REFACTORING ACOUSTIC MODELS USING VARIATIONAL DENSITY APPROXIMATION

Pierre L. Dognin, John R. Hershey, Vaibhava Goel, Peder A. Olsen

IBM T. J. Watson Research Center

{pdognin, jrhershe, vgoel, pederao}@us.ibm.com

ABSTRACT

In model-based pattern recognition it is often useful to change the structure, or *refactor*, a model. For example, we may wish to find a Gaussian mixture model (GMM) with fewer components that best approximates a reference model. One application for this arises in speech recognition, where a variety of model size requirements exists for different platforms. Since the target size may not be known *a priori*, one strategy is to train a complex model and subsequently derive models of lower complexity. We present methods for reducing model size without training data, following two strategies: GMM approximation and Gaussian clustering based on divergences. A variational expectation–maximization algorithm is derived that unifies these two approaches. The resulting algorithms reduce the model size by 50% with less than 4% increase in error rate relative to the same–sized model trained on data. In fact, for up to 35% reduction in size, the algorithms can *improve* accuracy relative to baseline.

Index Terms— Acoustic model clustering, KL divergence, Bhattacharyya divergence, variational approximations.

1. INTRODUCTION

A problem that arises in probabilistic modeling is to approximate one model using another model with a different structure (fewer parameters, different parameter sharing, etc.). For example, a common task in automatic speech recognition (ASR) is to reduce the number of components in a Gaussian mixture model (GMM) with a minimal loss of accuracy. Whereas state-of-the-art ASR systems require increasingly large acoustic models, commercial applications have a variety of model size requirements, ranging from server-based applications that can accommodate large models, to embedded applications with modest memory capacities. Although it is possible to train models of any given size, the desired size may not be known at training time. It would be convenient to adapt an existing model to different memory requirements, without having to revisit the training data.

To address this, we optimize the parameters of a *refactored* acoustic model to best match a larger *reference* model. One approach is to minimize divergence, such as the Kullback-Leibler (KL) [1] or Bhattacharyya [2] divergences, between the probability density functions (pdfs) of the reference and refactored models. To this end we introduce a variational expectation–maximization algorithm that minimizes the KL-divergence between the refactored and reference models. Another approach is to cluster the components of the reference model, based on their pair-wise divergences. Both methods use a maximum likelihood merging criterion, and reduce the size of the acoustic model without significant loss of accuracy. In fact, we show that for modest reductions in size the word error rate (WER) can even improve.

2. MODELS

Acoustic models are typically composed of phonetic states with observation models that are dependent on the phonetic context. The observation models are GMMs of the observed acoustic features. Diagonal covariance Gaussians are used as computation time and storage is greatly reduced compared to full covariance Gaussians. Reducing the overall size of our model requires reducing the number of components used by some or all of the GMMs. We first consider ways of reducing each GMM to a given size, before turning to the problem of choosing the number of Gaussians allocated to each GMM in order to reach the targeted total number of Gaussians for the whole acoustic model.

Let us consider a GMM f with continuous observation $\boldsymbol{x} \in \mathbb{R}^d$,

$$f(\boldsymbol{x}) = \sum_a \pi_a f_a(\boldsymbol{x}) = \sum_a \pi_a \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_a; \boldsymbol{\Sigma}_a), \tag{1}$$

where a indexes components of f, π_a is the prior probability, and $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_a; \boldsymbol{\Sigma}_a)$ is a Gaussian in $\boldsymbol{x}$ with mean vector $\boldsymbol{\mu}_a$ and covariance matrix $\boldsymbol{\Sigma}_a$.

3. DIVERGENCE MEASURES

The KL divergence [1] is commonly used to measure the dissimilarity of two pdfs $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$,

$$D_{\mathrm{KL}}(f||g) \stackrel{\mathrm{def}}{=} \int f(\boldsymbol{x}) \log \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} \tag{2}$$

$$= L(f\|f) - L(f\|g), \tag{3}$$

where $L(f\|g)$ is the expected log likelihood of g under f

$$L(f\|g) \stackrel{\mathrm{def}}{=} \int f(\boldsymbol{x}) \log g(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \tag{4}$$

