictive HMM for vector-valued observations with applications to speech recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 2, pp. 220–225, Feb. 1990.

[5] J. Su, H. Li, J. P. Haton, and K. T. Ng, "Speaker time-drifting adaptation using trajectory mixture hidden Markov models," in *Proc. ICASSP*, 1996, pp. 709–712.

[6] F. Korkmazskiy, B. H. Juang, and F. K. Soong, "Generalized mixture of HMMs for continuous speech recognition," in *Proc. ICASSP*, 1997, pp. 1443–1446.

[7] K. P. Murphy, *Dynamic Bayesian networks: representation, inference and learning*, Ph.D. thesis, Univ. Calif. Berkeley, 2002.

[8] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[9] H. G. Hirsch and D. Pearce, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ISCA ITRW ASR*, 2000, pp. 181–188.