RESTRUCTURING ACOUSTIC MODELS FOR CLIENT AND SERVER BASED AUTOMATIC SPEECH RECOGNITION

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

IBM T.J. Watson Research Center
Yorktown Heights, NY 10598, USA
{pdognin, jrhershe, vgoel, pederao}@us.ibm.com

ABSTRACT

A problem often encountered in probabilistic modeling is restructuring a model to change its number of components, parameter sharing, or some other structural constraints. Automatic Speech Recognition (ASR) has become ubiquitous and building acoustic models (AMs) that can retain good performance while adjusting their size to application requirements is a challenging problem. AMs are usually built around Gaussian mixture models (GMMs) that can be each restructured, impacting directly the properties of the overall AM. For instance, generating smaller AMs from a reference AM can simply be done by restructuring the underlying GMMs to have fewer components while best approximating the original GMMs. Maximizing the likelihood of a restructured model under the reference model is equivalent to minimizing their Kullback-Leibler (KL) divergence. For GMMs, this is analytically intractable. However, a lower bound to the likelihood can be maximized and a variational expectation-maximization (EM) can be derived. Using variational KL divergence and variational EM in the task of AM clustering, we define a greedy clustering algorithm that can build on demand clustered models of any size from a reference model. Our latest results show that clustered models are on average within 2.7% of the WERs for equivalent models built from data, and only at 9% for a model 20 times smaller than our reference model. This makes clustering a reference model a viable option to training models from data.

Index Terms— acoustic model clustering, variational approximation, variational expectation-maximization

1. INTRODUCTION

A problem commonly encountered in probabilistic modeling is to approximate one model using another model with a different structure. By changing the number of components or parameters, by sharing parameters differently, or simply by modifying some other constraints, a model can be *restructured* to better suit the requirements of an application. Model restructuring is particularly relevant to the field of Automatic Speech Recognition (ASR) since ASR is now present on a wide variety of platforms, ranging from huge server farms to small embedded devices. Each platform has its own resource limitations and this variety of requirements makes building acoustic models (AMs) for all these platforms a delicate task of balancing compromises between model size, decoding speed, and accuracy. We take the view that model restructuring can alleviate issues with building AMs for various platforms.

Clustering, parameter sharing, and hierarchy building are three different examples of AM restructuring commonly used in ASR. For instance, clustering an AM to reduce its size, despite a moderate degradation in recognition performance, has an impact on both ends of the platform spectrum. For servers, it means more models can fit into memory and more engines can run simultaneously sharing these models. For mobile devices, models can be custom-built to exactly fit into memory. Sharing parameters across components is another restructuring technique that efficiently provides smaller models with adjustable performance degradation. Restructuring an AM by using a hierarchy speeds up the search for the most likely components and directly impacts recognition speed and accuracy. This paper examines the problem of clustering models efficiently.

When restructuring models, an important issue that arises is measuring similarity between *reference* and *restructured* model. Minimizing the Kullback-Leibler (KL) divergence [1] between these two models, is equivalent to maximizing the likelihood of the restructured model under data drawn from the reference model. Unfortunately, this is intractable for Gaussian Mixture Models (GMMs), a core component of AMs, without resorting to expensive Monte Carlo techniques. However, it is possible to maximize a variational lower bound to the likelihood and derive from it a variational KL divergence [2] as well as an Expectation-Maximization (EM) algorithm [3] that will update the parameters of a model to better match a reference model. Both variational KL divergence and variational EM can be key components of model restructuring. This paper shows a greedy clustering algorithm based on these methods that provides clustered and refined models of any size, on demand. For other approaches, based on minimizing the mean–squared error between the two density functions, see [4], or based on compression using dimension–wise tied Gaussians optimized using symmetric KL divergences, see [5].

This paper builds on previous work and presents results showing that, starting with a large acoustic model, we can now derive smaller models that achieve word error rates (WERs) almost identical to those for similar size models trained from data. This eliminates the need to train models of every desired sizes.

2. MODELS

Acoustic models are typically structured around phonetic states and take advantage of phonetic context while modeling observation. Let $\mathcal{A}$ be an acoustic model composed of L context dependent (CD) states. L is chosen at training time and typically ranges from a few hundreds to a few thousands. Each CD state s uses a GMM f_s with N_s components resulting in $\mathcal{A}$ having $N = \sum_s N_s$ Gaussians. A GMM f with continuous observation $\boldsymbol{x} \in \mathbb{R}^d$ is specified as

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

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 $\boldsymbol{\Sigma}_a$ which is symmetric and positive definite. In general, $\boldsymbol{\Sigma}_a$ is a full matrix but in practice it is typically chosen to be diagonal for computation and storage efficiency.

3. VARIATIONAL KL DIVERGENCE

The KL divergence [1] is a commonly used measure of dissimilarity between 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,

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

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

$$L\big(f\|g\big) = \sum_a \pi_a \int f_a(\boldsymbol{x}) \log \sum_b \omega_b g_b(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{5}$$

where the integral $\int f_a \log \sum_b \omega_b g_b$ is analytically intractable. As a consequence, $D_{\mathrm{KL}}(f\|g)$ is intractable for GMMs.

