Acoustic Model Training Using Greedy EM

Rusheng Hu, Xiaolong Li, and Yunxin Zhao

Department of Computer Science
University of Missouri, Columbia, MO 65211, USA
{rhe02, xlm7b}@mizzou.edu, ZhaoY@missouri.edu

ABSTRACT

In this paper, we present a greedy EM (GEM) method for training Gaussian mixture density (GMD) based acoustic models. In the proposed approach, starting from a single Gaussian, GMD is built up by sequentially adding new components. Each new component is globally selected to avoid local optima. The sequential procedure offers more control over the model structure to achieve better coverage of data. GEM also provides a natural way of integrating information criterion for model complexity selection. Experimental results on WSJ task show that the new method performs consistently better than the conventional method in speech recognition word error rate.

1. INTRODUCTION

The EM algorithm [1] has been widely used in acoustic model training of Gaussian mixture densities for speech recognition. However, EM does not guarantee convergence to global optima and its solution is dependent on the initialization of mixture components. In Gaussian mixture modeling of phone units, local optima often involve overlapped mixture components in over-populated center regions and too few components near class boundary. A further problem associated with EM is determining the optimum number of mixture components. Since more complex models usually result in higher likelihood for training data, current systems often use a fixed, empirically determined number of components.

In this paper, we introduce a greedy EM (GEM) algorithm for training Gaussian mixture density (GMD) based acoustic models to overcome the above problems. GEM is based on recent works in the statistics literature. Li and Barron stated that a mixture model can be recursively constructed in a greedy manner, i.e., start with a single component and sequentially add in another optimal component, and the resulting model will perform as well as the true mixture density that generates the data [2]. In order to efficiently locate the optimal new component, a method which selects the best component from only a set of pre-located candidates was proposed by Verbeek and Vlassis [3]. Practically, GEM reduces the problem of learning a *k*-component mixture model to a sequential learning of two-component model, and it offers a mechanism of dynamically allocating new components outside the overpopulated center regions. GEM also allows more control on model structure by examining the location and shape of each new component before mixing it into an existing mixture and by terminating the greedy learning process when certain criterion is no longer met.

To illustrate the different effects of EM and GEM in acoustic modeling, in Figure 1, we show the 16-Gaussian component mixture densities for a sub-phonetic unit /dh/ generated by the two methods. The densities are plotted against the first two principle components derived from 39 speech feature components (details are discussed in section 5). Compared with the EM derived model, the GEM derived model has less overlaps in center regions and has more components located along the class boundary.

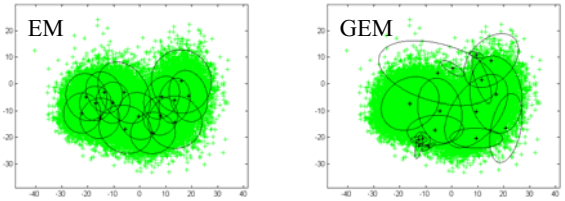

Figure 1.Two mixtures of 16 Gaussians obtained by EM & GEM.

In order to apply GEM to acoustic model training of GMD based HMMs, a three-level optimization/selection method is developed. Evaluation experiments on WSJ 20K Nov 92 task is performed on GEM in comparison with the conventional EM. The results show that the proposed GEM approach leads to consistent reduction in word error rates.

The organization of the rest of the paper is as follows. In sections 2 the greedy EM algorithm is discussed. In section 3, the greedy EM algorithm is extended for HMM training. In section 4, algorithm implementation and integration with Bayesian Information Criterion (BIC) [4] are presented. Test results are given in section 5, and conclusions are made in section 6.

2. GREEDY EM FOR GAUSSIAN MIXTURES

Let $X = (x_1, \ldots x_T)$ be i.i.d observations drawn from a mixture of multivariate normal densities of size *K*. The likelihood function is $f(X \mid \theta) = \prod_{t=1}^{T} \sum_{k=1}^{K} \alpha_k N(x_t \mid \theta_k)$ with $\theta_k = (\mu_k, \Sigma_k)$. The model parameters can be estimated by the well-known EM algorithm [1], where equations (1)-(4) are iteratively applied to all components $k = 1, 2, \ldots, K$ until convergence.