For two Gaussians $f_i(\boldsymbol{x})$ and $f_j(\boldsymbol{x})$ from GMM $f(\boldsymbol{x})$ with $\boldsymbol{x} \in \mathbb{R}^d$, $D_{\mathrm{KL}}(f_i, f_j)$ has a closed–form expression:

$$\begin{aligned} D_{\mathrm{KL}}(f_i||f_j) = \frac{1}{2}\big[\log \frac{|\boldsymbol{\Sigma}_j|}{|\boldsymbol{\Sigma}_i|} + \mathrm{Tr}(\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\Sigma}_i - \boldsymbol{I}_d) \\ + (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \big]. \end{aligned} \tag{5}$$

The KL divergence is not symmetric as $D_{\mathrm{KL}}(f||g) \neq D_{\mathrm{KL}}(g||f)$. It reaches a minimum for $f = g$ when $D_{\mathrm{KL}}(f||g) = 0$ and is always positive as $D_{\mathrm{KL}}(f||g) \geq 0 \, \forall f, g$.

The Bhattacharyya error bound is also a commonly used similarity measure [2],

$$B(f, g) \stackrel{\mathrm{def}}{=} \frac{1}{2} \int \sqrt{f(\boldsymbol{x}) g(\boldsymbol{x})} \mathrm{d}\boldsymbol{x}, \tag{6}$$

from which the Bhattacharyya divergence is derived as

$$D_{\mathrm{B}}(f,g) \stackrel{\mathrm{def}}{=} -\log 2B(f,g). \tag{7}$$

For two Gaussians $f_i(\boldsymbol{x})$ and $f_j(\boldsymbol{x})$, the Bhattacharyya divergence has a closed–form expression [3]:

$$\begin{aligned} D_B(f_i,f_j) = &\frac{1}{8}(\boldsymbol{\mu}_j-\boldsymbol{\mu}_i)^T\left(\frac{\boldsymbol{\Sigma}_i+\boldsymbol{\Sigma}_j}{2}\right)^{-1}(\boldsymbol{\mu}_j-\boldsymbol{\mu}_i) \\ &+\frac{1}{2}\log\left|\frac{\boldsymbol{\Sigma}_i+\boldsymbol{\Sigma}_j}{2}\right| - \frac{1}{4}\log|\boldsymbol{\Sigma}_j\boldsymbol{\Sigma}_i|. \end{aligned} \tag{8}$$

$B(f,g)$ is symmetric and, if $f=g$, then $B(f,g)=\frac{1}{2}$. Therefore $D_{\mathrm{B}}(f,g)$ is also symmetric, $D_{\mathrm{B}}(f,g)=0$ if and only if $f=g$ and $D_{\mathrm{B}}(f,g)\geq 0$ otherwise.

For GMMs, unfortunately, there are no closed–form expressions for either the KL or the Bhattacharyya divergence. Thus, to optimize based on divergence, we have to either use algorithms that depend only on divergence between individual Gaussians or use approximations to the divergences between GMMs such as [4, 5].

4. VARIATIONAL EXPECTATION-MAXIMIZATION

To optimize the similarity between the refactored model g, with parameters $\{\pi_b, \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b\}$ for each component b, and the reference model f, we first consider minimizing the KL divergence $D_{\mathrm{KL}}(f\|g) = L(f\|f) - L\big(f\|g\big)$. Using the variational approximation in [4] leads to a simple expectation–maximization (EM) algorithm to minimize the divergence. Ignoring the constant term $L(f\|f)$, we maximize the variational lower bound on $L\big(f\|g\big)$. Defining variational parameters $\phi_{b|a}>0$, where a indexes components of f and b indexes components of g, such that $\sum_b \phi_{b|a}=1$, and using Jensen's inequality, we obtain

$$\begin{aligned} L\big(f\|g\big) &\stackrel{\mathrm{def}}{=} \sum_a \pi_a \int f_a(\boldsymbol{x}) \log \sum_b \pi_b g_b(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &\geq \sum_a \pi_a \int f_a(\boldsymbol{x}) \sum_b \phi_{b|a} \log \frac{\pi_b g_b(\boldsymbol{x})}{\phi_{b|a}} \mathrm{d}\boldsymbol{x} \\ &\stackrel{\mathrm{def}}{=} \mathcal{L}_\phi\big(f\|g\big). \end{aligned} \tag{9}$$

This is a lower bound on $L\big(f\|g\big)$ for any ϕ, so we get the best bound by maximizing $\mathcal{L}_\phi\big(f\|g\big)$ with respect to ϕ by taking derivatives and using a Lagrange multiplier to enforce normalization. The maximum value gives the *estimation* (E) step:

$$\hat{\phi}_{b|a} = \frac{\pi_b e^{-D_{\mathrm{KL}}(f_a\|g_b)}}{\sum_{b'}\pi_{b'}e^{-D_{\mathrm{KL}}(f_a\|g_{b'})}}. \tag{10}$$

