INTERSPEECH 2009 BRIGHTON

Refactoring Acoustic Models using Variational Expectation-Maximization

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

IBM T.J. Watson Research Center

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

10.21437/Interspeech.2009-78

Abstract

In probabilistic modeling, it is often useful to change the structure, or *refactor*, a model, so that it has a different number of components, different parameter sharing, or other constraints. For example, we may wish to find a Gaussian mixture model (GMM) with fewer components that best approximates a *reference* model. Maximizing the likelihood of the refactored model under the reference model is equivalent to minimizing their KL divergence. For GMMs, this optimization is not analytically tractable. However, a lower bound to the likelihood can be maximized using a variational expectation-maximization algorithm. Automatic speech recognition provides a good framework to test the validity of such methods, because we can train reference models of any given size for comparison with refactored models. We show that we can efficiently reduce model size by 50%, with the same recognition performance as the corresponding model trained from data.

Index Terms: acoustic model clustering, KL divergence, variational approximation, variational expectation-maximization.

1. Introduction

In a variety of applications, it is useful to *refactor* a model by changing the number of components, parameter sharing, or other constraints, while preserving similarity to the original model. For example, in dynamical probabilistic models with continuous and discrete state dynamics, the number of components in the posterior increases over time during inference. To make inference efficient, the posterior must be approximated. Other applications include creating a hierarchy of approximations to a model to speed up the search for the most likely component, or compressing an existing model to reduce its footprint in a constrained computing environment.

Minimizing the KL divergence [1] between the *reference* model and the *refactored* model is equivalent to maximizing the likelihood of the refactored model under the reference model. Unfortunately, this is intractable for models such as Gaussian mixture models (GMMs) without resorting to expensive Monte Carlo techniques. However, it is possible to maximize a variational lower bound to the likelihood [2].

In order to test the validity of such methods, we apply them to an automatic speech recognition (ASR) task where we can train reference models of any given size. ASR is a great framework to experiment with model approximation because acoustic models typically have large number of Gaussian components.

In this framework, a simple refactoring task is to take a reference model $\mathcal{M}$, trained on data with $|\mathcal{M}|$ components, and approximate it with a refactored model $\mathcal{N}$ so that $|\mathcal{N}| < |\mathcal{M}|$. Performance is measured in terms of the difference in recognition results for $\mathcal{M}$ and $\mathcal{N}$. In this paper, we extend the methods of [2], introducing a new *weighted local maximum likelihood* (weighted LML) algorithm. We evaluate the methods more precisely using a more consistent recognition system with fewer approximations. We show that we can efficiently reduce model size by 50%, with the same recognition performance as the corresponding model trained from data, referred to as *trained model* in this paper.

For other approaches, based on minimizing the mean–squared error between the two density functions, see [3], or based on compression using dimension–wise tied Gaussians optimized using symmetric KL divergences, see [4].

2. Models

Acoustic models are typically structured around phonetic states and take advantage of phonetic context while modeling observation. Observation models are usually GMMs of the observed acoustic features. It is customary to use diagonal covariance for efficiency as computation time and storage are greatly reduced compared to using full covariance Gaussians.

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

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

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 $\boldsymbol{\Sigma}_a$. $f(\boldsymbol{x})$ is a probability density function, or *pdf*, represented as a GMM with parameters $\{\pi_a, \boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a\}$. Similarly, $g(\boldsymbol{x})$ will refer to a GMM with parameters $\{\pi_b, \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b\}$ for the rest of this paper.

3. Divergence Measures

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

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

$$= L(f\|f) - L(f\|g), \quad (3)$$

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

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

The KL divergence has the three following properties: it 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 it is always positive as $D_{\mathrm{KL}}(f\|g) \geq 0 \ \forall f, g$. 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:

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

For two GMMs f and g, $D_{\mathrm{KL}}(f\|g)$ is unfortunately intractable. One solution is to use a variational approximation for $D_{\mathrm{KL}}(f\|g)$. Since $D_{\mathrm{KL}}(f\|g) = L(f\|f) - L(f\|g)$, we need only find variational approximations for the expected log likelihood $L(f\|f)$ and $L(f\|g)$.