This work is supported in part by National Institutes of Health under the grant NIH 1 R01 dc04340-01A2 and the National Science Foundation under the grant NSF EIA 9911095.

$$p^{(n)}(k \mid x_t) = \frac{\alpha_k^{(n-1)} N\left(x_t \mid \theta_k^{(n-1)}\right)}{\sum_{l=1}^{K} \alpha_l^{(n-1)} N\left(x_t \mid \theta_l^{(n-1)}\right)} \quad (1)$$

$$\alpha_k^{(n)} = \frac{1}{T} \sum_{t=1}^{T} p^{(n)}(k \mid x_t) \quad (2)$$

$$\mu_k^{(n)} = \frac{\sum_{t=1}^{T} p^{(n)}(k \mid x_t) x_t}{\sum_{t=1}^{T} p^{(n)}(k \mid x_t)} \quad (3)$$

$$\sum_k^{(n)} = \frac{\sum_{t=1}^{T} p^{(n)}(k \mid x_t)\left(x_t - \mu_k^{(n-1)}\right)\left(x_t - \mu_k^{(n-1)}\right)^T}{\sum_{t=1}^{T} p^{(n)}(k \mid x_t)} \quad (4)$$

In contrast to the conventional EM, the greedy EM starts from a *k*-component mixture and learns the *(k+1)*-component mixture by finding an optimal new component. Given a k-component mixture $f_k(x)$, adding a new component $N(x;\theta_{k+1})$ yields the new mixture $f_{k+1}(x)$:

$$f_{k+1}(x) = (1-\alpha) f_k(x) + \alpha N(x;\theta_{k+1}),\ 0 < \alpha < 1 \quad (5)$$

In this case, $f_k(x)$ is fixed and only the weight α and the parameter θ_{k+1} need to be estimated. The iterative estimation of α and θ_{k+1} is therefore called partial EM. Since only one component needs to be updated in each iteration, partial EM requires much less computation than full EM, and therefore, a global search can be employed in partial EM for proper initialization of each new component. The greedy EM algorithm as proposed in [3] can be summarized as the following:

1. Initialize parameters θ_1 for the single Gaussian model, set *k*=1.
2. Find a new component θ_{k+1} and the corresponding mixing weight α by using partial EM, where f_k is fixed.
3. Set $f_{k+1}(x) = (1-\alpha) f_k(x) + \alpha N(x \mid \theta_{k+1})$, *k* = *k*+1.
4. Update f_k using EM until convergence (optional).
5. If stopping criterion is met then exit, else go to step 2.

The critical part of GEM is step 2, where a global search of the new component is made for the purpose of avoiding local optima. The global search can be approximated by an efficient heuristic algorithm that selects the best candidate among an appropriate set of pre-generated candidates. The candidates are obtained by randomly splitting the existing components. Step 4 is not proposed in the original GEM, but it is desirable to tune the model parameters after a new component is added. The stopping criterion in step 5 can be the maximum allowed number of components in each model, model complexity selection criterion such as BIC, or cross-validation. The location and shape of a new component can also be easily examined at each step [3]. More details on stopping criteria will be discussed in Section 4.

3. GREEDY EM FOR HMM

Consider an N-state HMM with parameter set $\lambda = (\pi, A, \theta)$, where π is the set of initial probabilities, A is the set of state transition probabilities, and θ is the set of parameters of emission probabilities. For a sample set $X = (x_1, \ldots x_T)$, the complete data is $y = (X, S, l)$, where S is the sequence of unobserved states, and l is the sequence of unobserved mixture component indices. As in EM, the auxiliary function in GEM can be decomposed into three independent component functions: $Q_\pi(\pi, \hat{\lambda}) = \sum_{i=1}^{N} \gamma_{i0} \log \pi_i$, $Q_A(A, \hat{\lambda}) = \sum_{i=1}^{N} Q_{a_i}(a_i, \hat{\lambda})$, and $Q_\theta(\theta, \hat{\lambda}) = \sum_{i=1}^{N} Q_{\theta_i}(\theta_i \mid \hat{\lambda})$ [5], where $\gamma_{it} = p(s_t = i \mid x, \hat{\lambda})$. The Q_{θ_i} component is in the form of

$$Q_{\theta_i}(\theta_i, \hat{\lambda}) = \sum_{t=1}^{T} \gamma_{it} \left(\frac{(1-\hat{\alpha}_i) f_{ik}(x_t) \log((1-\alpha_i) f_{ik}(x_t))}{(1-\hat{\alpha}_i) f_{ik}(x_t) + \hat{\alpha}_i N(x_t \mid \hat{\mu}_i, \hat{\Sigma}_i)} + \frac{\hat{\alpha}_i N(x_t \mid \hat{\mu}_i, \hat{\Sigma}_i) \log(\alpha_i N(x_t \mid \mu_i, \Sigma_i))}{(1-\hat{\alpha}_i) f_{ik}(x_t) + \hat{\alpha}_i N(x_t \mid \hat{\mu}_i, \hat{\Sigma}_i)} \right) \quad (6)$$

