ON THE USE OF TIED-MIXTURE DISTRIBUTIONS

Owen Kimball, Mari Ostendorf

Electrical, Computer and Systems Engineering
Boston University, Boston, MA 02215

ABSTRACT

Tied-mixture (or semi-continuous) distributions are an important tool for acoustic modeling, used in many high-performance speech recognition systems today. This paper provides a survey of the work in this area, outlining the different options available for tied mixture modeling, introducing algorithms for reducing training time, and providing experimental results assessing the trade-offs for speaker-independent recognition on the Resource Management task. Additionally, we describe an extension of tied mixtures to segment-level distributions.

1. INTRODUCTION

Tied-mixture (or semi-continuous) distributions have rapidly become an important tool for acoustic modeling in speech recognition since their introduction by Huang and Jack [1] and Bellegarda and Nahamoo [2], finding widespread use in a number of high-performance recognition systems. Tied mixtures have a number of advantageous properties that have contributed to their success. Like discrete, "non-parametric" distributions, tied mixtures can model a wide range of distributions including those with an "irregular shape," while retaining the smoothed form characteristic of simpler parametric models. Additionally, because the component distributions of the mixtures are shared, the number of free parameters is reduced, and tied-mixtures have been found to produce robust estimates with relatively small amounts of training data. Under the general heading of tied mixtures, there are a number of possible choices of parameterization that lead to systems with different characteristics. This paper outlines these choices and provides a set of controlled experiments assessing trade-offs in speaker-independent recognition on the Resource Management corpus in the context of the stochastic segment model (SSM). In addition, we introduce new variations on training algorithms that reduce computational requirements and generalize the tied mixture formalism to include segment-level mixtures.

2. PREVIOUS WORK

A central problem in the statistical approach to speech recognition is finding a good model for the probability of acoustic observations conditioned on the state in hidden-Markov models (HMM), or for the case of the SSM, conditioned on a region of the model. Some of the options that have been investigated include discrete distributions based on vector quantization, as well as Gaussian, Gaussian mixture and tied-Gaussian mixture distributions. In tied-mixture modeling, distributions are modeled as a mixture of continuous densities, but unlike ordinary, non-tied mixtures, rather than estimating the component Gaussian densities separately, each mixture is constrained to share the same component densities with only the weights differing. The probability density of observation vector $\mathbf{x}$ conditioned on being in state i is thus

$$\mathrm{p}(\mathbf{x} \mid s = i) = \sum_k w_{ik} \mathrm{p}_k(\mathbf{x}). \qquad (1)$$

Note that the component Gaussian densities, $\mathrm{p}_k(\mathbf{x}) \sim N(\mu_k, \Sigma_k)$, are not indexed by the state, i. In this light, tied mixtures can be seen as a particular example of the general technique of tying to reduce the number of model parameters that must be trained [3].

"Tied mixtures" and "semi-continuous HMMs" are used in the literature to refer to HMM distributions of the form given in Equation (1). The term "semi-continuous HMMs" was coined by Huang and Jack, who first proposed their use in continuous speech recognition [1]. The "semi-continuous" terminology highlights the relationship of this method to discrete and continuous density HMMs, where the mixture component means are analogous to the vector quantization codewords of a discrete HMM and the weights to the discrete observation probabilities, but, as in continuous density HMMs, actual quantization with its attendant distortion is avoided. Bellegarda and Nahamoo independently developed the same technique which they termed "tied mixtures" [2]. For simplicity, we use only one name in this paper, and choose the term tied mixtures, to highlight the relationship to other types of mixture distributions and because our work is based on the SSM, not the HMM.

Since its introduction, a number of variants of the tied mixture model have been explored. First, different assumptions can be made about feature correlation within

individual mixture components. Separate sets of tied mixtures have been used for various input features including cepstra, derivatives of cepstra, and power and its derivative, where each of these feature sets have been treated as independent observation streams. Within an observation stream, different assumptions about feature correlation have been explored, with some researchers currently favoring diagonal covariance matrices [4, 5] and others adopting full covariance matrices [6, 7].

