ROBUST HMM ESTIMATION WITH GAUSSIAN MERGING-SPLITTING AND TIED-TRANSFORM HMMS[*]

Ananth Sankar

Speech Technology and Research Laboratory
SRI International, Menlo Park, CA 94025
sankar@speech.sri.com

ABSTRACT

We present two different approaches for robust estimation of the parameters of context-dependent hidden Markov models (HMMs) for speech recognition. The first approach, the Gaussian Merging-Splitting (GMS) algorithm, uses Gaussian splitting to uniformly distribute the Gaussians in acoustic space, and merging so as to compute only those Gaussians that have enough data for robust estimation. We show that this method is more robust than our previous training technique. The second approach, called tied-transform HMMs, uses maximum-likelihood transformation-based acoustic adaptation algorithms to transform a small HMM to a much larger HMM. Since the transforms are shared or tied among Gaussians in the larger HMM, robust estimation is achieved. We show that this approach gives a significant improvement in recognition accuracy and a dramatic reduction in memory needed to store the models.

1. INTRODUCTION

Most conventional automatic speech recognition (ASR) systems are based on context-dependent (CD) phone-based hidden Markov models (HMMs) that use Gaussian mixture models (GMMs) for the state-conditioned observation densities. A commonly used CD unit is the triphone, which is a model of a phone in the context of a left and right phone. The number of triphones in typical HMMs is very large, and the training data limited, resulting in poor estimates of the model parameters. A popular solution to this problem is to use HMM state clustering where the states in a cluster share a set of parameters, such as a set of Gaussians [1, 2]. Pooling data among shared parameters in this way gives robust estimates.

In this paper, we report on two techniques we have recently developed for robust CD-HMM estimation. The first is a training algorithm called Gaussian Merging-Splitting (GMS), which we have also described in [3]. The GMS algorithm is a robust method to train the GMMs in state-clustered HMMs. In this approach, Gaussian splitting is used to uniformly distribute the Gaussians in the acoustic space. In addition, we use a Gaussian merging algorithm to automatically select the number of Gaussians in the GMMs for each state cluster subject to a constraint on the maximum possible number of Gaussians. This algorithm merges Gaussians that have too little data, effectively reducing the number of Gaussians in that state cluster, and leading to more robust estimation. This algorithm results in a variable number of Gaussians in each state cluster, where the number of Gaussians in each cluster is dependent on the amount of data segmented into the states in that cluster. We compare the GMS algorithm to our previous training algorithm, and show that it gives more robust model estimates.

While the GMS algorithm works well, it indirectly addresses the problem of robust estimation by estimating only those Gaussians for which there is enough data. We present a second approach called tied-transform HMMs (or T^2-HMMs) that directly addresses the problem of estimating Gaussian parameters with little data. In this approach, an HMM is first trained robustly using the GMS algorithm. This is then adapted to an HMM with a larger number of state clusters using maximum-likelihood (ML) transformation-based acoustic adaptation [4, 5, 6, 7]. The Gaussians in each state cluster in the larger HMM share the same transform, or set of transforms. Because of this sharing, or tying, we can robustly estimate the transforms, resulting in reliable estimates of the Gaussians in the larger HMM. We show that this approach gives a significant improvement in accuracy over the GMS algorithm. In addition, since the large HMM can be stored as a combination of the smaller HMM and the set of tied transforms, instead of having to store all the Gaussian parameters of the large HMM individually, we get a dramatic reduction in the number of model parameters that need to be stored.

In Section 2 we describe the GMS algorithm, and in Section 3 we describe the T^2-HMM approach. Experimental results for these methods are described in their respective sections. We summarize in Section 4.

2. GAUSSIAN MERGING-SPLITTING ALGORITHM

2.1. Previous Training Algorithm

SRI's DECIPHER^{TM} speech recognition system is based on HMM state clustering where the states in each cluster share the same set of Gaussians or Genone [1]. Each state in a cluster has a different mixture weight distribution to these shared Gaussians. The HMM states are clustered separately for each phone.