Note that $\hat{\phi}_{b|a}$ is a measure of the affinity between the Gaussians f_a and g_b. For a given ϕ, $\mathcal{L}_\phi\big(f\|g\big)$ is convex with respect to the parameters of g_b; setting derivatives to zero and solving for $\boldsymbol{\mu}_b$, $\boldsymbol{\Sigma}_b$ and π_b, yields the *maximization* (M) step:

$$\boldsymbol{\mu}_b = \frac{\sum_a \pi_a\phi_{b|a}\boldsymbol{\mu}_a}{\sum_a \pi_a\phi_{b|a}}, \tag{11}$$

$$\boldsymbol{\Sigma}_b = \frac{\sum_a \pi_a\phi_{b|a}\left[\boldsymbol{\Sigma}_a + (\boldsymbol{\mu}_a-\boldsymbol{\mu}_b)(\boldsymbol{\mu}_a-\boldsymbol{\mu}_b)^T)\right]}{\sum_a \pi_a\phi_{b|a}}, \tag{12}$$

$$\pi_b = \sum_a \pi_a\phi_{b|a}. \tag{13}$$

Even if all $\boldsymbol{\Sigma}_a$ are diagonal covariance, the maximum likelihood $\boldsymbol{\Sigma}_b$ will in general be full covariance. Constraining the refactored model to have diagonal covariance, this simplifies to

$$\boldsymbol{\Sigma}_b(k,k) = \frac{\sum_a \pi_a\phi_{b|a}\left[\boldsymbol{\Sigma}_a(k,k) + (\boldsymbol{\mu}_a(k)-\boldsymbol{\mu}_b(k))^2\right]}{\sum_a \pi_a\phi_{b|a}},$$

where $k\in 1,\ldots,n$, and $\boldsymbol{\Sigma}_b(k,k)$ is the kth diagonal element of $\boldsymbol{\Sigma}_b$. If we constrain ϕ to be discrete, then the E-step $\hat{\phi}_{b|a}$ selects the set of reference components a to be clustered to form component b. The M–step then computes the maximum-likelihood Gaussian given this selection. This discretization yields a slightly weaker bound used in [6] as a special case. This is also equivalent to K–means clustering of Gaussians using the KL divergence, as in [7]. For different approaches, based on minimizing the mean–squared error between the two density functions, see [8], or based on compression using dimension–wise tied Gaussians optimized using divergences, see [9].

The variational EM algorithm requires an initial refactored model g, with a given number of components. There are many methods to obtain such an initial model. In the interest of efficiency, we explore greedy methods in which Gaussians from the reference model are merged together pair by pair. These algorithms can be seen as performing an alternate version of the E-step above, followed by exactly the same M–step to merge the selected Gaussians.