One solution presented in [3] provides a variational approximation to $D_{\mathrm{KL}}(f\|g)$. This is done by first providing variational approximations to $L(f\|f)$ and $L(f\|g)$ and then using (3). In order to define a variational approximation to (5), variational parameters $\phi_{b|a}$ are introduced as a measure of the affinity between the Gaussian component f_a of f and component g_b of g. The variational parameters must satisfy the constraints

$$\phi_{b|a} \geq 0 \quad \text{and} \quad \sum_b \phi_{b|a} = 1. \tag{6}$$

By using Jensen's inequality, a lower bound is obtained for (5),

$$L\big(f\|g\big) = \sum_a \pi_a \int f_a(\boldsymbol{x}) \log \sum_b \phi_{b|a} \frac{\omega_b g_b(\boldsymbol{x})}{\phi_{b|a}} \mathrm{d}\boldsymbol{x}$$

$$\geq \sum_a \pi_a \int f_a(\boldsymbol{x}) \sum_b \phi_{b|a} \log \frac{\omega_b g_b(\boldsymbol{x})}{\phi_{b|a}} \mathrm{d}\boldsymbol{x}$$

$$= \sum_a \pi_a \sum_b \phi_{b|a} \left(\log \frac{\omega_b}{\phi_{b|a}} + L(f_a\|g_b) \right) \tag{7}$$

$$\stackrel{\mathrm{def}}{=} \mathfrak{L}_\phi\big(f\|g\big). \tag{8}$$

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

$$\widehat{\phi}_{b|a} = \frac{\omega_b e^{L(f_a\|g_b)}}{\sum_{b'} \omega_{b'} e^{L(f_a\|g_{b'})}} \tag{9}$$

By substituting $\widehat{\phi}_{b|a}$ in (7), the following expression for $\mathfrak{L}_{\widehat{\phi}}\big(f\|g\big)$ is obtained:

$$\mathfrak{L}_{\widehat{\phi}}(f\|g) = \sum_a \pi_a \sum_b \widehat{\phi}_{b|a} \left(\log \frac{\omega_b}{\widehat{\phi}_{b|a}} + L(f_a\|g_b) \right)$$

$$= \sum_a \pi_a \log \left(\sum_b \omega_b e^{L(f_a\|g_b)} \right). \tag{10}$$

$\mathfrak{L}_{\widehat{\phi}}\big(f\|g\big)$ is the best variational approximation of the expected log likelihood $L\big(f\|g\big)$ and is referred to as *variational likelihood*. Similarly, the variational likelihood $\mathfrak{L}_{\widehat{\psi}}\big(f\|f\big)$, which maximizes a lower bound on $L\big(f\|f\big)$, is

$$\mathfrak{L}_{\widehat{\psi}}\big(f\|f\big) = \sum_a \pi_a \log \left(\sum_{a'} \pi_{a'} e^{L(f_a\|f_{a'})} \right). \tag{11}$$

The variational KL divergence $\mathfrak{D}_{\mathrm{KL}}(f\|g)$ is obtained directly from (10) and (11) since $\mathfrak{D}_{\mathrm{KL}}(f\|g) = \mathfrak{L}_{\widehat{\psi}}\big(f\|f\big) - \mathfrak{L}_{\widehat{\phi}}\big(f\|g\big)$,

$$\mathfrak{D}_{\mathrm{KL}}(f\|g) = \sum_a \pi_a \log \left(\frac{\sum_{a'} \pi_{a'} e^{-D_{\mathrm{KL}}(f_a\|f_{a'})}}{\sum_b \omega_b e^{-D_{\mathrm{KL}}(f_a\|g_b)}} \right), \tag{12}$$

where $\mathfrak{D}_{\mathrm{KL}}(f\|g)$ is based on the KL divergences between all individual components of f and g.

4. GENERALIZED VARIATIONAL KL DIVERGENCE

We propose to extend the variational KL to weighted densities $\overline{f}$ and $\overline{g}$ where $\alpha = \int \overline{f}$ and $\beta = \int \overline{g}$ so that $f = \overline{f}/\alpha$ and $g = \overline{g}/\beta$ are the corresponding pdfs. The generalized KL divergence in the Bregman divergence family, as given in [6], is

$$\overline{D}_{\mathrm{KL}}(\overline{f}\|\overline{g}) = \int \overline{f}(\boldsymbol{x}) \log \frac{\overline{f}(\boldsymbol{x})}{\overline{g}(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} + \int \overline{g}(\boldsymbol{x}) - \overline{f}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}. \tag{13}$$

$$= \alpha D_{\mathrm{KL}}(f\|g) + \alpha \log \frac{\alpha}{\beta} + \beta - \alpha. \tag{14}$$

For pdfs f and g, $\int f = \int g = 1$ and (14) yields $D_{\mathrm{KL}}(f\|g)$ as expected. If $\overline{f}$ and $\overline{g}$ are weighted GMMs, it is straightforward to define a *generalized* variational KL from (12) and (14),

$$\overline{\mathfrak{D}}_{\mathrm{KL}}(\overline{f}\|\overline{g}) \stackrel{\mathrm{def}}{=} \alpha \mathfrak{D}_{\mathrm{KL}}(f\|g) + \alpha \log \frac{\alpha}{\beta} + \beta - \alpha, \tag{15}$$

where $\overline{\mathfrak{D}}_{\mathrm{KL}}(\overline{f}\|\overline{g})$ is based on the variational KL divergence between GMMs f and g.