Consider the problem of training an HMM with 32 Gaussians per Genone. This is done by first training a phonetically tied mixture

[*] THIS WORK WAS SPONSORED BY DARPA THROUGH NAVAL COMMAND AND CONTROL OCEAN SURVEILLANCE CENTER UNDER CONTRACT N66001-94-C-6048.

(PTM) system, where all states in a phone share the same set of 100 Gaussians. The states in this phone are then clustered using bottom-up agglomerative clustering. For clustering, the distance between two states is given by the weighted-by-counts increase in entropy of the mixture weight distribution (to the shared 100 Gaussians) due to merging the two states [1].

The Gaussians in each state cluster are initialized using the corresponding 100 PTM Gaussians. The 100 Gaussians in each phone are clustered down to the required number for each state cluster through a series of steps involving the selection of the most likely Gaussians for each state cluster, and also Gaussian merging. Details of the algorithm can be found in [1].

This approach poses the following potential problem for the initial values of the Gaussians in the state clusters and hence the final models: The 100 PTM Gaussians cover the entire acoustic space for a particular phone; however, each state cluster for this phone covers only a small part of this large acoustic space. Thus, the PTM Gaussians may not be appropriate for initializing the Gaussians in the individual state clusters, and may result in inefficient use of the parameters. Since the expectation-maximization (EM) algorithm, which is commonly used to estimate HMMs, is locally optimal, good initial values are important.

To address this issue, we developed an algorithm that uses Gaussian splitting to uniformly distribute the Gaussians in the acoustic space for each state cluster. We then combined this with Gaussian merging so as to make sure that each Gaussian had at least a threshold of data. The combination of these methods thus gives good acoustic coverage for the Gaussians and also robust parameter estimates. We now briefly describe these methods.

2.2. Gaussian Splitting

We implemented an initialization scheme based on the splitting strategy commonly used in vector quantization [8]. In this approach, we first estimate a single Gaussian model for each Genone. Given the segmentation of data into HMM states, the ML estimate of these (single) Gaussians is globally optimal. We then split the Gaussian for each Genone into two by slightly perturbing the mean of the Gaussian along the direction of the standard-deviation vector, and reestimate the model by further EM training. This process of splitting and retraining is repeated until the required number of Gaussians is achieved. At each stage, we can choose how many Gaussians to split. Thus, if there are currently n Gaussians which we want to increase to m Gaussians, then we split the $m-n$ Gaussians which have the largest average sample variance. This average, computed by using the geometric mean, is a measure of the likelihood of the training data modeled by that single Gaussian model. The Gaussian with the largest variance is the one for which the training data likelihood is minimum. Since our goal is to maximize the training data likelihood, splitting this Gaussian is intuitively appealing. A similar Gaussian splitting algorithm is used in the Cambridge University HTK system, though a different criterion is used to select which Gaussian to split [2].

The Gaussian splitting approach can be configured in a variety of ways. For example, we may split all Gaussians at each stage, or may split only the single largest variance Gaussian, or may do something in between these extremes. We experimented with many of these approaches. While there was not a very significant difference in performance, we decided on a simple strategy that splits all Gaussians at each stage until we have the desired number of Gaussians per Genone.

Database	Word Error Rate (%)			
	Old algorithm		GMS algorithm	
	991 Genones	2027 Genones	991 Genones	2027 Genones
WSJ1	23.7	25.3	23.5	23.9
WSJ2	13.7	15.5	13.5	14.1
WSJ3	24.3	26.0	23.9	25.1

Table 1: Comparison of word error rates (%) for systems with different numbers of parameters

2.3. Gaussian Merging

If there is too little training data segmented into an HMM state cluster, then the Gaussians in the corresponding Genone will not be well estimated. To ensure robust Gaussian estimation, we used a Gaussian merging algorithm. In this method, the Gaussians in a Genone are iteratively merged using bottom-up agglomerative clustering until all Gaussians have at least a threshold amount of data. This threshold is specified by the user, and its optimum value is experimentally determined. For clustering, the distance between two Gaussians is given by the weighted-by-counts increase in entropy due to merging the Gaussians. More details of the GMS algorithm can be found in [3].