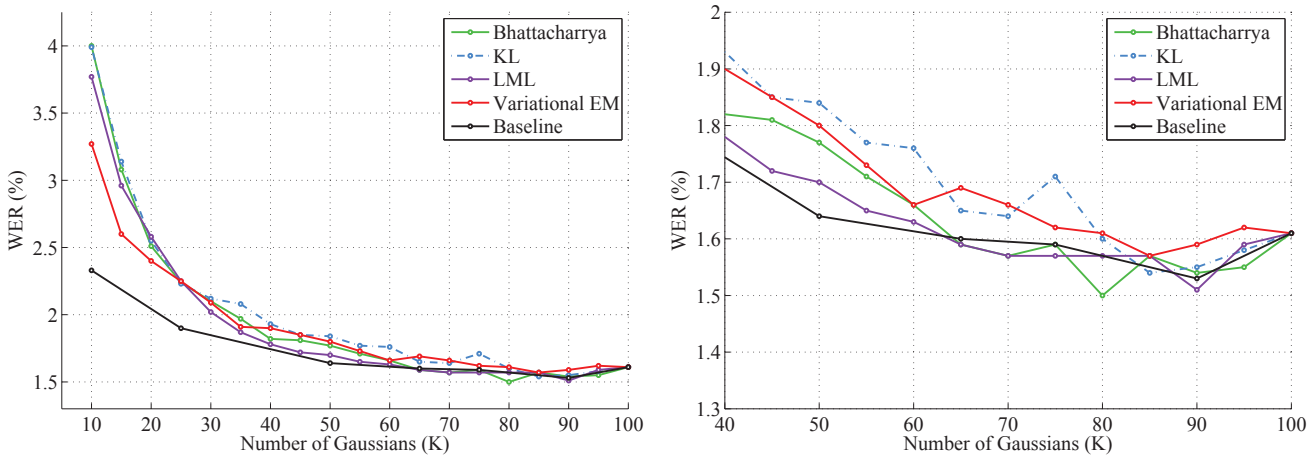
5. GREEDY CLUSTERING

A simple clustering algorithm results from iteratively selecting pairs of Gaussians to merge, based on an objective function that assigns a cost to each potential merge. The pair of Gaussians that minimizes the cost function is selected, and the Gaussians are merged. Merging is accomplished using the M–step (11)–(13), with the appropriate choice of ϕ, as described below. After merging two Gaussians we can treat the resulting model as the reference and iterate. To find the best number of Gaussians for each GMM given a global target, we choose the GMM in which to perform a merge at each iteration based on the cost. Thus, for each GMM f, we compute the cost for all pairs of Gaussians i, j in f, forming a matrix $C_f(i,j)$. Across all GMMs, we find the pair with the minimum cost.

Define a function $f' := \mathrm{merge}(f,i,j)$ taking GMM f and returning a new version of f with components i and j merged into a single Gaussian according to (11)–(13), where we set $\phi_{b|i}=1$ and $\phi_{b|j}=1$, and otherwise $\phi_{b|a}=1$ for $b=a$ and $a\notin i,j$.

Input: Acoustic Model
Output: Clustered Acoustic Model
foreach *GMM* f **do**
  foreach *Gaussian pair* $(i,j)\in f$ **do**
    Compute $C_f(i,j)$;
  end
end
while *Target number of Gaussians not reached* **do**
  foreach *GMM* f **do**
    Find pair with smallest cost $C_f(i,j)$;
    $(f',i',j') := \arg\min_{f,i,j} C_f(i,j)$;
  end
  In GMM f';
  Merge: $f' := \mathrm{merge}(f',i',j')$;
  Update $C_{f'}$;
end

Algorithm 1: A greedy clustering algorithm.



(a) WER results for all clustered and reference models.

(b) WER results for the 40K-100K Gaussians region.

Fig. 1. WER as a function of the number of Gaussians. Results for baseline models (built from data) as well as clustered models using Bhattacharyya, KL divergence, Local Maximum Likelihood and variational EM are plotted.

Note that some cost functions are symmetric, so only $\frac{N(N-1)}{2}$ cost values need be stored in $C_f(i,j)$, where N is the original number of Gaussians in the GMM. This algorithm is well suited to be parallelized since each GMM can be processed independently. Indeed, the only dependency across GMMs is created when looking for the next best pair to merge across all GMMs. However, within a GMM, the sequence of merges will *always* be the same. It is therefore possible to run the merge algorithm on each GMM in parallel all the way to one final Gaussian. The only information to keep is the cost and the indices of the best pairs of Gaussians for each merge. This information can be stored as a *merge–tree* for each GMM f. To generate a merged model, one needs only to search across all merge-trees for the best pair, then again for the next best pair, and so on.

In general any cost function may be used, including the variational divergence between the reference and refactored GMMs mentioned above. In such a case, the greedy algorithm will also improve a bound on the likelihood of the refactored model. However variational KL divergence is a function of all the Gaussians in the GMM and thus can be computationally expensive for every potential merge. Instead we can consider a *local* cost function, that is only a function of the two Gaussians to be merged.

5.1. Pairwise Divergence Clustering

A heuristic local cost function for greedy merging results from using KL or Bhattacharyya divergence between the merged Gaussians. That is, $C_f(i,j) = D_{\mathrm{KL}}(f_i||f_j)$, where f_i, and f_j are two Gaussians from the reference GMM f (and similarly for the Bhattacharyya divergence). The rationale is that the smaller the divergence between the Gaussians, the less distortion there should be in the merged model.

In models with diagonal covariance, this method presents a potential problem. It does not consider whether the full-covariance merged Gaussians may be poorly approximated by a diagonal covariance (or otherwise constrained) Gaussian. Thus, two pairs of Gaussians may have an equal divergence-based cost, despite the fact that one pair results in a good approximation, and the other pair results in a poor one.

