A NEW LOOK AT HMM PARAMETER TYING FOR LARGE VOCABULARY SPEECH RECOGNITION*

Ananth Sankar

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

ABSTRACT

Most current state-of-the-art large-vocabulary continuous speech recognition (LVCSR) systems are based on state-clustered hidden Markov models (HMMs). Typical systems use thousands of state clusters, each represented by a Gaussian mixture model with a few tens of Gaussians. In this paper, we show that models with far more parameter tying, like phonetically tied mixture (PTM) models, give better performance in terms of *both recognition accuracy and speed*. In particular, we achieved between a 5 and 10% improvement in word error rate, while cutting the number of Gaussian distance computations in half, for three different Wall Street Journal (WSJ) test sets, by using a PTM system with 38 phone-class state clusters, as compared to a state-clustered system with 937 state clusters. For both systems, the total number of Gaussians was fixed at about 30,000. This result is of real practical significance as we show that a conceptually simpler PTM system can achieve faster and more accurate performance than current state-of-the-art state-clustered HMM systems.

1. INTRODUCTION

Most state-of-the-art speech recognition systems use hidden Markov models (HMMs) to model triphone speech units. The number of triphones is usually very large. For example, models with 10,000 triphones are common. Since each triphone is usually modeled by at least three HMM states, this results in about 30,000 HMM states. Each state is typically modeled by a Gaussian mixture model (GMM) with a few Gaussians. Thus, the total number of Gaussian parameters can be on the order of a few hundreds of thousands.

Estimating a separate GMM for each triphone state will require a huge amount of training data. However, since training data is usually limited, it is not possible to reliably estimate such a large number of parameters. In one of the first approaches to robust HMM estimation, called the Tied-Mixture (TM) HMM, a single set of Gaussian distributions was shared (or tied) across all the states [1, 2]. Since the Gaussians were shared, data could be pooled from different HMM states to train them robustly. Each state was differentiated by a different mixture weight distribution to these shared Gaussians. The shared Gaussians along with the mixture weights defined the state-dependent GMMs. Because of robust parameter estimation, TM HMMs were found to perform significantly better than "fully continuous" HMMs, where each state used a separate GMM. To get more detailed models than TM systems, phonetically tied mixture (PTM) systems were proposed. In these systems, a separate Gaussian codebook was shared among all triphone states corresponding to the same base phone [3].

The next step was state-clustered HMMs [4, 5, 6], where the amount of tying was decreased even further, and which represent the state of the art in current speech recognition technology. In this approach, the amount of tying is considerably less than in a TM or PTM system. HMM states are clustered according to acoustic similarity. The states in each cluster either share the same GMM [4, 5], or only share the same set of Gaussians but use different mixture weights for each state [6, 7]. A small number of Gaussians is used for each cluster, and improved acoustic resolution is achieved by increasing the number of state clusters.

State-clustered HMMs were experimentally shown to be superior to TM and PTM HMMs (e.g., see [6]). However, it is important to note that, in this comparison, the TM and PTM systems had a total of 256 and 4000 Gaussians, respectively, drastically fewer than the state-clustered system, which had about 24,000 Gaussians [6]. Other previous experiments with TM and PTM systems [2, 8, 9] also appear to have used very few Gaussians in comparison to that used in most current state-clustered systems. This observation suggests the possibility that approaches with more tying, like TM and PTM models, if appropriately designed, may perform better than in the past. In particular, we achieved between a 5 and 10% improvement in word error rate for three Wall Street Journal (WSJ) test sets using a 38-cluster PTM system as compared to a 937-cluster state-clustered system, while simultaneously cutting the number of Gaussian computations during recognition in half. This result is extremely significant as we have achieved a simultaneous improvement in accuracy and speed. It is also the first time, to the author's knowledge, that the conceptually simpler PTM system has been shown to outperform the currently dominant state-clustered approaches.

In Section 2, we describe the effect of changing the amount of tying, by varying the number of state clusters, on recognition accuracy and speed. In Section 3 we give detailed experimental results comparing PTM and state-clustered systems. We summarize in Section 4.

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

2. EFFECT OF VARYING THE NUMBER OF STATE CLUSTERS

The number of Gaussian parameters that can be robustly estimated is limited by the finite amount of training data. However, the same total number of Gaussians can be achieved by using fewer state clusters and more Gaussians per cluster, or more state clusters and fewer Gaussians per cluster. For example, a system with 1000 state clusters and 32 Gaussians per cluster has the same number of Gaussians as one with 32 clusters and 1000 Gaussians per cluster. We discuss how varying the number of clusters affects recognition accuracy and speed.