4. Variational Likelihood

In the case of two GMMs f and g, the expression for $L(f\|g)$ is

$$\begin{aligned} L(f\|g) &\stackrel{\text{def}}{=} \int f(\boldsymbol{x}) \log g(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &= \sum_a \pi_a \int f_a(\boldsymbol{x}) \log \sum_b \pi_b g_b(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \end{aligned} \tag{6}$$

We define the variational parameters $\phi_{b|a}$ as a measure of the affinity between the Gaussian component f_a of f and component g_b of g. The variational parameters satisfy the constraints,

$$\phi_{b|a} \geq 0 \quad \text{and} \quad \sum_b \phi_{b|a} = 1. \tag{7}$$

Using Jensen's inequality, we obtain a lower bound for (6),

$$\begin{aligned} L(f\|g) &= \sum_a \pi_a \int f_a(\boldsymbol{x}) \log \sum_b \phi_{b|a} \frac{\pi_b g_b(\boldsymbol{x})}{\phi_{b|a}} \mathrm{d}\boldsymbol{x} \\ &\geq \sum_a \pi_a \int f_a(\boldsymbol{x}) \sum_b \phi_{b|a} \log \frac{\pi_b g_b(\boldsymbol{x})}{\phi_{b|a}} \mathrm{d}\boldsymbol{x} \\ &= \sum_a \pi_a \sum_b \phi_{b|a} \left(\log \frac{\pi_b}{\phi_{b|a}} + L(f_a\|g_b) \right) \qquad (8) \\ &\stackrel{\text{def}}{=} \mathcal{L}_\phi(f\|g). \qquad (9) \end{aligned}$$

The lower bound on $L(f\|g)$, given by the variational approximation $\mathcal{L}_\phi(f\|g)$ can be maximized with respect to ϕ. The best bound is given by

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

By substituting $\widehat{\phi}_{b|a}$ in (8), we can get the following expression for $\mathcal{L}_{\widehat{\phi}}(f\|g)$,

$$\begin{aligned} \mathcal{L}_{\widehat{\phi}}(f\|g) &= \sum_a \pi_a \sum_b \widehat{\phi}_{b|a} \left(\log \frac{\pi_b}{\widehat{\phi}_{b|a}} + L(f_a\|g_b) \right) \\ &= \sum_a \pi_a \log \left(\sum_b \pi_b e^{L(f_a\|g_b)} \right). \end{aligned} \tag{11}$$

$\mathcal{L}_{\widehat{\phi}}(f\|g)$ is the best variational approximation of the expected log likelihood $L(f\|g)$. It is referred to as *variational likelihood* in the rest of this paper. Similarly, we can find the variational likelihood $\mathcal{L}_{\widehat{\psi}}(f\|f)$, which maximizes a lower bound on $L(f\|f)$,

$$\mathcal{L}_{\widehat{\psi}}(f\|f) = \sum_a \pi_a \log \left(\sum_{a'} \pi_{a'} e^{L(f_a\|f_{a'})} \right). \tag{12}$$

The variational KL divergence $\mathfrak{D}_{\mathrm{KL}}(f\|g)$ is obtained directly from (11) and (12) since $\mathfrak{D}_{\mathrm{KL}}(f\|g) = \mathcal{L}_{\widehat{\psi}}(f\|f) - \mathcal{L}_{\widehat{\phi}}(f\|g)$,

$$\mathfrak{D}_{\mathrm{KL}}(f\|g) = \sum_a \pi_a \log \left(\frac{\sum_{a'} \pi_{a'} e^{L(f_a\|f_{a'})}}{\sum_b \pi_b e^{L(f_a\|g_b)}} \right) \tag{13}$$

$$= \sum_a \pi_a \log \left(\frac{\sum_{a'} \pi_{a'} e^{-D_{\mathrm{KL}}(f_a\|f_{a'})}}{\sum_b \pi_b e^{-D_{\mathrm{KL}}(f_a\|g_b)}} \right). \tag{14}$$

Both (13) and (14) are equivalent, while (14) seems more intuitive as it gives $\mathfrak{D}_{\mathrm{KL}}(f\|g)$ based on the KL divergences between all individual components of f and g.