5.2. Local Maximum Likelihood (LML) Merging

In order to take into account constraints such as diagonal covariance on the merged Gaussians, here we consider the merged Gaussian in the local cost function. For the Gaussian pair f_i and f_j to be merged, let us use the notation $\tilde{f}_i + \tilde{f}_j$ to refer to a *local* GMM f composed of two weighted Gaussians $\tilde{f}_i = \pi_i' f_i$ and $\tilde{f}_j = \pi_j' f_j$, where $\pi_i' = \pi_i/(\pi_i + \pi_j)$ and $\pi_j' = 1 - \pi_i'$. Let g be the single Gaussian resulting from merging $\tilde{f}_i$ and $\tilde{f}_j$ according to (11)–(13).

The KL divergence $D_{\mathrm{KL}}(\tilde{f}_i + \tilde{f}_j||g)$ is sensitive to how well $\tilde{f}_i + \tilde{f}_j$ is approximated by g. Thus. if g is constrained to be diagonal covariance, and merging f_i and f_j results in full covariances, then the cost will be greater.

This cost function also has a natural interpretation in terms of the likelihood of the refactored model. From (2), the KL divergence between the local GMM $\tilde{f}_i + \tilde{f}_j$ and g

$$D_{\mathrm{KL}}(\tilde{f}_i + \tilde{f}_j||g) = \int (\tilde{f}_i + \tilde{f}_j) \log \frac{\tilde{f}_i + \tilde{f}_j}{g} \quad (14)$$

$$= L(\tilde{f}_i + \tilde{f}_j \| \tilde{f}_i + \tilde{f}_j) - L(\tilde{f}_i + \tilde{f}_j \| g), \quad (15)$$

which is the difference of likelihoods between the local GMM $\tilde{f}_i + \tilde{f}_j$ and the merged Gaussian g. Thus, $D_{\mathrm{KL}}(\tilde{f}_i + \tilde{f}_j||g)$ gives a measure of the loss in likelihood if we were to merge $\tilde{f}_i$ and $\tilde{f}_j$. Therefore, the pair $(\tilde{f}_i, \tilde{f}_j)$ that minimizes (15) locally maximizes the likelihood of the merged model. An alternate version can be obtained using the un–normalized weights π_i and π_j. This would give priority to merging Gaussians with smaller weights. For a given KL distance between $\tilde{f}_i + \tilde{f}_j$ and g, less error is introduced if the Gaussian pair has smaller weights, so this alternative may work better.

No closed–form solution exists for the term $L(\tilde{f}_i + \tilde{f}_j \| \tilde{f}_i + \tilde{f}_j)$ in (15), and in our experiments ignoring this term produced poor results. Fortunately, we can use the variational approximation $D_{\mathrm{variational}}(\tilde{f}_i + \tilde{f}_j||g)$ given in [4].

6. EXPERIMENTS

All experiments were run on internal IBM databases. The training set is composed of 786 hours of US English data, consisting of 10,309 speakers for a total of 803,533 utterances. It consists of in-car speech in various noise conditions, recorded at 0 mph, 30 mph and 60 mph with 16KHz sampling frequency. The test set is 38,905 sentences for a total of 205,788 words. It is a set of 47 different tasks of in-car speech with various US regional accents. It is a superset of the test data described in [10].

The reference model for this paper is a 100K Gaussians model built on the training data. We use a set of 91 phonemes, each modeled with a three-state left to right hidden Markov model. These states are modeled using two-phoneme left context dependencies, yielding a total of 1519 context-dependent (CD) states. The acoustic models for these CD states are built on 40-dimensional features obtained using Linear Discriminant Analysis (LDA). CD states are modeled with 66 Gaussians on average. Training consists of a sequence of 30 iterations of EM algorithm where CD state alignments are re-estimated every few steps of EM.

We built 7 baseline models from training data with 10K, 25K, 50K, 65K, 75K, 90K and 100K Gaussians (our reference model). WERs for the baseline models are given in Table 1.

WER (%) vs. Model Size							
Methods	10K	25K	50K	65K	75K	90K	100K
Baseline	2.33	1.90	1.64	1.60	1.59	1.53	1.61

Table 1. Baseline numbers

The reference WER for 100K Gaussians is 1.61%. For smaller size models, WERs remain within 5% relative over the range of 50K to 100K Gaussians, with significant increases in error rate for 25K Gaussians or fewer.