4.1. Weighted Local Maximum Likelihood

Let us consider a weighted GMM $\overline{f}_{\mathcal{Q}}$ where $\overline{f}_{\mathcal{Q}} = \sum_{i\in\mathcal{Q}} \pi_i f_i$ with $\mathcal{Q} = \{i, j, ..., k\}$ being a subset of indices of the components in GMM f. Its normalized counterpart is $f_{\mathcal{Q}} = \sum_{i\in\mathcal{Q}} \hat{\pi}_i f_i$ where $\hat{\pi}_i = \pi_i / \sum_{i'\in\mathcal{Q}} \pi_{i'}$ is a normalized prior, and we have $\overline{f}_{\mathcal{Q}} = (\sum_{i\in\mathcal{Q}} \pi_i) f_{\mathcal{Q}}$. We define $\overline{g} = \mathrm{merge}(\overline{f}_{\mathcal{Q}})$, where $\overline{g}$ is the weighted Gaussian resulting from merging all the components of $\overline{f}_{\mathcal{Q}}$. The maximum likelihood parameters $\{\pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ associated with $\overline{g}$ are defined as

$$\pi = \sum_{i\in\mathcal{Q}} \pi_i, \tag{16}$$

$$\boldsymbol{\mu} = \frac{\sum_{i\in\mathcal{Q}} \pi_i \boldsymbol{\mu}_i}{\sum_{i\in\mathcal{Q}} \pi_i}, \tag{17}$$

$$\boldsymbol{\Sigma} = \frac{\sum_{i\in\mathcal{Q}} \pi_i \left[\boldsymbol{\Sigma}_i + (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T \right]}{\sum_{i\in\mathcal{Q}} \pi_i}. \tag{18}$$

An especially useful case of the generalized variational KL divergence is $\overline{\mathfrak{D}}_{\mathrm{KL}}(\overline{f}_{\mathcal{Q}}\|\overline{g})$. We define $\alpha = \int \overline{f}_{\mathcal{Q}}$ and $\beta = \int \overline{g}$. Clearly,

we have $\alpha = \beta = \sum_{i\in\mathcal{Q}} \pi_i$ and $\overline{\mathfrak{D}}_{\mathrm{KL}}(\overline{f}_{\mathcal{Q}}\|\overline{g})$ becomes

$$\overline{\mathfrak{D}}_{\mathrm{KL}}(\overline{f}_{\mathcal{Q}}\|\overline{g}) = \alpha\mathfrak{D}_{\mathrm{KL}}(f_{\mathcal{Q}}\|g) + \alpha\log\frac{\alpha}{\beta} + \beta - \alpha$$

$$= \big(\sum_{i\in\mathcal{Q}}\pi_i\big)\mathfrak{D}_{\mathrm{KL}}(f_{\mathcal{Q}}\|g). \tag{19}$$

$$= \big(\sum_{i\in\mathcal{Q}}\pi_i\big)\left[\mathfrak{L}\big(f_{\mathcal{Q}}\|f_{\mathcal{Q}}\big) - \mathfrak{L}\big(f_{\mathcal{Q}}\|g\big)\right], \tag{20}$$

where $g = \overline{g}/\beta$. When $\overline{g}$ is the merged Gaussian of all components of $\overline{f}_{\mathcal{Q}}$, the generalized variational KL becomes a weighted version of the variational KL divergence for $f_{\mathcal{Q}}$ and g properly normalized. Let us define a weighted GMM $\overline{p}$ from two weighted components $\overline{f}_i$ and $\overline{f}_j$ of GMM f such that $\overline{p} = \pi_i f_i + \pi_j f_j$, and where $\overline{q}$ is a weighted Gaussian such that $\overline{q} = \mathrm{merge}(\pi_i f_i, \pi_j f_j)$, we obtain from (19) that

$$\overline{\mathfrak{D}}_{\mathrm{KL}}(\overline{p}\|\overline{q}) = (\pi_i + \pi_j)\mathfrak{D}_{\mathrm{KL}}(p\|q). \tag{21}$$

In this special case, it is clear from (20) that $\overline{\mathfrak{D}}_{\mathrm{KL}}(\overline{p}\|\overline{q})$ measures the difference in variational likelihoods between the GMM $\overline{p}$ and the merged Gaussian $\overline{q}$, effectively providing the divergence between a pair of Gaussians and their resulting merge. For selecting similar pairs of Gaussians, when the Gaussians are constrained to have diagonal covariances, the pair f_i and f_j is well approximated by $\overline{q}$ under this constraint, i.e. the covariance of $\overline{q}$ will be closer to diagonal than for any other pairs. This is the cost function used in our clustering and it is called weighted Local Maximum Likelihood (wLML).