In the context of refactoring models, we can optimize the parameters of g to better match f by minimizing the KL divergence $D_{\mathrm{KL}}(f\|g)$. Since the variational KL divergence $\mathfrak{D}_{\mathrm{KL}}(f\|g)$ gives an approximation to $D_{\mathrm{KL}}(f\|g)$, we can maximize $\mathfrak{D}_{\mathrm{KL}}(f\|g)$ with respect to $\{\pi_b, \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b\}$, parameters of g. It is clearly sufficient to maximize the variational $\mathcal{L}_\phi(f\|g)$, as $\mathcal{L}_\psi(f\|f)$ is constant in g. Although (11) is not easily maximized with respect to the parameters of g, $\mathcal{L}_\phi(f\|g)$ in (8) leads to an Expectation-Maximization (EM) algorithm.

5. Variational Expectation-Maximization

We need to maximize $\mathcal{L}_\phi(f\|g)$, with respect to ϕ and the parameters $\{\pi_b, \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b\}$ of g. This can be achieved by defining a *variational* Expectation-Maximization (varEM) algorithm. Previously, we found the best lower bound on $L(f\|g)$ with $\mathcal{L}_{\widehat{\phi}}(f\|g)$ by estimating $\widehat{\phi}_{b|a}$. This is the *expectation* (E) step:

$$\widehat{\phi}_{b|a} = \frac{\pi_b e^{-D_{\mathrm{KL}}(f_a\|g_b)}}{\sum_{b'} \pi_{b'} e^{-D_{\mathrm{KL}}(f_a\|g_{b'})}}. \tag{15}$$

For a given $\widehat{\phi}_{b|a}$, it is now possible to find the parameters of g that maximize $\mathcal{L}_{\widehat{\phi}}(f\|g)$. The *maximization* (M) step is:

$$\pi_b = \sum_a \pi_a \widehat{\phi}_{b|a}, \tag{16}$$

$$\boldsymbol{\mu}_b = \frac{\sum_a \pi_a \widehat{\phi}_{b|a} \boldsymbol{\mu}_a}{\sum_a \pi_a \widehat{\phi}_{b|a}}, \tag{17}$$

$$\boldsymbol{\Sigma}_b = \frac{\sum_a \pi_a \widehat{\phi}_{b|a} \left[\boldsymbol{\Sigma}_a + (\boldsymbol{\mu}_a - \boldsymbol{\mu}_b)(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b)^T \right]}{\sum_a \pi_a \widehat{\phi}_{b|a}}. \tag{18}$$

The algorithm alternates between the E–step and M–step, increasing the variational likelihood in each step. We can test for convergence by measuring the increase in variational likelihood during each step, and stopping when it is sufficiently small.

5.1. Discrete Variational EM

If we constrain $\phi_{b|a}$ to take discrete $\{0, 1\}$ values, and maximize the same variational objective function, we obtain an algorithm equivalent to K–means clustering of Gaussians using KL divergence as the distance measure, as in [5]. This provides a hard assignment of the components of f to the components of g. Let $\Phi_{b|a}$ be the constrained $\phi_{b|a}$. In the constrained E–step, for a given a, the optimal solution is to assign it to the b for which $\phi_{b|a}$ is greatest. That is, we find $\hat{b} = \arg\max_b \phi_{b|a}$, and set $\Phi_{\hat{b}|a} = 1$ and $\Phi_{b|a} = 0$ for all $b \neq \hat{b}$. The M–step remains the same, and the resulting g_b is the maximum likelihood Gaussian given the selection of components from f provided by Φ. We call this the *discrete* variational EM (discrete varEM).

A potential caveat of this algorithm is if no cluster is assigned to a component g_b. This can happen, for instance, if there are two components of g that have similar means and variances, but different priors. It results in $\sum_a \Phi_{b|a} = 0$ for the orphaned b. In the M-step, this leads to a zero π_b and infinite components for $\boldsymbol{\mu}_b$ and $\boldsymbol{\Sigma}_b$. One solution to this problem is to *delete* the orphaned component g_b. This ensures that the variational likelihood increases with every step, but it reduces the number of clustered components.