Second, the issue of parameter initialization can be important, since the training algorithm is an iterative hill-climbing technique that guarantees convergence only to a local optimum. Many researchers initialize their systems with parameters estimated from data subsets determined by K-means clustering, e.g. [6], although Paul describes a different, bootstrapping initialization [4]. Often a large number of mixture components are used and, since the parameters can be overtrained, contradictory results are reported on the benefits of parameter re-estimation. For example, while many researchers find it useful to reestimate all parameters of the mixture models in training, BBN reports no benefit for updating means and covariances after the initialization from clustered data [7].

Another variation, embodied in the CMU senone models [8], involves tying mixture weights over classes of context-dependent models. Their approach to finding regions of mixture weight tying involves clustering discrete observation distributions and mapping these clustered distributions to the mixture weights for the associated triphone contexts.

In addition to the work described above, there are related methods that have informed the research concerning tied mixtures. First, mixture modeling does not require the use of Gaussian distributions. Good results have also been obtained using mixtures of Laplacian distributions [9, 10], and presumably other component densities would perform well too. Ney [11] has found strong similarities between radial basis functions and mixture densities using Gaussians with diagonal covariances. Recent work at BBN has explored the use of elliptical basis functions which share many properties with tied mixtures of full-covariance Gaussians [12]. Second, the positive results achieved by several researchers using non-tied mixture systems [13] raise the question of whether tied-mixtures have significant performance advantages over untied mixtures when there is adequate training data. It is possible to strike a compromise and use limited tying: for instance the context models of a phone can all use the same tied distributions (e.g. [14, 15]).

Of course, the best choice of model depends on the nature of the observation vectors and the amount of training data. In addition, it is likely that the amount of tying in a system can be adjusted across a continuum to fit the particular task and amount of training data. However, an assessment of modeling trade-offs for speaker-independent recognition is useful for providing insight into the various choices, and also because the various results in the literature are difficult to compare due to differing experimental paradigms.

3. TRAINING ALGORITHMS

In this section we first review properties of the SSM and then describe the training algorithm used for tied mixtures with the SSM. Next, we describe an efficient method for training context-dependent models, and lastly we describe a parallel implementation of the trainer that greatly reduces experimentation time.

3.1. The SSM and "Viterbi" Training with Tied Mixtures

The SSM is characterized by two components: a family of length-dependent distribution functions and a deterministic mapping function that determines the distribution for a variable-length observed segment. More specifically, in the work presented here, a linear time warping function maps each observed frame to one of m regions of the segment model. Each region is described by a tied Gaussian mixture distribution, and the frames are assumed conditionally independent given the length-dependent warping. The conditional independence assumption allows robust estimation of the model's statistics and reduces the computation of determining a segment's probability, but the potential of the segment model is not fully utilized. Under this formulation, the SSM is similar to a tied-mixture HMM with a phone-length-dependent, constrained state trajectory. Thus, many of the experiments reported here translate to HMM systems.

The SSM training algorithm [16] iterates between segmentation and maximum likelihood parameter estimation, so that during the parameter estimation phase of each iteration, the segmentation of that pass gives a set of known phonetic boundaries. Additionally, for a given phonetic segmentation, the assignment of observations to regions of the model is uniquely determined. SSM training is similar to HMM "Viterbi training", in which training data is segmented using the most likely state sequence and model parameters are updated using this segmentation. Although it is possible to define an SSM training algorithm equivalent to the Baum-Welch algorithm for HMMs, the computation is prohibitive for the SSM because of the large effective state space.

The use of a constrained segmentation greatly simplifies parameter estimation in the tied mixture case, since there is only one unobserved component, the mixture mode. In this case, the parameter estimation step of the iterative segmentation/estimation algorithm involves the standard iterative expectation-maximization (EM) approach to estimating the parameters of a mixture distribution [17]. In contrast, the full EM algorithm for tied mixtures in an HMM handles both the unobserved state in the Markov chain and the unobserved mixture mode [2].