5. VARIATIONAL EXPECTATION-MAXIMIZATION

In the context of restructuring models, the variational KL divergence $\mathfrak{D}_{\mathrm{KL}}(f\|g)$ can be minimized by updating the parameters of the restructured model g to match the reference model f. Since the variational KL divergence $\mathfrak{D}_{\mathrm{KL}}(f\|g)$ gives an approximation to $D_{\mathrm{KL}}(f\|g)$, we can minimize $\mathfrak{D}_{\mathrm{KL}}(f\|g)$ w.r.t. the parameters of g, $\{\pi_b, \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b\}$. It is sufficient to maximize $\mathfrak{L}_\phi(f\|g)$, as $\mathfrak{L}_\psi(f\|f)$ is constant in g. Although (10) is not easily maximized w.r.t. the parameters of g, $\mathfrak{L}_\phi(f\|g)$ in (7) can be maximized leading to an Expectation-Maximization (EM) algorithm.

We need to maximize $\mathfrak{L}_\phi(f\|g)$, w.r.t 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 where we first maximize $\mathfrak{L}_\phi(f\|g)$ w.r.t ϕ. With ϕ fixed, we then maximize $\mathfrak{L}_\phi(f\|g)$ w.r.t the parameters of g. Previously, we found the best lower bound on $L(f\|g)$ with $\mathfrak{L}_{\widehat{\phi}}\big(f\|g\big)$ is given by $\widehat{\phi}_{b|a}$ in (9). This is the *expectation* (E) step:

$$\widehat{\phi}_{b|a} = \frac{\omega_b e^{-D_{\mathrm{KL}}(f_a\|g_b)}}{\sum_{b'}\pi_{b'}e^{-D_{\mathrm{KL}}(f_a\|g_{b'})}}. \tag{22}$$

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

$$\pi_b^\star = \sum_a \pi_a\phi_{b|a}, \tag{23}$$

$$\boldsymbol{\mu}_b^\star = \frac{\sum_a \pi_a\phi_{b|a}\boldsymbol{\mu}_a}{\sum_a \pi_a\phi_{b|a}}, \tag{24}$$

$$\boldsymbol{\Sigma}_b^\star = \frac{\sum_a \pi_a\phi_{b|a}\left[\boldsymbol{\Sigma}_a + (\boldsymbol{\mu}_a - \boldsymbol{\mu}_b^\star)(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b^\star)^T\right]}{\sum_a \pi_a\phi_{b|a}}. \tag{25}$$

The algorithm alternates between the E–step and M–step, increasing the variational likelihood in each step.

5.1. Discrete Variational EM

If we constrain $\phi_{b|a}$ to $\{0, 1\}$, 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, a given a is assigned 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 $b \neq \hat{b}$. In the rare case where several components are assigned the same value $\max_b \phi_{b|a}$, we choose the smallest index of all these components as our $\hat{b}$. The M–step remains the same, and the resulting g_b is the maximum likelihood Gaussian given the subset $\mathcal{Q}$ of components indices from f provided by Φ; the equations (23)–(25) are then similar to the merge steps in (16)–(18) with Φ providing the subset $\mathcal{Q}$. This is called the *discrete* variational EM (discrete varEM).

6. MODEL CLUSTERING

Clustering down an acoustic model $\mathcal{A}$ of size N into a model $\mathcal{A}^{\mathrm{c}}$ of target size N^{c} means reducing the overall number of Gaussian components. In the reference model $\mathcal{A}$, each state s uses a GMM f_s with N_s components to model observation. We create a new model $\mathcal{A}^{\mathrm{c}}$ by clustering down each f_s independently into f_s^{c} of size N_s^{c} such that $1 \leq N_s^{\mathrm{c}} \leq N_s$. The final clustered model $\mathcal{A}^{\mathrm{c}}$ has N^{c} Gaussian components with $N^{\mathrm{c}} = \sum_s N_s^{\mathrm{c}} \leq N$.

A greedy approach is taken to produce f_s^{c} from f_s which finds the best sequence of merges to perform within each f_s so that $\mathcal{A}^{\mathrm{c}}$ reaches the target size N^{c}. This procedure can be divided into two independent parts: 1) cluster down f_s optimally to *any* size. 2) define a criterion to decide the optimal target size N_s^{c} for each f_s under the constraint $N^{\mathrm{c}} = \sum_s N_s^{\mathrm{c}}$. Once the best sequence of merges and N_s^{c} are known, it is straightforward to produce f_s^{c} by using equations (16)–(18).

6.1. Greedy Clustering and Merge-Sequence Building

The proposed greedy algorithm clusters down f_s by sequentially merging component pairs that are similar. For a given cost function, this sequence of merges is deterministic and unique. It is therefore possible to cluster f_s all the way to one final component, while recording each merge and its cost into a *merge–sequence* $\mathcal{S}(f_s)$.

Algorithm 1 shows how to build $\mathcal{S}(f_s)$ for any f_s. At each step, the pair of component (f_i, f_j) in f_s that gives the smallest cost is merged. This results in a new GMM f_s' which is used as f_s for the next iteration. The algorithm iterates until only one component is left in f_s while recording each step in $\mathcal{S}(f_s)$. Algorithm 1 is clearly independent of the cost function C_{f_s}. For our proposed greedy clustering (GC), wLML is our cost function since it measures the likelihood loss when merging a component pair, as discussed earlier.