An alternative is to heuristically re-allocate Gaussians from a larger cluster to the orphaned component. In this case, the variational likelihood does not necessarily increase during the reallocation step, but if iterated will continue to increase on subsequent E and M–steps. We chose this approach to keep the number of Gaussians constant when we compare across the different techniques. In the continuous varEM, however, it is possible that two components, g_b and $g_{b'}$, converge to the same mean and variance, which is equivalent to reducing the number of Gaussians in g. This may set varEM at a slight disadvantage relative to discrete varEM.

6. Weighted Local Maximum Likelihood

The variational EM algorithm is sensitive to the choice of initial model g^0. A greedy clustering approach based on local maximum likelihood (LML) was proposed in [2] to provide g^0. A cost is computed for every pair of components in the model f. The pair with lowest cost is merged providing a new model f'. The algorithm iterates until the desired number of components is reached. LML measures the divergence between a pair of Gaussians and their resulting merge. When the Gaussian parameters are constrained (e.g., diagonal covariances), the selected pair is well approximated under the constraints.

The LML cost function is as follows: consider two Gaussians f_i and f_j with weights π_i and π_j, define $p = \pi_i f_i + \pi_j f_j$, and consider $q = \mathrm{merge}(\pi_i f_i, \pi_j f_j)$. q is the Gaussian resulting from merging the components of p. This merge can be performed by using (16)–(18) and the proper Φ. Since KL divergence is defined for distributions, LML defines weights $\tilde{\pi}_i = \pi_i/(\pi_i + \pi_j)$, $\tilde{\pi}_j = 1 - \tilde{\pi}_i$, to form a GMM $\tilde{p} = \tilde{\pi}_i f_i + \tilde{\pi}_j f_j$, and the merged Gaussian, $\tilde{q} = \mathrm{merge}(\tilde{\pi}_i f_i, \tilde{\pi}_j f_j)$. LML is the KL divergence between normalized distributions $\tilde{p}$ and $\tilde{q}$.

$$\mathrm{LML}(\tilde{p}, \tilde{q}) = D_{\mathrm{KL}}(\tilde{p}, \tilde{q}) = \int \tilde{p}(\boldsymbol{x}) \log \frac{\tilde{p}(\boldsymbol{x})}{\tilde{q}(\boldsymbol{x})} \,\mathrm{d}\boldsymbol{x}. \qquad (19)$$

Since $\tilde{p}$ and $\tilde{q}$ are properly normalized, (19) benefits from all the properties of the KL divergence.

We now propose to use a generalized KL-divergence in the Bregman divergence family, as given in [6]

$$\check{D}_{\mathrm{KL}}(p\|q) = \int p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} + \int q(\boldsymbol{x}) - p(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}, \quad (20)$$

where p and q are un–normalized. If we consider $p = \alpha\tilde{p}$ and $q = \beta\tilde{q}$ where $\tilde{p}$ and $\tilde{q}$ are normalized, then (20) becomes

$$\begin{aligned} \check{D}_{\mathrm{KL}}(p\|q) &= \int \alpha\tilde{p} \log \frac{\alpha\tilde{p}}{\beta\tilde{q}} + \int (\beta\tilde{q} - \alpha\tilde{p}) \\ &= \alpha D_{\mathrm{KL}}(\tilde{p}\|\tilde{q}) + \alpha \log \frac{\alpha}{\beta} + \beta - \alpha. \qquad (21) \end{aligned}$$

Since, in our case $p = (\pi_i + \pi_j)\tilde{p}$ and $q = (\pi_i + \pi_j)\tilde{q}$, then $\alpha = \beta = \pi_i + \pi_j$ and (21) becomes

$$\begin{aligned} \check{D}_{\mathrm{KL}}(p\|q) &= (\pi_i + \pi_j) D_{\mathrm{KL}}(\tilde{p}\|\tilde{q}) \qquad (22) \\ &= (\pi_i + \pi_j) \mathrm{LML}(\tilde{p}, \tilde{q}). \end{aligned}$$

WER (%) vs. Model Size (K)										
Models	5	10	15	20	25	30	35	40	45	50
Baseline	2.49	2.00	1.68	1.49	1.37	1.38	1.39	1.33	1.27	1.31
100K–STC	2.53	1.89	1.70	1.51	1.39	1.35	1.34	1.30	1.32	1.28
Models	55	60	65	70	75	80	85	90	95	100
Baseline	1.29	1.27	1.27	1.29	1.22	1.21	1.30	1.20	1.21	1.18
100K-STC	1.28	1.28	1.25	1.23	1.16	1.23	1.20	1.23	1.22	1.18

Table 1: WERs for baseline and 100K-STC models.