2.4. Experimental Results

We trained HMMs using a small subset of the Wall Street Journal (WSJ) SI-284 male training data. We used 71 of the 142 male training speakers and about 50 sentences from each for a total of about 3500 training sentences. We created three different WSJ test sets, denoted as WSJ1, WSJ2, and WSJ3, each with 10 male speakers and about 3600 words, for a total of about 10,900 words in all. For speed of experimentation, recognition was run from bigram lattices described in [9].

To measure the robustness of the training algorithms, we trained two HMMs, one with about 1000 Genones and the second with about 2000 Genones. Both models had 32 Gaussians per Genone. We trained models using both our old algorithm and the new GMS algorithm. Table 1 shows that the GMS algorithm performs similarly to the old method for the smaller model, but is significantly superior for the larger model, where the number of parameters is very large relative to the amount of training data. This shows the robustness of the GMS algorithm relative to our previous approach. We have given a more detailed comparison of the GMS algorithm with our previous training approach in [3].

3. TIED-TRANSFORM HMM

3.1. Algorithm Description

While the GMS algorithm gives robust parameter estimates, it does so indirectly by creating only as many Gaussians as there is enough data to estimate robustly. Thus, the total number of Gaussians is limited by the amount of training data. It would be advantageous to be able to reliably estimate a larger number of Gaussian parameters with the same amount of limited data. The tied-transform HMM (T^2-HMM) algorithm is one approach that achieves this goal.

As explained in Section 2, we use bottom-up agglomerative clustering to cluster HMM states in a state-cluster tree. This tree can be cut at different levels to create different numbers of state clusters. For each state cluster set, we can train a state-clustered HMM. The larger the number of clusters, the more difficult it is to robustly estimate the parameters with a limited amount of data.

We explain the concept of T^2-HMMs using the state-cluster tree in Figure 1. Suppose our goal is to train an HMM for the larger number of state clusters N. However, we do not have enough data to robustly estimate each Gaussian. In the T^2-HMM, we solve this problem by training an HMM for the smaller number of state clusters M, for which we assume we have enough data to robustly estimate each Gaussian. We can always select a small enough M so that robust Gaussian estimates are possible. Each state cluster in the larger HMM is a descendent of a state cluster in the smaller HMM as shown in the figure. Thus, we can define a mapping from the smaller to the larger HMM in terms of this ancestor-descendent relationship. The Gaussians in the state clusters of the larger HMM are transformed versions of the ancestor Gaussians in the smaller HMM. In the figure, the transformations $T(1),\ldots,T(m)$ are used to map the Gaussians in $GMM(0)$ to the Gaussians in $GMM(1),\ldots,GMM(m)$. $T(i)$ can also be a set of transforms, each tied to a cluster of acoustically similar Gaussians in a state cluster. Since the transforms are tied to a set of Gaussians in the N-state-cluster HMM, they can be estimated with the pooled data from all those Gaussians. This results in robust estimates of the transforms. In contrast, it is not possible to separately estimate the Gaussians in the N-cluster HMM because there is not enough training data for each Gaussian.