Given f_s and $\mathcal{S}(f_s)$, it is possible to generate clustered models $f_s^{\mathrm{c}}\{k_s\}$ of any size simply by applying the sequence of merges recorded in $\mathcal{S}(f_s)$ to f_s all the way to the k_s-th merge step. These clustered models span from the original $f_s^{\mathrm{c}}\{0\} = f_s$ to a final $f_s^{\mathrm{c}}\{N_s{-}1\}$. At each step k_s, every new model $f_s^{\mathrm{c}}\{k_s\}$ has one component less than $f_s^{\mathrm{c}}\{k_s{-}1\}$. Therefore, f_s^{c} of *any* target size N_s^{c} can be generated from f_s and $\mathcal{S}(f_s)$. To generate f_s^{c} from f_s, there exists another equivalent option to applying sequentially the best merges in $\mathcal{S}(f_s)$. At each merge step k_s, $\mathcal{S}(f_s)$ can be analyzed to provide $\mathcal{Q}_{k_s}^b$, the set of components indices from f_s whose sequential merges generated component $f_s^{\mathrm{c}}(b)$ of f_s^{c}. With $\mathcal{Q}_{k_s}^b$ known, each $f_s^{\mathrm{c}}(b)$ can be computed in one step using (16)–(18). Since $\mathcal{Q}_{k_s}^b$ refers to indices of the original components of f_s, it means that $\mathcal{Q}_{k_s}^b$ contains

Input: GMM f_s
Output: Merge-Sequence $\mathcal{S}(f_s)$
foreach *Gaussian pair* $(i,j) \in f_s$ **do**
  merging cost: $C_{f_s}(i,j)$
end
merge sequence index: $k_s = 1$
while $(N_s = |f_s|)$ *is greater than 1* **do**
  Find pair with smallest cost $C_{f_s}(i,j)$:
  $(i',j') = \arg\min_{i,j} C_{f_s}(i,j)$
  Merge: $f'_s = \text{merge}(f_s, i', j')$
  Record:
  $\mathcal{S}(f_s)[k_s].\text{pair} = (i',j')$,
  $\mathcal{S}(f_s)[k_s].\text{cost} = C_{f_s}(i',j')$
  Update:
  $f_s = f'_s$
  $k_s = k_s + 1$
  C_{f_s}
end

Algorithm 1: Building Merge-Sequence $\mathcal{S}(f_s)$ for a GMM f_s.

the *ancestry* of $f_s(b)$ at each step k_s.

A useful property of our GC algorithm is that if a weight λ_s is applied to f_s, the sequence of merges in $\mathcal{S}(f_s)$ will remain unchanged, only their corresponding costs will be modified. Indeed, if we apply λ_s to f_s so that $\overline{f}_s = \lambda_s f_s$, each component of f_s is weighted by λ_s too. In this special case, computing wLML from (21), we obtain

$$\overline{\mathfrak{D}}_{\text{KL}}(\lambda_s \overline{p} \| \lambda_s \overline{q}) = \lambda_s(\pi_i + \pi_j)\mathfrak{D}_{\text{KL}}(p\|q). \tag{26}$$

Therefore, applying a weight λ_s to each state s just impacts the wLML costs in $\mathcal{S}(f_s)$.

If $\mathcal{A}$ is composed of L contextual states and N Gaussians, once $\mathcal{S}(f_s)$ are built, models $\mathcal{A}^{\text{c}}$ of any size N^{c} can be created, such that $L \leq N^{\text{c}} \leq N$. The minimum size for $\mathcal{A}^{\text{c}}$ is L because one Gaussian is the smallest any f_s^{c} can be. Since $\mathcal{S}(f_s)$ are independently computed across all states s, the only difference between two AMs with identical size N^{c} is their respective number of components N_s^{c}. Finding good Gaussian assignments N_s^{c} is crucial for clustering models.

6.2. Gaussian Assignment

For any given target size N^{c}, many $\mathcal{A}^{\text{c}}$ can be produced, each of them with different N_s^{c} for state s. N_s^{c} are chosen so that $N^{\text{c}} = \sum_s N_s^{\text{c}}$. However, choosing N_s^{c} means putting all states into competition to find an optimal sharing of N^{c} components. This raises several issues. First, the selection of N_s^{c} should take into account the distortion brought into the model $\mathcal{A}$ when creating $\mathcal{A}^{\text{c}}$. Second, one may want to use some prior information about each state s when determining N_s^{c}. The first issue can be addressed by using the cost information recorded in $\mathcal{S}(f_s)$ when selecting N_s^{c}. These costs are directly related to the distortion brought into the model by each merge. The second issue can be addressed by applying a weight λ_s to all the costs recorded in $\mathcal{S}(f_s)$ so to amplify or decrease them compared to cost for other states. If choosing Nc_s is based on wLML costs across states, this will result in assigning more (or less) Gaussians to this state. λ_s can be chosen in many ways. For instance, during training time, each state is associated with some frames from the training data. The number of frames associated to state s divided by the total number of frames in the training data is the prior for state s and this prior can be used as λ_s. In practice, it is common that states modeling silence observe a large amount of the training data and therefore should be assigned greater N_s^{c} than for other states. λ_s could also be based on the language models or grammars used for decoding since they also reveal how often a state s is expected to be observed. Three assignment strategies are presented that address either one or both of the two issues raised in this section.