2.1. State Clustering and Accuracy

Consider a state-clustered system where each state cluster shares the same set of Gaussians, and each triphone state has a separate mixture weight distribution to these shared Gaussians. Suppose we can robustly train at most N state clusters with M Gaussians per cluster, given a certain amount of training data. It is possible to decrease the number of clusters, while increasing the number of Gaussians per cluster, without affecting the robustness of the Gaussian parameter estimates, since the total number of Gaussians can be held constant. We now examine the effect of decreasing the number of clusters on accuracy.

If the Gaussian distributions for the N state clusters do not overlap in acoustic space, then further grouping of the clusters will have no effect on performance, as the resulting models will be effectively the same, as shown in Figure 1. However, state clusters do overlap

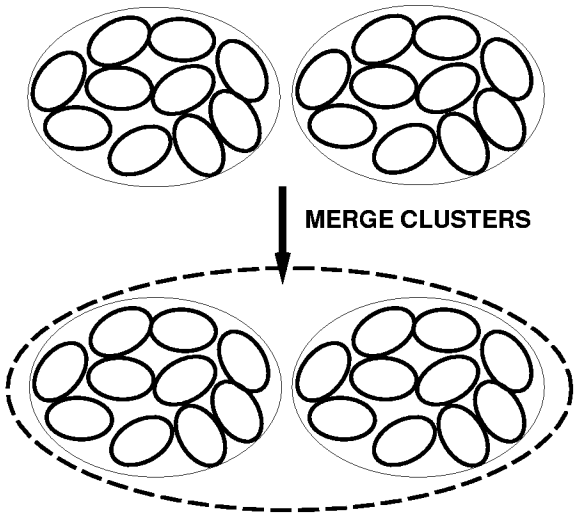

Figure 1: Merging of nonoverlapping state clusters

in acoustic space, as shown in Figure 2. In the overlap region, Gaussians are separately estimated for each state cluster. This causes two potential problems:

1. Since the data in the overlap region is divided between the two state clusters, the Gaussians in this region may not be robustly estimated.
2. There may be redundancy between the Gaussians from the two state clusters in the overlap region, resulting in wasted parameters.

We can address these problems by merging the two clusters into one cluster with $2M$ Gaussians. Since data from the two clusters is now used to estimate a single set of $2M$ Gaussians, there is more robust estimation in the overlap region. Further, the previously redundant Gaussians can now be more effectively used to increase acoustic resolution, as shown in Figure 2. The improved Gaussian estimates and the better acoustic resolution can lead to improved recognition accuracy.

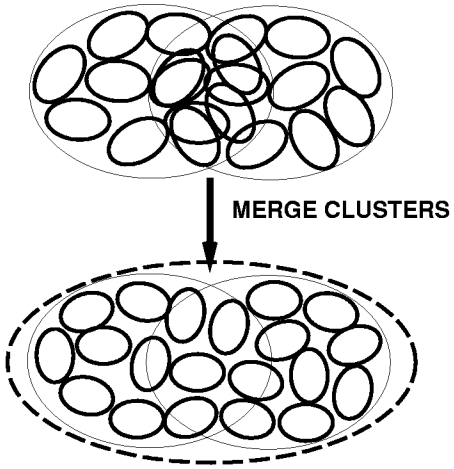

Figure 2: Merging of overlapping state clusters

While merging the two clusters has these advantages, it also has a potential drawback: it may be necessary to separately estimate Gaussians in the overlap regions to be able to aid in discriminating between the clusters, and merging the clusters can reduce this discriminability. Since decreasing the number of clusters can have both a positive and a negative effect on accuracy, the optimal number of state clusters can be determined experimentally so as to minimize the word error on development test data.

2.2. State Clustering and Speed

Computation of the frame-log-likelihoods for all the Gaussian components in each active triphone state during the Viterbi search is a significant cost affecting recognition speed. In SRI's DECIPHER^{TM} speech recognition system, this cost is reduced using Gaussian caching, Gaussian pruning, and Gaussian shortlists. We now examine how these methods are affected by changing the number of state clusters.

Gaussian Caching In Gaussian caching, we cache the log-likelihoods for the Gaussians in a mixture as soon as they are computed for each frame. If the same Gaussian mixture needs to be evaluated at that frame for another triphone state, the cache is used, rather than recomputing the likelihoods of the Gaussians in this mixture. This results in a significant cost saving as many triphone states share the same Gaussian mixture.

When state clusters are merged, the number of mixtures is reduced,

but the number of Gaussians per mixture is increased. Thus, while fewer Gaussian mixtures will be computed and cached, the number of Gaussians in each will be proportionally larger. Thus, we expect no significant effect due to reducing the number of state clusters on the number of Gaussians computed and cached. However, as we show in the next section, reducing the number of state clusters can decrease the cost of each Gaussian computation.