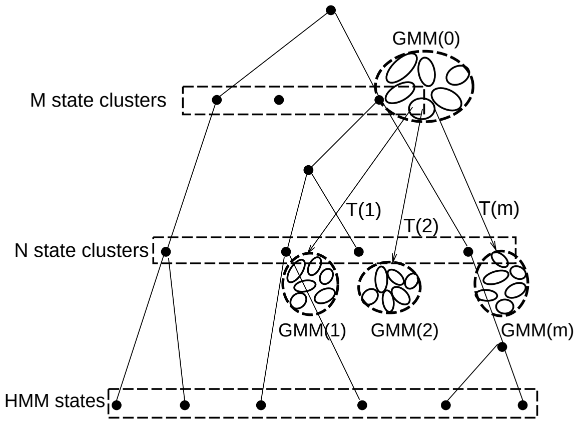

Figure 1: Illustration of T^2-HMM

The estimation problem is now that of computing the parameters of the smaller HMM and the parameters of the transformations. We can use different types of transformations as have been described in the acoustic adaptation literature [4, 5, 6, 7, 10]. In this paper, we chose to use the block-diagonal affine matrix transform of the Gaussian means as this has given us good performance in the past for speaker adaptation [10]. We solve the ML estimation problem iteratively. First, we assume identity transforms and estimate the parameters of the smaller HMM. Then we keep the parameters of the small HMM fixed, and estimate the transformations. This procedure can be iterated. However, in our experiments, we used only one iteration of this approach. The ML estimation of HMM parameters is well established, and that of the transformations has previously been studied in the context of acoustic adaptation [4, 5, 6, 7].

The T^2-HMM idea is related to that of Bayesian estimation of HMM parameters [11]. In Bayesian estimation too, a small HMM is adapted to a large HMM, but using Bayesian smoothing, instead of ML transformation-based adaptation as in T^2-HMMs. The T^2-HMM approach has the advantage that we need to store only the parameters of the small HMM and the tied transformation parameters, while in the Bayesian approach, all the Gaussian parameters of the large HMM must be individually stored. This results in a dramatic reduction in storage for the T^2-HMMs.

3.2. Experimental Results

We experimented using the Hub4 broadcast news domain. This is the domain for current U.S. Government-sponsored continuous speech recognition evaluations. For training we used the male subset of the 100 hours of Hub4 training data released by NIST for the 1997 DARPA-sponsored Hub4 evaluation. For testing, we used the 1996 Hub4 male development test. We ran recognition using trigram lattices generated with our recently developed lattice tools [12]. The Hub4 data is categorized into 7 different acoustic categories. These range from the planned speaking style of news announcers (F0), to noisy speech (F4), to speech that is not classifiable into any acoustic category (FX). A detailed description of this task can be found in [13].

Table 2 gives the recognition word error rates on this test set comparing the GMS algorithm, the T^2-HMM approach, and a Bayesian smoothing approach similar to that of [11]. We trained a crossword state clustered HMM with 2209 state clusters, and one with 8409 clusters. The 2209-cluster system is the one we used for the 1997 Hub4 evaluations. Table 2 shows that the 8409-cluster model gives worse performance than the 2209-cluster model when trained using the GMS algorithm. We then trained the 8409-cluster system by adapting the 2209-cluster system using both Bayesian smoothing and the T^2-HMM approach. Both techniques give an improvement over the GMS algorithm for the 8409-cluster system (32.0% to 30.7%). A smaller, but significant, improvement is observed over the 2209-cluster system (31.4% to 30.7%).

From Table 2, we see that the Bayesian smoothing algorithm and the T^2-HMM algorithm gave the same word error rate. However, the T^2-HMM can be stored much more efficiently, because we need to store only the smaller HMM and the set of transforms, as opposed to the Bayesian algorithm, where we must independently store each Gaussian in the larger model. In particular, as shown in Table 2, the T^2-HMM needs a factor of 3 less parameters to store the Gaussian distributions as compared to the Bayesian-trained HMM.