Using un–normalized distributions for LML leads to a weighted LML of the normalized distributions. Since $\pi_i + \pi_j > 0$, all the properties of the KL divergence hold for $\check{D}_{\mathrm{KL}}(p\|q)$. We use the *weighted* Local Maximum Likelihood (weighted LML or wLML) as cost function with the greedy clustering described previously to provide g^0 for the variational EM.

7. Experiments

The experimental setup is very close to the one described in [2]. The same internal IBM databases were used for all our experiments. The training set is composed of 786 hours of US English data, with 10.3K speakers for a total of 803K utterances. It consists of in-car speech in various noise conditions, recorded at 0, 30 and 60 mph with 16KHz sampling frequency. The test set is 38.9K sentences for a total of 206K words. It is a set of 47 different tasks of in-car speech with various US regional accents.

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) combined with Semi Tied Covariance (STC) transformation. 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 20 baseline models from training data from 5K, 10K, ..., 100K Gaussians (our reference model). All these models have different STCs and lie in different feature spaces. Since all clustered models are in the reference model feature space, for consistency we built 20 models using the 100K model's STC (100K–STC). Differences in the WERs for these models and the baseline are small, as shown in Table 1.

The results presented in this paper differ significantly from those reported in [2], as Table 1 reveals. Indeed, the setup in [2] was our internal product setup where, prior to decoding, acoustic models are compressed using Band Quantization (BQ), refactored for speed by integrating a hierarchy [5], and where likelihood computation robustness is ensured [7]. These product enhancements introduce dynamics in decoding that blur the exact impact of refactoring the acoustic models. We provide here results on a research setup with likelihood computation without BQ, hierarchy nor likelihood robustness activated. They are significantly better than for the product setup.

Baseline results show that the reference WER for 100K model is 1.18%. WERs remain within 15% relative from 95K down to 40K, then start to increase significantly below 25K. At 5K, WER increased 110% relative to 100K. We used our greedy clustering algorithm to cluster the reference 100K model down to 5K, saving intermediate models every 5K Gaussians, for a total of 19 clustered models. Sets of 19 models were created

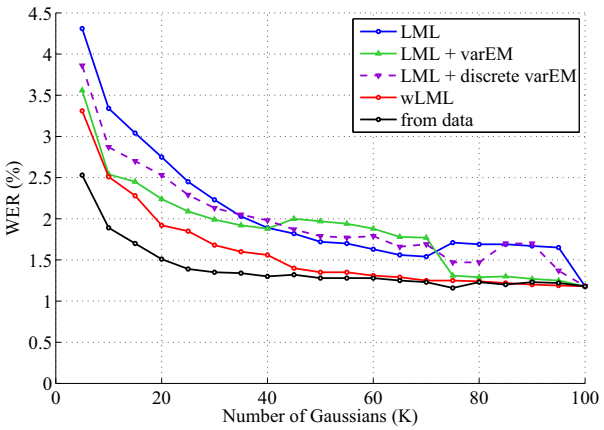

Figure 1: WER as a function of the number of Gaussians for models trained from data (100K–STC), models clustered using LML, LML with varEM and discrete varEM, and wLML.

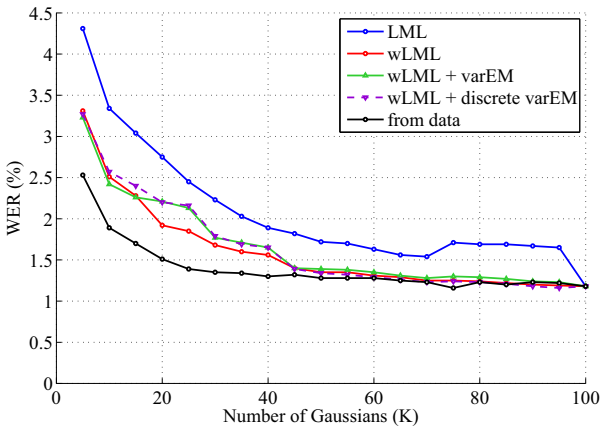

Figure 2: WER as a function of the number of Gaussians for models trained from data (100K–STC), models clustered using LML, wLML, wLML with varEM and discrete varEM.

