Design of Compact Acoustic Models through Clustering of Tied-Covariance Gaussians

Mark Z. Mao[†⋆] *and Vincent Vanhoucke*[⋆]

[†] Department of Electrical Engineering, Stanford University, CA, USA
[⋆] Nuance Communications, Menlo Park, CA, USA
markmao@stanford.edu, vincent@nuance.com

Abstract

We propose a new approach for designing compact acoustic models particularly suited to large systems that combine multiple model sets to represent distinct acoustic conditions or languages. We show that Gaussians based on mixtures of inverse covariances (MIC) with shared parameters can be clustered using an efficient Lloyd algorithm. As a result, more compact acoustic models can be built by clustering Gaussians across tied mixtures. In addition, we show that the tied parameters of MIC models can be shared across acoustic models and languages, making it possible to build more efficient multi-model systems which take advantage of a common pool of clustered Gaussians.

1. Introduction

In order to build accurate automatic speech recognition (ASR) systems, it is necessary to have large acoustic models that can represent the statistics of a large number of phonetic states accurately. In addition, if the system is to be used in multi-channel or multilingual environments, it is common to combine several acoustic models trained independently, each dealing with a specific acoustic/channel condition or language, into a single ASR system. However, it is difficult to have both large acoustic models and more acoustic condition/language specific acoustic models at the same time since the total size and computational cost of the system can become prohibitive.

A natural way of addressing the issue of acoustic model size/complexity is by tying the parameters used to describe acoustic probabilities. Assuming that the acoustic model uses Gaussian Mixture Models (GMM) in a Hidden Markov Models (HMM) framework, this tying can be performed at several different levels: the Gaussian parameters (means and covariances), the Gaussian densities, the GMM, HMM states, or even larger sub-word units.

In the context of multilingual applications, much work has been focused on sharing at the HMM state level and above [1, 2, 3, 4]. It has been shown, however, that features of acoustic models at the Gaussian and GMM level can be efficiently ported from one language to another [5], and that acoustic features are strongly correlated across languages [6], which suggests that there are sufficient commonalities across languages for an efficient sharing of parameters at the Gaussian level and below.

The difficulty resides in the fact that typical parameter sharing techniques are often derived from the HMM structure, by sharing mixtures at the level of a phone [7] or state cluster [8], or sub-state level [9]. However, if the parameters belong to heterogeneous HMMs, which is the case if they belong to acoustic models trained completely independently, the tying has to be performed independently of this structure. The same argument also holds for distinct GMMs within a model set: redundancies in the model which are not captured by the phonetic structure of the language can only be taken advantage of by using a parameter tying approach which is independent of the phonetic structure.

In this paper, we will focus on the combination of two sharing techniques which operate at the level of the Gaussians and the Gaussian parameters, and which make no assumption as to which mixture or model set the Gaussians belong to. We will show that those two techniques work both for sharing Gaussian parameters within a single acoustic model set and for sharing across model sets. In [10], it was shown that, within a given acoustic model, covariance parameters could be tied using a MIC model and achieve a level of accuracy similar to a full covariance model, but at a memory and CPU cost comparable to what those of a diagonal covariance model. Based on this shared covariance structure, we show in Section 2 how clustering of the Gaussians can be performed to reduce the total number of parameters in the system, and demonstrate experimentally in Section 3.2 the efficiency of the method. In Sections 3.3 and 3.4, we show that the tied parameters of a MIC model can be shared across heterogeneous acoustic models and languages, and demonstrate experimentally the benefits of clustering Gaussians across those model sets.

2. Clustering of MIC Parameters

Vector quantization (VQ [11]) is a well-known technique used for Gaussian clustering [12, 13, 14]. In the following, we extend the VQ algorithm to the case of Gaussians tied using a MIC model. A GMM for a D-dimensional input vector o, composed of M Gaussians with priors w_i, means μ_i and covariances Σ_i can be expressed as:

$$f(o) = \sum_{i=1}^{M} w_i \mathcal{N}(o, \mu_i, \Sigma_i) \tag{1}$$

A mixture of inverse covariances [10] is defined by set of K prototype symmetric matrices Ψ_k, shared across the model set, such that for each Gaussian i there is a vector Λ_i with components $\lambda_{k,i}$ satisfying:

$$\Sigma_i^{-1} = \sum_{k=1}^{K} \lambda_{k,i} \Psi_k \tag{2}$$

Each Gaussian in the MIC model is uniquely determined by $D + K$ parameters, as opposed to the $D(D + 3)/2$ parameters needed to represent a full covariance Gaussian. In order to quantize the Gaussians, the Lloyd algorithm can be more efficiently applied in this lower dimensional space. First, a suitable distortion measure needs to be defined between the Gaussians to be clustered and the cluster centroids. Second, the computation of the Lloyd centroid minimizing the divergence in each cluster needs to be derived for Gaussians tied using a MIC model.