6.2.1. α-Assignment

Appropriately assigning a number of components to each f_s is a problem first encountered during AM training. A common approach is to link N_s to the number of frames in the training data modeled by state s. Let us now define c_s as the count of frames in the training data that align to a particular state s. N_s is commonly [7] defined by using

$$N_s(\alpha,\beta) = \lfloor \alpha c_s^{\beta} \rfloor \tag{27}$$

$$N(\alpha,\beta) = \sum_s \lfloor \alpha c_s^{\beta} \rfloor = N, \tag{28}$$

where $\beta = 0.2$. Since training data can be largely composed of silence, β is empirically chosen to ensure that states modeling silence do not grab most of the N Gaussians, at the expense of states modeling actual speech. With β fixed, finding α is done with an iterative process. For clustering, c_s can be obtained for $\mathcal{A}$ and this method provides N_s^{c} given N^{c}. It is referred to as α-*assignment*, which is intrinsically sensitive to the state priors through c_s. However, it does not account for distortion when producing $\mathcal{A}^{\text{c}}$. Combining our greedy clustering with α-assignment is referred to as GC-α in the rest of this paper.

6.2.2. Global Greedy Assignment

The *global greedy assignment* (GGA) extends the greedy approach used in building $\mathcal{S}(f_s)$ to find the best sequence of merges *across* all states. $\mathcal{S}(f_s)$ records the sequence of merges within f_s, using wLML cost function. GGA begins by setting merge sequence index $k_s = 1$ for all states s. Then, GGA finds the best merge across all states by comparing costs recorded in each $\mathcal{S}(f_s)[k_s]$. For s', state of the next best merge, the merge sequence index k_s is increased, $k_{s'} = k_{s'} + 1$, to point to the next best merge within $f_{s'}$. For each state s, GGA keeps track of which merge sequence index k_s it must use to access the next best merge recorded in $\mathcal{S}(f_s)[k_s]$. This simple algorithm iterates until the target N^{c} is reached, ultimately providing N_s^{c} as the number of Gaussians left in each f_s^{c}. If a state s has only one component left before N^{c} is reached, it will be assigned only one Gaussian.

GGA is fast and requires very simple book-keeping. Indeed, no real merge occurs. At each iteration, GGA only needs to follow the sequence within each $\mathcal{S}(f_s)$. GGA follows a merge sequence that tries, to some extent, to minimize merging distortion to the original model $\mathcal{A}$. For GGA, each state s is equally likely to host the next best merge. However, it is straightforward to allow for different state prior λ_s by modifying the costs in $\mathcal{S}(f_s)[k_s]$ accordingly, as discussed earlier. Changing the state priors will change the sequence of best merges across states (but not within states). Combining GC with GGA is referred to as GC-GGA in this paper.

6.2.3. Viterbi Selection

Finding an optimal set of N_s^{c} can be done by using a Viterbi procedure inspired from a different optimal allocation problem in [8].

Within each state s in $\mathcal{A}$, f_s is of size N_s. The size of f_s^{c} at each merge step k_s in $\mathcal{S}(f_s)[k_s]$ is $\mathcal{N}_s - k_s$ for $1 \leq k_s \leq N_s - 1$. The following Viterbi procedure finds the optimal merge step k_s^* so that $N^{\mathrm{c}} = \sum_s N_s - k_s^*$. For each state s, the cumulative wLML cost is required at each step k_s such that $E(s, k_s) = \sum_{i=1}^{k_s} \mathcal{S}(f_s)[i].\mathrm{cost}$. Each cost can be adjusted for weight λ_s if required. The procedure is

1. Initialize $V(1, r) = E(1, r)$
2. For $s = 2, \ldots L$ apply the recursive relation

$$V(s, r) = \min_{r_1 + r_2 = r} (E(s, r_1) + V(s-1, r_2))$$

3. Once $s = L$ is reached, backtrack to find the best k_s^* which gives the best assignment $N_s - k_s^*$.

This procedure gives the Gaussian assignments that minimize overall cost. It therefore optimizes both Gaussian assignment and attempts to minimize model distortion simultaneously. Again, λ_s can be taken into account simply by modifying the cumulative costs before running the Viterbi procedure. Combining our greedy clustering and Viterbi selection is referred to as GC-viterbi in this paper.

7. MODEL REFINEMENT

Once $\mathcal{A}$ is clustered down into $\mathcal{A}^{\mathrm{c}}$, it is possible to refine the parameters of $\mathcal{A}^{\mathrm{c}}$ with varEM by using $\mathcal{A}$ as the model to match. Parameters of each f_s^{c} will be updated to minimize $\mathfrak{D}_{\mathrm{KL}}(f_s \| f_s^{\mathrm{c}})$. For each state s, f_s is used as reference model and f_s^{c} is used as initial model for varEM. At the end of convergence, we obtain a new model $\mathcal{A}^{\mathrm{r}}$ composed by the set of f_s^{r} that minimizes $\sum_s \mathfrak{D}_{\mathrm{KL}}(f_s \| f_s^{\mathrm{c}})$. The motivation for refining $\mathcal{A}^{\mathrm{c}}$ into $\mathcal{A}^{\mathrm{r}}$ is that our greedy clustering changes the structure of $\mathcal{A}$ by decreasing N_s to N_s^{c} following a sequence of merges with minimum local costs. However, it may be beneficial to use a global criterion to update parameters in $\mathcal{A}^{\mathrm{c}}$ by allowing the parameters of f_s^{c} to match better f_s, potentially recovering some distortion created within each f_s^{c} when merging components.