3.2. Tied-Mixture Context Modeling

We have investigated two methods for training context-dependent models. In the first, weights are used to combine the probability of different types of context. These weights can be chosen by hand [18] or derived automatically using a deleted-interpolation algorithm [3]. Paul evaluated both types of weighting for tied-mixture context modeling and reported no significant performance difference between the two [4]. In our experiments, we evaluated just the use of hand-picked weights.

In the second method, only models of the most detailed context (in our case triphones) are estimated directly from the data and simpler context models (left, right, and context-independent models) are computed as marginals of the triphone distributions. The computation of marginals is negligible since it involves just the summing and normalization of mixture weights at the end of training. This method reduces the number of model updates in training in proportion to the number of context types used, although the computation of observation probabilities conditioned on the mixture component densities, remains the same. In recognition with marginal models, it is still necessary to combine the different context types, and we use the same hand-picked weights as before for this purpose. We compared the two training methods and found that performance on an independent test set was essentially the same for both methods (marginal training produced 2 fewer errors on the Feb89 test set) and the marginal trainer required 20 to 35% less time, depending on the model size and machine memory.

3.3. Parallel Training

To reduce computation, our system prunes low probability observations, as in [4], and uses the marginal training algorithm described above. However, even with these savings, tied-mixture training involves a large computation, making experimentation potentially cumbersome. When the available computing resources consist of a network of moderately powerful workstations, as is the case at BU, we would like to make use of many machines at once to speed training. At the highest level, tied mixture training is inherently a sequential process, since each pass requires the parameter estimates from the previous pass. However, the bulk of the training computation involves estimating counts over a database, and these counts are all independent of each other. We can therefore speed training by letting machines estimate the counts for different parts of the database in parallel and combine and normalize their results at the end of each pass.

To implement this approach we use a simple "bakery" algorithm to assign tasks: as each machine becomes free, it reads and increments the value of a counter from a common location indicating the sentences in the database it should work on next. This approach provides load balancing, allowing us to make efficient use of machines that may differ in speed. Because of the coarse grain of parallelism (one task typically consists of processing 10 sentences), we can use the relatively simple mechanism of file locking for synchronization and mutual exclusion, with no noticeable efficiency penalty. Finally, one processor is distinguished as the "master" processor and is assigned to perform the collation and normalization of counts at the end of each pass. With this approach, we obtain a speedup in training linear with the number of machines used, providing a much faster environment for experimentation.

4. MODELING & ESTIMATION TRADE-OFFS

Within the framework of tied Gaussian mixtures, there are a number of modeling and training variations that have been proposed. In this section, we will describe several experiments that investigate the performance implications of some of these choices.

4.1. Experimental Paradigm

The experiments described below were run on the Resource Management (RM) corpus using speaker-independent, gender-dependent models trained on the standard SI-109 data set. The feature vectors used as input to the system are computed at 10 millisecond intervals and consist of 14 cepstral parameters, their first differences, and differenced energy (second cepstral differences are not currently used). In recognition, the SSM uses an N-best rescoring formalism to reduce computation: the BBN BYBLOS system [7] is used to generate 20 hypotheses per sentence, which are rescored by the SSM and combined with the number of phones, number of words, and (optionally) the BBN HMM score, to rerank the hypotheses. The weights for recombination

are estimated on one test set and held fixed for all other test sets. Since our previous work has indicated problems in weight estimation due to test-set mismatch, we have recently introduced a simple time normalization of the scores that effectively reduces the variability of scores due to utterance length and leads to more robust performance across test sets.

Although the weight estimation test set is strictly speaking part of the training data, we find that for most experiments, the bias in this type of testing is small enough to allow us to make comparisons between systems when both are run on the weight-training set. Accordingly some of the experiments reported below are only run on the weight training test set. Of course, final evaluation of a system must be on an independent test set.

4.2. Experiments

We conducted several series of experiments to explore issues associated with parameter allocation and training. The results are compared to a baseline, non-mixture SSM that uses full covariance Gaussian distributions.

The first set of experiments examined the number of component densities in the mixture, together with the choice of full- or diagonal-covariance matrices for the mixture component densities. Although the full covariance assumption provides a more detailed description of the correlation between features, diagonal covariance models require substantially less computation and it may be possible to obtain very detailed models using a larger number of diagonal models.