2.1. Divergence Measure

The Kullback-Leibler (KL) divergence is a natural measure of the divergence between two probability densities. The divergence between two densities f and g can be expressed as:

$$d(f, g) = \int f(x) \log \frac{f(x)}{g(x)} dx \tag{3}$$

It is also common to use a symmetric form of this distortion measure [15], of which a simple instance is:

$$d_s(f, g) = d(f, g) + d(g, f) \tag{4}$$

Both distance measures are known to perform well on Gaussian clustering tasks [12, 13].

2.2. Clustering using the Lloyd algorithm

The Lloyd algorithm [11] is a procedure in which two distinct optimizations are applied alternatively to globally minimize the distortion between the data and its quantized representation.

In the first (*nearest neighbor*) step, the centroid closest to every data point is determined by evaluating the distance between each pair. The second (*centroid*) step, determines the best centroid given the nearest neighbor assignment. Given M Gaussians belonging to a given cluster with centroid h, and the divergence δ ($\delta = d$ or d_s), the total divergence to be minimized is:

$$Q_\delta = \frac{1}{\gamma} \sum_{m=1}^{M} w_m \delta(f_m, h) \tag{5}$$

where w_m is the prior of each Gaussian, and $\gamma = \sum_{m=1}^{M} w_m$. In the case of Gaussians within a GMM/HMM structures, those priors can be determined from the state occupancy probabilities. This divergence can be minimized by setting its partial derivatives with respect to the centroid parameters to zero.

When $\delta = d$, and the covariance of the centroid distribution is not constrained, it is easy to see that the mean and covariance of the centroid are:

$$\mu_c^d = \frac{1}{\gamma} \sum_{m=1}^{M} w_m \mu_m \tag{6}$$

$$\Sigma_c^d = \frac{1}{\gamma} \sum_{m=1}^{M} w_m [\Sigma_m + (\mu_m - \mu_c)(\mu_m - \mu_c)^\top] \tag{7}$$

The case of $\delta = d_s$ has been addressed in [13].

However, when the covariance parameters of the Gaussians are tied using a MIC model, it is obvious from Equation 7 that the unconstrained minimum distortion centroid covariance can not in general be expressed as a tied covariance using the same prototypes Ψ_k. This is also the case when using the symmetric distortion measure.

Instead of minimizing Q_δ with respect to the untied covariance parameters, the minimization needs to be performed with respect to the MIC weights of the centroid Λ_c.

2.3. Centroid Optimization using $\delta = d$

In the case of $\delta = d$, the optimization of the mean vector is unaffected (Equation 6). The covariance optimization can be shown to be equivalent to finding the maximum likelihood estimates of the MIC weights Λ_c, assuming that the centroid has the sample covariance provided by Equation 7. This optimization problem can be solved as in [10, Section V.B] or [16].

2.4. Centroid Optimization using $\delta = d_s$

In the case of $\delta = d_s$, [13] showed that the unconstrained problem could be solved by iterating through optimization of the centroid mean μ_c given its covariance Σ_c using:

$$\mu_c = [\sum_{m=1}^{M} w_m (\Sigma_m^{-1} + \Sigma_c^{-1})]^{-1} \sum_{m=1}^{M} w_m (\Sigma_m^{-1} + \Sigma_c^{-1}) \mu_m \tag{8}$$

and solving a Riccati equation optimizing the covariance of the centroid given its mean. In the context of a MIC model, Equation 8 still applies, with:

$$\Sigma_c^{-1} = \sum_{k=1}^{K} \lambda_{k,c} \Psi_k \tag{9}$$

Let us define the expected inverse covariance $\bar{\Sigma}_c^{-1}$ as:

$$\bar{\Sigma}_c^{-1} = \sum_{k=1}^{K} \bar{\lambda}_{k,c} \Psi_k, \text{ with } \bar{\lambda}_{k,c} = \frac{1}{\gamma} \sum_{m=1}^{M} w_m \lambda_{k,m} \tag{10}$$