8. EXPERIMENTS

The goal is to provide clustered model $\mathcal{A}^{\mathrm{c}}$, refined or not, that can match closely the decoding performance of models trained from data, measured using WERs. The training set for our reference model is composed of 800 hours of US English data, with 10K speakers for a total of 800K utterances (4.3M words). 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 39K utterances and contains 206K words. It is a set of 47 different tasks of in-car speech with various US regional accents.

The reference model $\mathcal{A}_{100\mathrm{K}}$ is a 100K Gaussians model built on the training data. A set of 91 phonemes is used, each phoneme 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 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 using GMMs with 66 Gaussians on average. Training consists of a sequence of 30 iterations of the EM algorithm where CD state alignments are re-estimated every few steps of EM. We built 20 baseline models from training data using 5K, 10K, ..., 100K Gaussians. 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 19 models using the 100K model's STC (100K-STC) from $\mathcal{A}_{5\mathrm{K}}$ to $\mathcal{A}_{95\mathrm{K}}$. Differences in WERs for these models and the baseline are small, as shown in Table 1.

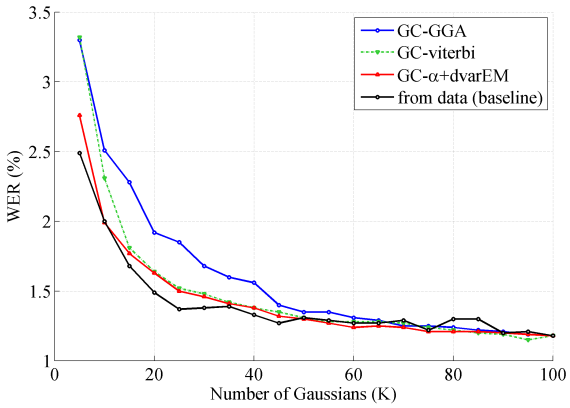

Fig. 1. WERs for models trained from data (baseline), clustered using GC-GGA, GC-viterbi, GC-α refined with dvarEM.

Baseline results show that the reference WER for $\mathcal{A}_{100\mathrm{K}}$ is 1.18%. WERs remain within 15% relative from 95K down to 40K, then start increasing significantly below 25K. At 5K, WER has increased 110% relative to WER at 100K. For each gaussian assignment strategy, we used GC with wLML to cluster $\mathcal{A}_{100\mathrm{K}}$ down to 5K, saving intermediate models every 5K Gaussians ($\mathcal{A}_{95\mathrm{K}}^{\mathrm{c}}$, ..., $\mathcal{A}_{5\mathrm{K}}^{\mathrm{c}}$), for a total of 19 clustered models for each GC-GGA, GC-α and GC-viterbi technique. GC-GGA was the first technique implemented and showed promising results. WERs stay close to the 100K-STC results from 95K-65K (sometimes even slightly improving on them), but then diverge slowly afterward and more sharply below 45K. At 5K, GC-GGA gives 3.30% WER, within 30% relative to 2.53% given by $\mathcal{A}_{5\mathrm{K}}$. In Figure 1, results for only a few techniques from Table 1 are plotted. Clearly, our ultimate goal is to follow the curve for results from baseline models for as long as possible while decreasing model size, which GC-GGA fails to do below 45K.

Results for a technique called GC-models are also reported. GC-models refers to taking the Gaussian assignments directly from the 100K-STC models trained from data. This gives us the best assignment N_s^* chosen by the training procedure. GC-models results are consistently better than that of GC-GGA over the entire 5K-95K range. GC-models is an unrealistic technique as you need to train models first to find Gaussian assignments to create clustered models of the same size. However, its results give a clear indication that Gaussian assignment is crucial to cluster the reference model optimaly, especially when creating small models. For 5K models, each f_s has an average of only 3.3 Gaussians. A good assignment is key. One difference is that our training procedure allows for splitting and merging of Gaussians within CD state during EM. Interestingly, GC-α gives similar WERs to GC-models for the entire 5K-95K range. This is not entirely surprising since it uses a similar criterion to assign Gaussians as in our training. However, only merging is allowed when clustering down $\mathcal{A}_{100\mathrm{K}}$. From 45K-95K, GC-$\alpha$ matches or improves on 100K-STC results. Below 45K, a small divergence begins and, at 5K, GC-α gives 2.87%, only within 13% of $\mathcal{A}_{5\mathrm{K}}$, a clear