using LML proposed in [2], and the newly proposed weighted LML cost function. Results for these clustered models are plotted in Figure 1 for LML and Figure 2 for weighted LML. The proposed weighted LML *significantly* outperforms LML for all model sizes as we can see in Figure 1. In average, a 24% relative improvement over the LML results is observed. Weighted LML even improves on the models trained from data by 2.5% relative at 90K and 95K. From 45K–95K, weighted LML is within 8% of the trained models and at 45K, it shows only a 6% relative degradation compared to the 45K trained model. At 5K, a model trained from data gives 2.53% WER, weighted LML 3.31% and LML 4.31%. That is a 23% improvement from the LML result.

Results for both varEM optimization techniques (varEM and discrete varEM) are also plotted for LML and weighted LML in Figure 1 and Figure 2 respectively. Initial models were clustered using LML and weighted LML, then several iterations of varEM were performed to change model parameters as to better match the 100K reference model. For LML, from 75K–95K, both varEM techniques improve on the performance of the clustered models. In this range, varEM gives a consistent 30% relative gain over LML results, much better than discrete varEM. From 40K–80K, both techniques cannot improve on the LML models, with discrete varEM keeping within 9% of the LML performance, closer than varEM. Over the 5K–35K range, a clear trend of improvement is observed for both techniques, reaching its peak at 10K. Indeed, at 10K varEM gives a 2.54% WER which is a 31% relative gain over LML's 3.34% WER, almost reaching the performance of weighted LML with 2.51% WER. We have observed that, not surprisingly, varEM needs a lot more iterations to converge than discrete varEM when updating the same CD state using identical initialization. Since discrete varEM always gives performance closer to its initial model, we can only conjecture that the constrained nature of its E–step (discrete ϕ) may be helping converge to local maxima closer to the initial model, rather than for the relatively unconstrained varEM.

For weighted LML, both varEM techniques remain close to the initial model results from 45K–95K. From 20K–40K, both varEM techniques seem to slightly diverge from the weighted LML results only to converge again in the 5K–15K range where varEM gives the best results. At 5K, varEM gives a WER of 3.23%, which is a 2.48% relative improvement over weighted LML with 3.31% WER. Overall, weighted LML is a solid and significant improvement over LML.

8. Conclusion

We have introduced a new greedy clustering algorithm based on weighted LML, which improves upon the previously published methods. We demonstrated the validity of the varEM and discrete varEM optimization methods on a speech recognition task. There is still a gap in performance between refactored and trained models for large size reductions. To reduce this gap may require stronger methods that globally optimize the number of components per GMM. However, we show that weighted LML can reduce model size by 50%, with almost the same recognition performance as the corresponding model trained from data. Weighted LML is an order of magnitude faster than training from data (the run time is hours instead of days), making it a viable alternative for refactoring models.

9. References

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

[2] P. L. Dognin, J. R. Hershey, V. Goel, and P. A. Olsen, "Refactoring acoustic models using variational density approximation," in *ICASSP*, April 2009, pp. 4473–4476.

[3] K. Zhang and J. T. Kwok, "Simplifying mixture models through function approximation," in *NIPS 19*. MIT Press, 2007, pp. 1577–1584.

[4] X.-B. Li, F. K. Soong, T. A. Myrvoll, and R.-H. Wang, "Optimal clustering and non-uniform allocation of gaussian kernels in scalar dimension for hmm compression," in *ICASSP*, March 2005, pp. 669–672.

[5] R. Bakis, D. Nahamoo, M. A. Picheny, and J. Sedivy, "Hierarchical labeler in a speech recognition system," U.S. Patent 6023673.

[6] I. Csiszár, "Why least squares and maximum entropy? an axiomatic approach to inference for linear inverse problems," *Annals of Statistics*, vol. 19, no. 4, pp. 2032–2066, 1991.

[7] L. R. Bahl, P. V. de Souza, P. S. Gopalakrishnan, D. Nahamoo, and M. A. Picheny, "Robust methods for using context-dependent features and speech recognition models in a continuous speech recognizer," in *ICASSP*, April 1994, pp. 533–536.