It can be shown that the total divergence Q_{d_s} is a convex function of the MIC weights of the centroid Λ_c. The covariance parameter optimization given the centroid mean can be expressed as the solution of a convex problem, which amounts to solving the following system of equations for $1 \leq i \leq K$:

$$\frac{\partial Q_{d_s}}{\partial \lambda_{i,c}} = \frac{1}{2} \mathrm{Tr}\{\Psi_i (\Sigma_c^d - \Sigma_c \bar{\Sigma}_c^{-1} \Sigma_c)\} = 0 \tag{11}$$

with Σ_c^d as in Equation 7. This system can be solved using standard convex optimization techniques. Given μ_c, the component of the Hessian with respect to $(\lambda_{i,c}, \lambda_{j,c})$ is:

$$\frac{\partial^2 Q_{d_s}}{\partial \lambda_{i,c} \partial \lambda_{j,c}} = \frac{1}{2} \mathrm{Tr}\{(\Psi_i \Sigma_c \Psi_j + \Psi_j \Sigma_c \Psi_i) \Sigma_c \bar{\Sigma}_c^{-1} \Sigma_c\} \tag{12}$$

Hence, the optimal covariance given the mean can be computed efficiently through Newton iterations. The algorithm can be initialized using the expected mean in Equation 6 and the expected inverse covariance in Equation 10, which guarantees a positive definite initial estimate.

2.5. Implementation

When a large number of high-dimensional Gaussians are involved, repeated evaluation of the distance measure during the nearest neighbor step can be computationally expensive. However, when the Gaussians in the model share the same MIC prototypes Ψ_k, both distance computations can be expressed in a simple way as a function of:

$$\mu, \ \log |\Sigma|, \ \Sigma^{-1}\mu, \ \mu^\top \Sigma^{-1} \mu, \ \mu^\top \Psi_k \mu, \ \mathrm{Tr}(\Psi_k \Sigma) \tag{13}$$

Precomputation of those $2(D+K+1)$ quantities for each Gaussian makes a significant portion of the computational cost be proportional to the number of Gaussians (M), as opposed to M times the number of centroids. The nearest-neighbor step is also a computation very suitable to parallelizing on several machines. In addition, the distance between Gaussians is typically dominated by the distance between the means. Centroid candidates can be quickly discarded after computing a lower bound on their distance to the Gaussian considered, and comparing this bound to the smallest distance computed up to that point [11, Section 12.16]. A good lower bound on the symmetric distortion, which comes as a partial result of the distance computation, is:

$$\frac{1}{2}(\mu_f - \mu_g)^\top(\Sigma_f^{-1} + \Sigma_g^{-1})(\mu_f - \mu_g) \leq d_s(f, g) \qquad (14)$$

The distance to the centroid which was the closest to the Gaussian considered at the previous VQ iteration can be computed first, providing a tight smallest distance to compare that lower bound to, and allows the algorithm to skip most distance computations. The centroid step, because it involves carrying out a series of convex optimizations, is also expensive, but simple to parallelize. In addition, a given best centroid is uniquely determined by the Gaussians allocated to its cluster, which implies that if a cluster has not changed after an iteration, there is no need to reestimate the cluster centroid, which significantly speeds up the final iterations of the VQ algorithm.

3. Experiments

3.1. Experimental Setup

The experiments were run on speaker-independent American English and Italian systems. The recognition engine used is a context-dependent HMM system with 14000 triphones for English, 4500 triphones for Italian and tied mixtures based on Genones [8]: each state cluster shares a common set of Gaussians called Genone, while the mixture weights are state-dependent. The system uses 2000 Genones for American English and 1000 Genones for Italian, and 32 Gaussians per Genone in both cases. The size of each model set has been optimized with respect to the amount of training data used (370 hours for English, 120 hours for Italian). The features are 27 dimensional, uncluding MFCC, Δ and $\Delta\Delta$. The MIC model uses 41 prototypes. The test-set is a collection of 26000 American English and 9600 Italian utterances of telephony speech spanning several tasks, including digits, proper names and command lists. The accuracy is evaluated using a sentence understanding error rate, which measures the proportion of utterances in the test-set that were interpreted incorrectly semantically.

3.2. Clustering across tied mixtures, within model set

In the following experiments, we compare the reduction of the number of Gaussians in an American English model set using three different methods: reducing the number of tied mixtures, reducing the number of Gaussians per mixture, or clustering the Gaussians across tied mixtures.