WER (%) vs. Model Size (K)																				
Models	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
Baseline	2.49	2.00	1.68	1.49	1.37	1.38	1.39	1.33	1.27	1.31	1.29	1.27	1.27	1.29	1.22	1.21	1.30	1.20	1.21	1.18
100K–STC	2.53	1.89	1.70	1.51	1.39	1.35	1.34	1.30	1.32	1.28	1.28	1.28	1.25	1.23	1.16	1.23	1.20	1.23	1.22	1.18
GC-models	2.87	2.05	1.79	1.64	1.52	1.49	1.40	1.39	1.32	1.30	1.27	1.25	1.25	1.23	1.21	1.19	1.21	1.19	1.18	1.18
GC-α	2.87	2.04	1.78	1.65	1.52	1.47	1.40	1.38	1.32	1.30	1.26	1.24	1.25	1.24	1.21	1.21	1.21	1.20	1.19	1.18
GC-viterbi	3.32	2.31	1.81	1.64	1.52	1.48	1.42	1.38	1.35	1.31	1.28	1.28	1.28	1.27	1.24	1.22	1.20	1.19	1.15	1.18
GC-GGA	3.30	2.51	2.28	1.92	1.85	1.68	1.60	1.56	1.40	1.35	1.35	1.31	1.29	1.25	1.25	1.24	1.22	1.21	1.19	1.18
GC-α+dvarEM	2.76	1.99	1.77	1.63	1.50	1.46	1.41	1.38	1.32	1.30	1.27	1.24	1.25	1.24	1.21	1.21	1.21	1.20	1.19	1.18

Table 1. WERs for baseline, 100K-STC models. Models clustered with GC-models, GC-viterbi, GC-GGA, GC-α (then refined with dvarEM).

improvement over GC-GGA at 30% of $\mathcal{A}_{5\mathrm{K}}$.

GC-viterbi gives results equivalent to GC-α from 95K to 10K. For 10K, it is slightly better than GC-GGA, but it is almost the same as GC-GGA for 5K. This is counter intuitive as we expected GC-viterbi to give an "optimal" assignment and therefore better WERs. However, after analysis of $\mathcal{N}_s$ given by GC-viterbi, it is clear that the states modeling silence have much smaller N_s^{c} than for GC-α. In $\mathcal{A}_{100\mathrm{K}}$, silence states have more Gaussians than any other states, and are likely to overlap more. Therefore, $\mathcal{S}(f_s)$ for those states have merge steps with smaller costs than all other states. Adjusting for state priors is *absolutely* necessary when using GC-viterbi. Without it, the silence models are basically decimated to the benefit of the other states. By using state prior $\lambda_s^* = \lambda_s^\beta / \sum_{s'} \lambda_{s'}^\beta$ with $\beta = 0.2$ reminiscent of α-assignment, we get the best results for GC-viterbi reported in Table 1. However, WERs are really sensitive to the choice of β. If we chose $\beta = 0 < 0.2$, silence states are decimated early on in the merging steps (models $\mathcal{A}_{95\mathrm{K}}^{\mathrm{c}}$ already shows signs of decimated silence states). For small models $\mathcal{A}_{5\mathrm{K}}^{\mathrm{c}}$, they are assigned only one Gaussian each, increasing WER significantly. For $\beta = 2 > 0.2$, silence states are grabbing a large number of Gaussians which starve states modeling speech, especially in small models. Here also WERs increase rapidly.

When clustering $\mathcal{A}_{100\mathrm{K}}$ into very small models like $\mathcal{A}_{5\mathrm{K}}^{\mathrm{c}}$ (20 times smaller), achieving WERs close to 100K-STC models WERs becomes a delicate equilibrium of allocating Gaussians between speech states and silence states. This is implicitly done with $\beta = 0.2$ in (28). Since β was historically tuned for large models, one could tune it for smaller models. However, we feel that a step further should be taken to treat silence and speech states as two different categories.

To improve upon GC-α, model refinement was used using discrete varEM (dvarEM). WERs are better overall and, at 5K, GC-α with dvarEM reaches 2.76%, within 9% of $\mathcal{A}_{5\mathrm{K}}$. Over the 5K-95K range, models built from GC-α with dvarEM are on average within 2.7% of the WERs for 100K-STC models built from data. In fact, for 9 out of 19 clustered models, GC-α with dvarEM is *better* than the baseline models. This makes clustering a reference model a viable option to training models from data.

9. CONCLUSION

We presented a set of algorithmic tools for restructuring acoustic models in an ASR task, while preserving recognition performance compared to equivalent models built from data. We applied these tools to define a greedy clustering technique that can efficiently generate from a reference model smaller clustered models of any size. The clustered models have ASR performance comparable to models of same size built from training data. Advances in Gaussian assignment techniques lead to significant improvement in WER, especially for clustered models with large size reduction. For 5K, we went from being within 30% of the model built from data to 9% with GC-α with discrete varEM. We presented a greedy clustering composed of two independent steps. The first step generates the sequence of best merges for each CD state, while the second step provides a Gaussian assignment for every state. This two step approach is particularly suited for parallelization and is key to handling large models. Furthermore, this greedy clustering algorithm can generate clustered models on demand as most of the computation is done up front or 'offline'. This renders possible applications where smaller models can be built on demand from a reference model to accommodate new and changing constraints over time.

10. REFERENCES

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

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

[3] Pierre L. Dognin, John R. Hershey, Vaibhava Goel, and Peder A. Olsen, "Refactoring acoustic models using variational expectation-maximization," in *Interspeech*, September 2009, pp. 212–215.

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

[5] 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*, March 2005, pp. 669–672.

[6] Imre 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] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland, *The HTK Book (for HTK Version 3.3)*, Cambridge University Engineering Department, 2005.

[8] Etienne Marcheret, Vaibhava Goel, and Peder A. Olsen, "Optimal quantization and bit allocation for compressing large feature space transforms," *ASRU*, December 2009, to appear.