where the re-estimation formulas for α and θ_{k+1} of state *i* can be shown to be

$$p^{(n)}(i, k+1 \mid x_t) = \gamma_{it} \frac{\alpha_i^{(n-1)} N\left(x_t \mid \mu_i^{(n-1)}, \Sigma_i^{(n-1)}\right)}{(1-\alpha_i^{(n-1)}) f_{ik}(x_t) + \alpha_i^{(n-1)} N\left(x_t \mid \mu_i^{(n-1)}, \Sigma_i^{(n-1)}\right)} \quad (7)$$

$$\alpha_i^{(n)} = \frac{\sum_{t=1}^{T} p^{(n)}(i, k+1 \mid x_t)}{\sum_{t=1}^{T} \gamma_{it}} \quad (8)$$

$$\mu_i^{(n)} = \frac{\sum_{t=1}^{T} p^{(n)}(i, k+1 \mid x_t) x_t}{\sum_{t=1}^{T} p^{(n)}(i, k+1 \mid x_t)} \quad (9)$$

$$\sum_i^{(n)} = \frac{\sum_{t=1}^{T} p^{(n)}(i, k+1 \mid x_t)\left(x_t - \mu_i^{(n-1)}\right)\left(x_t - \mu_i^{(n-1)}\right)^T}{\sum_{t=1}^{T} p^{(n)}(i, k+1 \mid x_t)} \quad (10)$$

The GEM procedure is also applicable to the segmental K-means algorithm [6]. The most likely state sequence S^* is decoded by the Viterbi algorithm, and the greedy EM algorithm is used to estimate the parameters θ within each state. Let $\delta(x)$ be an indicator function, with $\delta(x) = 1,\ x = 0;\ \delta(x) = 0,$ otherwise. Then the update equations (7)-(10) are reduced to that for the case of Gaussian mixtures in each state with $\gamma_{it} = \delta(s_t - i)$.

Although both Baum-Welch and segmental K-means methods can be used for GEM HMM training, the latter is more convenient for performing global search of new Gaussian components. Although candidates of a new component can be globally generated by method of [3], in general, evaluation of the new component candidates using an entire set of training data will result in very high computation cost. An efficient strategy is to hierarchically divide the task into three levels: HMM-level optimization, state-level new component selection, and component-level candidate generation, which will be discussed in the next section.

4. APPROXIMATE GEM FOR HMM

The approximate GEM algorithm is developed based on the following observations:

a. The global search for a new component in GEM requires evaluation of a set of candidates by training data, which is unfeasible when the evaluation data set is large or the number of candidates is large.
b. GEM starts from the coarsest model, i.e., the single Gaussian model, and introduces finer models sequentially. Therefore, it is reasonable to reduce the range of training data in candidate evaluation as *k* gets larger, as in the case of sparse EM [7].
c. As shown in Figure 1, placing Gaussian components along class boundary may result in better coverage of data for classification tasks. To enhance this behavior, the influence on the class boundary from data in the center regions needs to be reduced.

The above three observations indicate that it is desirable to evaluate the candidates of each new component in a localized neighborhood and this strategy is employed in the approximate GEM to improve training speed and model quality. Computation cost prohibits using GEM for HMM training on a large training data set. Based on ideas in segmental K-means [6] and the candidate generation algorithm of [3], the proposed approach adopts a three-level search and optimization strategy: candidates are generated and evaluated within existing components, GMDs are greedily built for states, and HMMs are finally optimized by EM. The algorithm is summarized as the following three steps:

1. Train single Gaussian HMMs and segment training data to states of phone units by Viterbi segmentation.
2. Use GEM to train GMDs for individual state.
3. Re-estimate HMMs by conventional EM.

For each phone state, in order to generate candidates for the *(k+1)*th component, the training data set is quantized into *k* disjoint sets: $Q_i = \left\{ x \in X : i = \arg\max_{j=1,\ldots,k} P(j \mid x) \right\}$. Then for each set Q_i, a pair of candidates is generated by randomly splitting Q_i into two disjoint subsets. The data sample means and variances in these two sets are chosen as candidate parameters, and the initial weight for each candidate component is set to be half of the weight of $N(\bullet \mid \theta_i)$. If more candidates are needed from this component, then the random splitting process is carried out repeatedly to obtain the required number of candidates. Assuming *m* candidates are generated from each existing component, then *km* candidates are generated for the new component. Each candidate is re-estimated by using the partial EM. The candidates are first validated by their shapes (eigenvalues) and volumes (determinants) with pre-defined thresholds. Among surviving candidates, the one that gives the greatest likelihood increment when mixed into the existing mixture becomes the new member of the model.