Table 1: *Error rate as a function of the clustering method.*

Experiment	Genones × Gauss.	Error Rate
Baseline	2000 × 32	7.8%
Fewer Genones	1000 × 32	8.2%
Fewer Gauss./Gen.	2000 × 16	8.7%
VQ 32K Gaussians	(2000) × (32)	8.0%

Table 1 shows the performance of system when reducing the number of Gaussians by a half. While in both experiments using fewer Gaussians or Genones the acoustic model was entirely retrained on the same data, the VQ clustering was performed directly on the baseline model set, without requiring any form of retraining. Either distortion measure (d or d_s) results in the same performance. Clustering Gaussians across Genones performs significantly better than both other methods of reducing the number of Gaussians. Table 2 shows the performance of the system when further reducing the number of Gaussians using VQ clustering across Genones, compared to reducing the number of Genones. Even when reducing the number of Gaussians by as much as 75%, the error rate increase of the system is only a relatively small 5%.

Table 2: *Sharing Gaussians across Genones.*

Number of Gaussians	Fewer Genones	Clustering	
		d_s	d
64000	7.8%		
32000	8.2%	8.0%	8.0%
16000	8.8%	8.2%	8.2%
8000	9.5%	9.2%	9.1%

Since both distance measures performed very similarly, in the following we will only report results using the distance d_s.

3.3. Clustering across model sets

Table 3 compares two systems, both combining three model sets (Gender Independent (GI) / Male / Female) using 64000 Gaussians each in a two-pass approach. The first system was trained using a separately optimized set of MIC prototypes for each model set, while in the second one the prototypes were trained on the GI model set and used on all three. Since there is no accuracy penalty for such sharing, it is natural to consider clustering the Gaussians in that system across all model sets.

Table 3: *Sharing MIC prototypes across acoustic models.*

Prototypes	Distinct	Shared
Error Rate	6.4%	6.4%

Table 4 shows the performance of such clustered system. The systems where Gaussians are clustered jointly across acoustic model sets (fourth column), perform better, with any given total number of Gaussians, than systems where the Gaussians are clustered independently within each model set (third column). The number of Gaussians, in this latter case, was kept identical in each acoustic model (32K, 16K and 8K). Similarly to what was observed in Section 3.2, the number of Gaussians could be reduced by 75% at a cost in accuracy of less than 5%.

Table 4: *Sharing Gaussians across acoustic models.*

Experiment	Gaussians	Not Shared	Shared
Baseline	192000	6.4%	
VQ	96000	6.5%	6.4%
	48000	6.8%	6.7%
	24000	7.6%	7.2%

3.4. Clustering across languages

Table 5 compares two Italian systems: The first one uses its own set of MIC prototypes, while the second uses prototypes trained for the English model set. The second system's error rate is within a relative 3% of the error rate of the standalone system, which suggests that Gaussian clustering can also be used to cluster Gaussians across languages.

Table 5: *Sharing MIC prototypes across languages.*

Prototypes	Italian	English
Error Rate (Italian)	8.3%	8.5%

In the following experiment, two very different model sets are combined: the first one is the American English acoustic model used in section 3.2, and the other a smaller Italian acoustic model trained on much less data. Both share the same front-end processing, but were otherwise trained independently. Table 6 shows the performance of the system when clustering the Gaussians either separately (columns 3 and 5) jointly (columns 4 and 6). When sharing the Gaussians across languages, the total prior of each language was normalized. In the case of a separate clustering, the ratio between the number of Gaussians in each model set (2 to 1 for the English model) was preserved.

Table 6: *Sharing Gaussians across languages.*

Language		English		Italian	
Exp.	Gauss.	Not Sh.	Shared	Not Sh.	Shared
Base.	96000	7.8%		8.5%	
VQ	48000	8.0%	8.0%	8.9%	8.7%
	24000	8.2%	8.1%	10.0%	9.6%
	12000	9.2%	9.1%	11.1%	10.6%

On English tasks, both setups performed very similarly, which shows that the clustering of the better trained models with a coarser model from a distinct language didn't affect the performance of the English model adversely. The performance on Italian tasks was much improved by the sharing, which shows that the Italian acoustic model was able to leverage efficiently the Gaussians trained on the English data to compensate for the reduced parametric complexity.