Gaussian Pruning When computing the set of Gaussians for a state and frame, it is possible to reduce the amount of Gaussian computations by retaining only those Gaussians whose log-likelihoods are within a threshold of the best Gaussian computed so far. By expanding the diagonal covariance Gaussian likelihood computation, it is easy to see that we can decide if a Gaussian is within this threshold *before computing all the distance components* for this frame of speech. This results in a significant reduction in computation cost. Intuitively, the larger the overlap between Gaussians, the larger the number of Gaussians that must be retained for any frame, and the larger the number of distance components that must be computed.

When state clusters are merged to create a model with less tying, the redundant Gaussians in the state cluster overlap region are more effectively used to cover the acoustic space of the clusters. The resulting Gaussians will also have smaller variances, as shown in Figure 2. Since smaller variances imply less Gaussian overlap, we expect the number of Gaussian distance components computed to be reduced.

Gaussian Shortlists Gaussian shortlists are another way to reduce the Gaussian computation during recognition [7]. In this approach, the acoustic space is vector quantized. For each vector quantization (VQ) region, a shortlist of Gaussians that have training data likelihood above some threshold is maintained for each state cluster. During recognition, we find the VQ region corresponding to the frame being evaluated, and only compute the likelihoods for the Gaussians in the corresponding shortlists of the state clusters for that VQ region, resulting in a significant speed-up.

When state clusters are merged to create systems with fewer clusters and more tying, the Gaussian variances are reduced, as in Figure 2. The reduced variance results in less coverage of the acoustic space by each Gaussian. Thus, Gaussians that previously belonged in a shortlist for a VQ region may no longer have likelihoods high enough to belong in the shortlist for that region. Thus, we expect a reduction in the size of the shortlists when we decrease the number of state clusters, and a corresponding reduction in Gaussian computation.

3. EXPERIMENTAL RESULTS

We experimented using the Wall Street Journal (WSJ) database. For training, we used 18,000 SI-284 male training sentences, and for testing we used three different WSJ-based test sets. Each test set had 10 speakers, and consisted of about 3600 words, for a total of about 10,900 words. The WSJ domain has been used in previous U.S. Government-sponsored speech recognition evaluations. The test sets we used were created for internal development, and are not standardized test sets from the WSJ domain. A 20,000-word bigram language model (LM) was used for recognition. We refer to the three test sets as WSJ1, WSJ2, and WSJ3.

System	Word Error Rate (%)		
	WSJ1	WSJ2	NABN
State-clustered	21.65	14.08	18.29
PTM	20.49	12.58	16.78

Table 1: Word error rates for different levels of tying

System	Shortlist Size
State-clustered	5830534
PTM	2773199

Table 2: Shortlist size for different levels of tying

We compared two different systems with different levels of tying. The first is a state-clustered system with 937 clusters and 32 Gaussians per cluster. We chose this baseline configuration because it has given us good performance in the past. The second is a 38-class PTM system with 789 Gaussians per class. Notice that both systems have a total of about 30,000 Gaussians. Both these systems were trained using the Gaussian Merging Splitting (GMS) algorithm that we recently developed [10]. This method computes only as many Gaussians as can be robustly estimated given the amount of training data, thus giving reliable models. Table 1 compares the word error rates for the two systems on the three different test sets. It is clear that the PTM system is significantly more accurate than the state-clustered system on all three test sets. In particular, the word error rate is reduced by 5 to 10%.

In Table 1, we did not use Gaussian shortlists. For the remaining experiments, we used Gaussian shortlists and only used the WSJ1 test set. In Table 2, we compare the size of the Gaussian shortlists for the state-clustered and the PTM systems. Here "size" refers to the number of Gaussians in the shortlists. The number of Gaussians in the PTM system shortlists is half that in the state-clustered shortlists.

Next, we conducted experiments to evaluate the effect of clustering on recognition computation and speed. We did this by varying the level of pruning in the Viterbi beam search and plotting the word error rate for the WSJ1 test set against different parameters of interest. These are the number of Gaussians we start computing per frame, the number of actual distance components computed, and the recognition speed of our system. While the first two parameters are an objective measure of the Gaussian computation cost incurred during recognition, the system speed is implementation-dependent. Figures 3, 4, and 5 show these plots.