In initial experiments with just female speakers, we used diagonal covariance Gaussians and compared 200- versus 300-density mixture models, exploring the range typically reported by other researchers. With context-independent models, after several training passes, both systems got 6.5% word error on the Feb89 test set. For context-dependent models, the 300-density system performed substantially better, with a 2.8% error rate, compared with 4.2% for the 200 density system. These results compare favorably with the baseline SSM which has an error rate on the Feb89 female speakers of 7.7% for context-independent models and 4.8% for context-dependent models.

For male speakers, we again tried systems of 200 and 300 diagonal covariance density systems, obtaining error rates of 10.9% and 9.1% for each, respectively. Unlike the females, however, this was only slightly better than the result for the baseline SSM, which achieves 9.5%. We tried a system of 500 diagonal covariance densities, which gave only a small improvement in performance to 8.8% error. Finally, we tried using full-covariance Gaussians for the 300 component system and obtained an 8.0% error rate. The context-dependent performance for males using this configuration showed similar improvement over the non-mixture SSM, with an error rate of 3.8% for the mixture system compared with 4.7% for the baseline. Returning to the females, we found that using full-covariance densities gave the same performance as diagonal. We have adopted the use of full-covariance models for both genders for uniformity, obtaining a combined word error rate of 3.3% on the Feb89 test set. In the RM SI-109 training corpus, the training data for males is roughly 2.5 times that for females, so it is not unexpected that the optimal parameter allocation for each may differ slightly.

Unlike other reported systems which treat cepstral parameters and their derivatives as independent observation streams, the BU system models them jointly using a single output stream, which gives better performance than independent streams with a single Gaussian distribution (non-mixture system). Presumably, the result would also hold for mixtures.

Since the training is an iterative hill climbing technique, initialization can be important to avoid converging to a poor solution. In our system, we choose initial models, using one of the two methods described below. These models are used as input to several iterations of context-independent training followed by context-dependent training. We add a small padding value to the weight estimates in the early training passes to delay premature parameter convergence.

We have investigated two methods for choosing the initial models. In the first, we cluster the training data using the *K-means* algorithm and then estimate a mean and covariance from the data corresponding to each cluster. These are then used as the parameters of the component Gaussian densities of the initial mixture. In the second method, we initialize from models trained in a non-mixture version of the SSM. The initial densities are chosen as means of triphone models, with covariances chosen from the corresponding context-independent model. For each phone in our phone alphabet we iteratively choose the triphone model of that phone with the highest frequency of occurrence in training. The object of this procedure is to attempt to cover the space of phones while using robustly estimated models.

We found that the *K-means* initialized models converged slower and had significantly worse performance on independent test data than that of the second method. Although it is possible that with a larger padding value added to the weight estimates and more training passes, the *K-means* models might have "caught up" with the

System	Test set	
	Oct 89	Sep 92
Baseline SSM	4.8	8.5
T.M. SSM	3.6	7.3
T.M. SSM + HMM	3.2	6.1

Table 1: Word error rate on the Oct89 and Sep92 test sets for the baseline non-mixture SSM, the tied-mixture SSM alone and the SSM in combination with the BYBLOS HMM system.

other models, we did not investigate this further.

The various elements of the mixtures (means, covariances, and weights) can each be either updated in training, or assumed to have fixed values. In our experiments, we have consistently found better performance when all parameters of the models are updated.

Table 1 gives the performance on the RM Oct89 and Sept92 test set for the baseline SSM, the tied-mixture SSM system, and the tied-mixture system combined in N-best rescoring with the BBN BYBLOS HMM system. The mixture SSM's performance is comparable to results reported for many other systems on these sets. We note that it may be possible to improve SSM performance by incorporating second difference cepstral parameters as most HMM systems do.