4. Conclusion

We have introduced a new approach for designing compact acoustic models through clustering of Gaussians with shared tied MIC parameters. We showed that VQ can be efficiently applied to Gaussians sharing MIC prototypes to vastly reduce the number of parameters in an acoustic model. We showed that sharing MIC prototypes across model sets and languages was possible for a given front-end, and that the total number of Gaussians in multi-model systems could be substantially reduced at a low cost in accuracy.

5. Acknowledgments

The authors would like to thank R.M. Gray, J.F. Crespo, A. Sankar, and B. Strope for their input. This research was supported in part by NSF grant CCR-0309701 and by Nuance Communications.

6. References

[1] Vergyri, D., Tsakalidis, S. and Byrne, W., "Minimum Risk Acoustic Clustering for Multilingual Acoustic Model Combination," *Proceedings of ICASSP 2000*, Istanbul, Turkey, 2000.

[2] Byrne, W., Beyerlein, P., Huerta, J. M., Khudanpur,S., Marthi, B., Morgan, J., Peterek, N., Picone, J., Vergyri, D. and Wang, W., "Towards Language Independent Acoustic Modeling," *Proceedings of ICASSP 2000,* Istanbul, Turkey, 2000.

[3] Kanthak, S. and Ney, H., "Multilingual Acoustic Modeling Using Graphemes," *Proceedings of Eurospeech 2003*, Geneva, Switzerland, 2003.

[4] Wang, Z., Topkara, U., Schultz, T. and Waibel, A., "Towards Universal Speech Recognition," *Proceedings of the International Conference on Multimodal Interfaces (ICMI-2002)*, Pittsburgh, PA, USA, October 14-16, 2002.

[5] Beaufays, F., Boies, D. and Weintraub, M., "Porting Channel Robustness Across Languages," *Proceedings of ICSLP 2002*, Denver, CO, USA, Sept. 2002.

[6] Stüker, S., Schultz, T., Metze, F. and Waibel, A., "Multilingual Articulatory Features," *Proceedings of ICASSP 2003*, Hong Kong, China, April 2003.

[7] Sankar, A. and Gadde, V. R. R., "Parameter Tying and Gaussian Clustering for Faster, Better, and Smaller Speech Recognition," *Proceedings of Eurospeech 1999*, vol. 4, pp. 1711–1714, Budapest, Hungary, 1999.

[8] Digalakis, V., Monaco, P. and Murveit, H., "Genones: Generalized mixture tying in continuous hidden Markov model-based speech recognizers," *IEEE Trans. on Speech and Audio Processing*, vol. 4, no. 4, pp. 281–289, 1996.

[9] Gu, L., and Rose, K., "Sub-state Tying in Tied Mixture Hidden Markov Models," *Proceedings of ICASSP 2000*, Istanbul, Turkey, 2000.

[10] Vanhoucke, V. and Sankar, A., "Mixtures of Inverse Covariances," *IEEE Trans. on Speech and Audio Processing*, vol. 12, no. 3, May 2004.

[11] Gersho, A. and Gray, R. M., *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1992.

[12] Gray, R. M., " Gauss mixtures quantization: clustering Gauss mixtures," *Proceedings of the Math Sciences Research Institute Workshop on Nonlinear Estimation and Classification, Mar. 17–29, 2002*, Denison, D. D., Hansen, M. H., Holmes, C. C., Mallick, B. and Yu, B. Eds. pp. 189–212, Springer, New York, 2003.

[13] Myrvoll, T. A. and Soong, F. K., "On Divergence Based Clustering of Normal Distributions and Its Application to HMM Adaptation," *Proceedings of Eurospeech 2003*, Geneva, Switzerland, 2003.

[14] Rigazio, L., Tsakam, B. and Junqua, J.-C., "An Optimal Bhattacharyya Centroid Algorithm for Gaussian Clustering with Applications in Automatic Speech Recognition," *Proceedings of ICASSP 2000*, Istanbul, Turkey, 2000.

[15] Johnson, D. H. and Sinanović, S., "Symmetrizing the Kullback- Leibler Distance," (Unpublished, March 2001), http://www.ece.rice.edu/˜dhj/, 2001.

[16] Axelrod, S., Gopinath, R. and Olsen, P., "Modeling with a subspace constraint on inverse covariance matrices," *Proceedings of ICSLP 2002*, pp. 2177–2180, Denver, CO, USA, Sept. 2002.