It is clear from these figures that a significant computation saving is gained by using the PTM system over the state-clustered system. At a word error rate of 22%, the PTM system has about a factor of 2 less Gaussians started, a factor of 2 less distance component computations, and a factor of 5 speed-up. Further, at almost all speeds, the PTM system has a lower word error rate, as shown in Figure 5. In all three figures we notice that at very high error rates, the PTM system is worse in terms of Gaussian computation and speed (where the curves cross). This occurs because at these error

rates, there are only a few active hypotheses in the search per frame, requiring the computation of only a few mixtures. The fact that the state-clustered system has only 32 Gaussians per state cluster as compared to 789 Gaussians for the PTM system then outweighs the computational benefits of the PTM model described in Section 2.2. However, we do not anticipate operating in this high word error region of the curve.

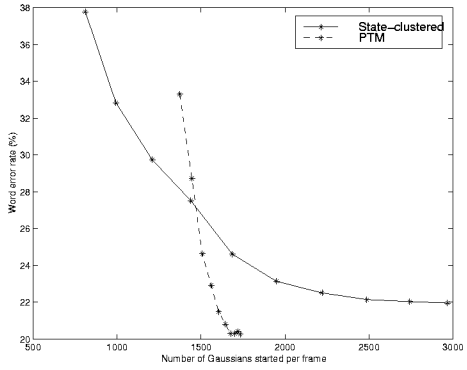

Figure 3: Word error vs. number of Gaussians started

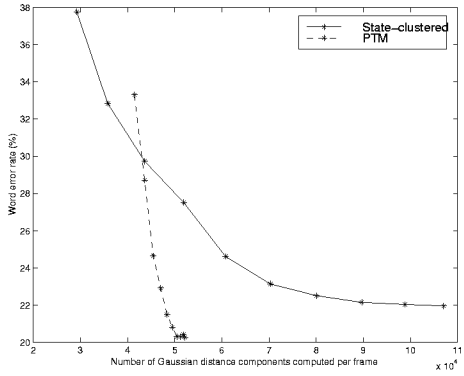

Figure 4: Word error vs. number of Gaussian distance components computed

4. SUMMARY

We provided a new view of parameter tying in HMM-based speech recognition systems. We showed that PTM systems, if properly trained, can significantly outperform the currently dominant state-clustered HMM-based approach. In particular, we achieved between 5 and 10% reduction in the word error rate. The number of Gaussians in the shortlists was reduced by half. Finally, at a fixed word error rate, we achieved a factor of 2 reduction in Gaussian distance computation during recognition, and a factor of 5 speed-up.

To the best of our knowledge, this is the first paper that shows a significant performance gain in accuracy, computation, and speed by using PTM systems as opposed to state-clustered systems.

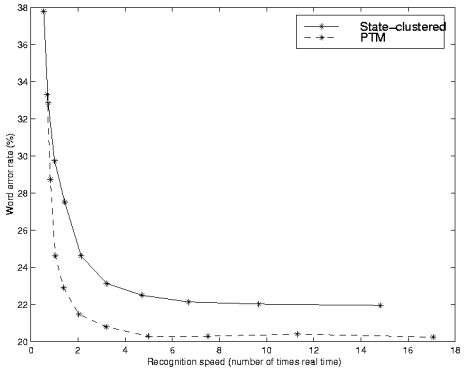

Figure 5: Word error vs. recognition speed

5. REFERENCES

1. X. Huang and M. Jack, "Semi-Continuous Hidden Markov Models for Speech Signals," *Computer Speech and Language*, vol. 3, pp. 239–252, 1989.
2. J. Bellagarda and D. Nahamoo, "Tied Mixture Continuous Parameter Modeling for Speech Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, pp. 2033–2045, December 1990.
3. D. Paul, "The Lincoln Robust Continuous Speech Recognizer," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 449–452, May 1989.
4. M.-Y. Hwang, X. Huang, and F. Alleva, "Predicting Unseen Triphones with Senones," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. II–311 – II–314, 1993.
5. P. Woodland, J. Odell, V. Valtchev, and S. Young, "Large Vocabulary Continuous Speech Recognition Using HTK," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. II–125 – II–128, 1994.
6. V. Digalakis and H. Murveit, "High-Accuracy Large-Vocabulary Speech Recognition Using Mixture Tying and Consistency Modeling," in *Proceedings of the DARPA Human Language Technology Workshop*, (Plainsboro, NJ), 1994.
7. 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.
8. O. Kimball and M. Ostendorf, "On the Use of Tied-Mixture Distributions," in *Proceedings of the DARPA Human Language Technology Workshop*, (Plainsboro, NJ), 1993.
9. D. B. Paul, "The Lincoln Large-Vocabulary Stack-Decoder Based HMM CSR," in *Proceedings of the DARPA Human Language Technology Workshop*, pp. 399–404, 1994.
10. 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.