Using Algorithm 1 from Section 5, the reference model (100K) was clustered down all the way to 10K Gaussians. Intermediate models were saved every 5K Gaussians, i.e. 95K, 90K, ..., 10K. This was carried out for each similarity measure (KL, Bhattacharyya, and LML), resulting in 18 models per condition for a total of 18x3 clustered models. Decoding results with these models as a function of model size are plotted in Figure 1. The Bhattacharyya and LML perform somewhat better than the KL divergence over the 100K to 40K range. Below 40K, the error rates begin to increase, reaching a WER 62% higher relative to baseline for 10K (2.33% to 3.77% for LML). Over the 40K to 100K range, LML follows the baseline results more closely than the other methods.

Variational EM was carried out by initializing the procedure using LML clustered models. The decoding results for these models are also plotted in Figure 1. We note that, below 25K Gaussians, variational EM improves error rates relative to LML, with a 15% relative WER improvement at 10K. Surprisingly, however, the variational EM algorithm does slightly worse than LML for clusterings over 25K, despite the fact that both variational EM and LML are derived from the variational KL divergence. Possibly, this stems from the mismatch between the optimized model and the model used by the recognizer. A standard optimization used in recognition is

$$f(\boldsymbol{x}) = \sum_a \pi_a f_a(\boldsymbol{x}) \approx \max_a \pi_a f_a(\boldsymbol{x}). \tag{16}$$

This *max approximation* is more accurate for GMMs in which Gaussian components overlap less. Such an approximation may favor hard clustering done in the greedy algorithms over soft clustering done by variational EM. This may also explain the better performance of the greedy algorithms. Since the 100K model is trained using the sum rather than the max approximation, it may have significantly overlapping Gaussians. We are investigating this currently by comparing recognition and training using both the max and sum models. In addition we are experimenting with the hard-clustering version of the variational EM algorithm, in which ϕ is constrained to be discrete.

7. CONCLUSION

We have described methods for optimizing a refactored GMM model to best match a reference GMM model. The greedy clustering methods that we introduce are specific to mixture models and tend to work especially well in the case of GMM-based acoustic models. In particular, the performance of LML refactored models closely follows that of models trained from data over a range of model sizes from 100K down to 40K. The variational EM algorithm can be used to optimize models with structures that go beyond GMMs. We envision that the variational EM algorithm will be useful for a variety of other applications.

8. REFERENCES

[1] S. Kullback, *Information Theory and Statistics*, Dover Publications, Mileona, New York, 1997.

[2] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by probability distributions," *Bull. Calcutta Math. Soc.*, vol. 35, pp. 99–109, 1943.

[3] Keinosuke Fukunaga, *Statistical Pattern Recognition*, Academic Press, Inc., San Diego, CA, 1990.

[4] John Hershey and Peder Olsen, "Approximating the Kullback Leibler divergence between gaussian mixture models," in *ICASSP*, Honolulu, Hawaii, April 2007.

[5] Peder Olsen and John Hershey, "Bhattacharyya error and divergence using variational importance sampling," in *ICSLP*, Antwerp, Belgium, August 2007.

[6] Jacob Goldberger and Sam Roweis, "Hierarchical clustering of a mixture model," in *Advances in Neural Information Processing Systems 17*, Lawrence K. Saul, Yair Weiss, and Léon Bottou, Eds., pp. 505–512. MIT Press, Cambridge, MA, 2005.

[7] Raimo Bakis, David Nahamoo, Michael A. Picheny, and Jan Sedivy, "Hierarchical labeler in a speech recognition system," U.S. Patent 6023673. filed June 4, 1997, and issued February 8, 2000.

[8] Kai Zhang and James T. Kwok, "Simplifying mixture models through function approximation," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds., pp. 1577–1584. MIT Press, Cambridge, MA, 2007.

[9] Xiao-Bing Li, Frank K. Soong, Tor André Myrvoll, and Ren-Hua Wang, "Optimal Clustering and Non-Uniform Allocation of Gaussian Kernels in Scalar Dimension for HMM Compression," in *ICASSP*, 2005, vol. 1, pp. 669–672.

[10] Sabine Deligne, Ellen Eide, Ramesh Gopinath, Dimitry Kanefsky, Benoit Maison, Peder Olsen, Harry Printz, and Jan Sedivy, "Low-resource speech recognition of 500–word vocabularies," in *Eurospeech*, 2001.