5. SEGMENTAL MIXTURE MODELING

In the version of the SSM described in this paper, in which observations are assumed conditionally independent given model regions, the dependence of observations over time is modeled implicitly by the assumption of time-dependent stationary regions in combination with the constrained warping of observations to regions. Because segmentation is explicit in this model, in principle it is straightforward to model distinct segmental trajectories over time by using a mixture of such segment-level models, and thus take better advantage of the segment formalism. The probability of the complete segment of observations, $\mathbf{Y}$, given phonetic unit α is then

$$\mathrm{p}(\mathbf{Y} \mid \alpha) = \sum_k w_k \, \mathrm{p}(\mathbf{Y} \mid \alpha_k),$$

where each of the densities $\mathrm{p}(\mathbf{Y} \mid \alpha_k)$ is an SSM. The component models could use single Gaussians instead of tied mixtures for the region dependent distributions and they would remain independent frame models, but in training all the observations for a phone would be updated jointly, so that the mixture components capture distinct trajectories of the observations across a complete segment. In practice, each such trajectory is a point in a very high-dimensional feature space, and it is necessary to reduce the parameter dimension in order to train such models. There are several ways to do this. First, we can model the trajectories within smaller, subphonetic units, as in the microsegment model described in [19, 20]. Taking this approach and assuming microsegments are independent, the probability for a segment is

$$\mathrm{p}(\mathbf{Y} \mid \alpha) = \prod_j \sum_k w_{jk} \, \mathrm{p}(\mathbf{Y}_j \mid \alpha_{jk}), \qquad (2)$$

where α_{jk} is the k^{th} mixture component of microsegment j and $\mathbf{Y}_j$ is the subset of frames in $\mathbf{Y}$ that map to microsegment j. Given the SSM's deterministic warping and assuming the same number of distributions for all mixture components of a given microsegment, the extension of the EM algorithm for training mixtures of this type is straightforward. The tied-mixture SSM discussed in previous sections is a special case of this model, in which we restrict each microsegment to have just one stationary region and a corresponding mixture distribution.

A different way to reduce the parameter dimension is to continue to model the complete trajectory across a segment, but assume independence between subsets of the features of a frame. This case can be expressed in the general form of (2) if we reinterpret the $\mathbf{Y}_j$ as vectors with the same number of frames as the complete segment, but for each frame, only a specific subset of the original frame's features are used. We can of course combine these two approaches, and assume independence between observations representing feature subsets of different microsegmental units. There are clearly a large number of possible decompositions of the complete segment into time and feature subsets, and the corresponding models for each may have different properties. In general, because of constraints of model dimensionality and finite training data, we expect a trade-off between the ability to model trajectories across time and to model the correlation of features within a local time region.

Although no single model of this form may have all the properties we desire, we do not necessarily have to choose one to the exclusion of all others. All the models discussed here compute probabilities over the same observation space, allowing for a straightforward combination of different models, once again using the simple mechanism of non-tied mixtures:

$$\mathrm{p}(\mathbf{Y} \mid \alpha) = \sum_i \prod_j \sum_k w_{ijk} \, \mathrm{p}(\mathbf{Y}_j \mid \alpha_{ijk}).$$

In this case, each of the i components of the leftmost summation is some particular realization of the general

model expressed in Equation (2). Such a mixture can combine component models that individually have beneficial properties for modeling either time or frequency correlation, and the combined model may be able to model both aspects well. We note that, in principle, this model can also be extended to larger units, such as syllables or words.

6. SUMMARY

This paper provided an overview of work using tied-mixture models for speech recognition. We described the use of tied mixtures in the SSM as well as several innovations in the training algorithm. Experiments comparing performance for different parameter allocation choices using tied-mixtures were presented. The performance of the best tied-mixture SSM is comparable to HMM systems that use similar input features. Finally, we presented a general method we are investigating for modeling segmental dependence with the SSM.

ACKNOWLEDGMENTS

The authors gratefully acknowledge BBN Inc. for their help in providing the N-best sentence hypotheses. We thank J. Robin Rohlicek of BBN for many useful discussions. This research was jointly funded by NSF and DARPA under NSF grant number IRI-8902124, and by DARPA and ONR under ONR grant number N00014-92-J-1778.