	GMS		Bayesian smoothing	T^2-HMM
	Number of clusters			
	2209	8409	8409	8409
	32 Gaussians per cluster			
	Number of Gaussian parameters in Millions			
	5.5	30	30	9.8
F0	14.2	15.6	14.2	14.4
F1	30.5	30.7	29.3	29.2
F2	37.5	38.2	36.2	36.4
F3	29.0	30.8	30.5	29.9
F4	27.5	27.4	26.2	26.2
F5	28.2	29.3	28.0	28.2
FX	56.4	56.2	56.0	56.0
All	31.4	32.0	30.7	30.7

Table 2: Comparison of word error rates (%) for different training algorithms on the 1996 Hub4 development data

4. SUMMARY

We presented two algorithms to robustly train state-clustered HMM systems. The first method, the GMS algorithm, addresses the problem indirectly by computing only those Gaussians for which there is enough data. The second algorithm, the T^2-HMM, does this directly by transforming well-estimated Gaussians in a smaller HMM to Gaussians in a larger HMM. The T^2-HMM algorithm gives robust estimates where we are unable to estimate the Gaussians directly because of limited training data. Experimental results show that the GMS algorithm is more robust than our previous training procedure for state-clustered HMMs. The T^2-HMM gives a significant improvement in accuracy over the GMS algorithm. It also allows us to estimate much larger HMMs than possible with the GMS algorithm, leading to improved accuracy. The T^2-HMM gave similar word error rates as compared to a Bayesian training algorithm. However, it required a factor of 3 less parameters to store the Gaussians.

5. REFERENCES

1. V. Digalakis, P. Monaco, and H. Murveit, "Genones: Generalized Mixture Tying in Continuous Hidden Markov Model-Based Speech Recognizers," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 4, pp. 281–289, 1996.

2. S. Young and P. Woodland, "The Use of State Tying in Continuous Speech Recognition," in *Proceedings of EUROSPEECH*, pp. 2203–2206, 1993.

3. A. Sankar, "Experiments with a Gaussian Merging-Splitting Algorithm for HMM training for Speech Recognition," in *Proceedings of DARPA Speech Recognition Workshop*, (Lansdowne, VA), February 1998.

4. A. Sankar and C.-H. Lee, "Stochastic Matching for Robust Speech Recognition," *IEEE Signal Processing Letters*, vol. 1, pp. 124–125, August 1994.

5. A. Sankar and C.-H. Lee, "A Maximum-Likelihood Approach to Stochastic Matching for Robust Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 4, pp. 190–202, May 1996.

6. V. Digalakis, D. Rtischev, and L. Neumeyer, "Speaker Adaptation Using Constrained Reestimation of Gaussian Mixtures," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 5, pp. 357–366, 1995.

7. C. J. Legetter and P. C. Woodland, "Flexible Speaker Adaptation Using Maximum Likelihood Linear Regression," in *Proceedings of the Spoken Language Systems Technology Workshop*, pp. 110–115, 1995.

8. Y. Linde, A. Buzo, and R. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications*, vol. COM-28, pp. 84–95, January 1980.

9. H. Murveit, J. Butzberger, V. Digalakis, and M. Weintraub, "Large-Vocabulary Dictation Using SRI's DECIPHER(TM) Speech Recognition System: Progressive-Search Techniques," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. II 319–322, 1993.

10. L. Neumeyer, A. Sankar, and V. Digalakis, "A Comparative Study of Speaker Adaptation Techniques," in *Proceedings of EUROSPEECH*, pp. 1127–1130, 1995.

11. J. Gauvain and C.-H. Lee, "Bayesian Learning for Hidden Markov Models with Gaussian Mixture State Observation Densities," *Speech Communication*, vol. 11, 1992.

12. F. Weng, A. Stolcke, and A. Sankar, "New developments in lattice-based search strategies in SRI's H4 system," in *Proceedings of DARPA Speech Recognition Workshop*, (Lansdowne, VA), February 1998.

13. R. Stern, "Specification of the 1996 Hub4 Broadcast News Evaluation," in *Proceedings of the DARPA Speech Recognition Workshop*, (Chantilly, VA), 1997.