References

1. Huang, X. D. and Jack, M. A., "Performance comparison between semi-continuous and discrete hidden Markov models," *IEE Electronics Letters*, Vol. 24 no. 3, pp. 149-150.
2. Bellegarda, J. R. and Nahamoo, D., "Tied Mixture Continuous Parameter Modeling for Speech Recognition," *IEEE Trans. on Acoust., Speech and Signal Processing*, Dec 1990, pp. 2033–2045.
3. Jelinek, F. and Mercer, R.L., "Interpolated Estimation of Markov Source Parameters from Sparse Data," in *Proc. Workshop Pattern Recognition in Practice*, May 1980, pp. 381-397.
4. Paul, D.B., "The Lincoln Tied-Mixture HMM Continuous Speech Recognizer," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, May 1991, pp. 329–332.
5. Murveit, H., Butzberger, J., Weintraub, M., "Speech Recognition in SRI's Resource Management and ATIS Systems," *Proc. of the DARPA Workshop on Speech and Natural Language*, June 1990, pp. 94–100.
6. Huang, X.D., Lee, K.F., Hon, H.W., and Hwang, M.-Y., "Improved Acoustic Modeling with the SPHINX Speech Recognition System," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, May 1991, pp. 345–348.
7. Kubala, F., Austin, S., Barry, C., Makhoul, J. Placeway, P., and Schwartz, R., "BYBLOS Speech Recognition Benchmark Results," *Proc. of the DARPA Workshop on Speech and Natural Language*, Asilomar, CA, Feb. 1991, pp. 77-82.
8. Hwang, M.-Y., Huang, X. D., "Subphonetic Modeling with Markov States - Senone," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, March 1992, pp. I-33–36.
9. Ney, H., Haeb-Umbach, R., Tran, B.-H., Oerder, M., "Improvements in Beam Search for 10000-Word Continuous Speech Recognition," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, April 1992, pp. I-9–12.
10. Baker, J. K., Baker, J. M., Bamberg, P., Bishop, K., Gillick, L., Helman, V., Huang, Z., Ito, Y., Lowe, S., Peskin, B., Roth, R., Scattone, F., "Large Vocabulary Recognition of Wall Street Journal Sentences at Dragon Systems," *Proc. of the DARPA Workshop on Speech and Natural Language*, February 1992, pp. 387-392.
11. H. Ney, "Speech Recognition in a Neural Network Framework: Discriminative Training of Gaussian Models and Mixture Densities as Radial Basis Functions," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, May 1991, pp. 573–576.
12. Zavaliagkos, G., Zhao, Y., Schwartz, R., and Makhoul,J., to appear in *Proc. of the DARPA Workshop on Artificial Neural Networks and CSR*, Sept. 1992.
13. Pallett, D., Results for the Sept. 1992 Resource Management Benchmark, presented at the DARPA Workshop on Artificial Neural Networks and CSR, Sept. 1992.
14. Lee, C., Rabiner, L., Pieraccini, R., and Wilpon, J., "Acoustic Modeling for Large Vocabulary Speech Recognition," *Computer Speech and Language*, April. 1990, pp. 127–165.
15. Paul, D. B., "The Lincoln Robust Continuous Speech Recognizer," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, May 1989, pp. 449–452.
16. Ostendorf, M. and Roukos, S. , "A Stochastic Segment Model for Phoneme-Based Continuous Speech Recognition," *IEEE Trans. on Acoust., Speech and Signal Processing*, Dec. 1989, pp. 1857–1869.
17. Dempster, A., Laird, N. and Rubin, D., "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statist. Soc. Ser. B*, Vol. 39 No. 1, pp. 1–22, 1977.
18. Schwartz, R., Chow, Y. L., Kimball, O., Roucos, S., Krasner, M. and Makhoul, J., "Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, March 1985, pp. 1205–1208.
19. Digalakis, V. *Segment-Based Stochastic Models of Spectral Dynamics for Continuous Speech Recognition*, Boston University Ph.D. Dissertation, 1992.
20. Kannan, A., and Ostendorf, M., "A Comparison of Trajectory and Mixture Modeling in Segment-Based Word Recognition," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, April 1993.