Candidates are evaluated locally by a sparse partial EM. If a candidate is generated from the component Q_i, then it is evaluated only by data of Q_i. Specifically, the sparse algorithm approximates the likelihood of data x as $p'(x) = Cp(x)$, $x \in Q_i$; $p(x) = 0$, otherwise, where C is a normalizing constant taken as 1. Based on this approximation, the updating formulas for partial EM are put in the forms of (11-14) [3]. This approximation greatly reduces computation cost, and enables local measurement of each candidate on its capacity of modeling local pattern.

$$p^{(n)}(k+1 \mid x_t) = \frac{\alpha^{(n-1)} N\left(x_t \mid \mu^{(n-1)}, \Sigma^{(n-1)}\right)}{\left(1-\alpha^{(n-1)}\right) f_k(x_t) + \alpha^{(n-1)} N\left(x_t \mid \mu^{(n-1)}, \Sigma^{(n-1)}\right)} \quad (11)$$

$$\alpha^{(n)} = \frac{1}{T_j} \sum_{x_t \in Q_i} p^{(n)}(k+1 \mid x_t) \quad (12)$$

$$\mu^{(n)} = \frac{\sum_{x_t \in Q_i} p^{(n)}(k+1 \mid x_t) x_t}{\sum_{x_t \in Q_i} p^{(n)}(k+1 \mid x_t)} \quad (13)$$

$$\Sigma^{(n)} = \frac{\sum_{x_t \in Q_i} p^{(n)}(k+1 \mid x_t)\left(x_t - \mu^{(n-1)}\right)\left(x_t - \mu^{(n-1)}\right)^T}{\sum_{x_t \in Q_i} p^{(n)}(k+1 \mid x_t)} \quad (14)$$

The sequential procedure of adding in Gaussian components makes it natural to use an information criterion to determine the optimum model size. BIC is one of the most widely used model selection criterion, defined as $BIC(M) = \log L(X, M) - \frac{1}{2} \#(M) \times \log(n)$, where *#(M)* is the number of parameters in the model *M*, *n* is the size of the data set *X,* and $L(X, M)$ is the likelihood of *X* under *M* [4]. When BIC is used, the GEM procedure terminates if adding in new component results in a decrease of BIC value.

As discussed in section 2, an optional step in GEM is the EM step after each new component is inserted, which is also known as the "retuning" step. This step can also include post-selection of the "retuned" components.

GEM requires more computation than EM. Denote T_j as the size of the data set segmented to phone sate *j*. Assuming *m* candidates are generated from each existing mixture component, then the cost for one component search is $O(mT_j)$. Adding the cost for EM update of f_i, which is $O(iT_j)$, the sum is $O((m+i)T_j)$. The run time of training a sequence of 1 to *k* mixture models in the phone state is then $O(\sum_{i=1}^{k} T_j(m+i)) \propto O(k^2 T_j)$ if $m < k$. In total, the running time of GEM training will be $O(\sum_j T_j k^2) \propto O(k^2 T)$, where the complete training data size is $T = \sum_j T_j$, which is a factor of *k* times slower than conventional EM.

5. EXPERIMENTS

The approximate GEM algorithm was evaluated on the WSJ 20K Nov 92 task. The standard training data set (WSJ0+WSJ1) including speech of 384 speakers were used. Speech feature analysis was made at a 10msec frame rate with a 25msec window-size. Speech feature components included 13 MFCCs and their first and second derivatives. Cepstral means were removed for every utterance. The baseline acoustic model was

trained using HTK with a fixed number of Gaussians in each mixture.

The GEM based acoustic models were trained as the following. First, single Gaussian models were trained using conventional EM and were tied by phonetic decision tree with HTK [8]. Second, a Viterbi forced alignment using the trained single Gaussian models was performed to segment training data into phone states. Third, GEM models were trained for each tied state using segmented data, where the maximum allowed number of Gaussians for each phone state was 32. The number of candidates used for each component was 10. As the last step, an ordinary embedded EM was applied to all the GEM derived models by using the entire set of training data.

For the WSJ task, standard trigram language model provided by LDC was used, including 19,982 unigrams, 3,518,595 bigrams, and 3,153,527 trigrams. Only within-word triphone acoustic model was tested, even though GEM is equally applicable to cross-word triphone model. One-pass time-synchronous beam search was used for decoding speech with conservative pruning thresholds optimized for testing. Experimental results from 333 sentences of the si_et_20 evaluation set are listed in Table 1. Word accuracy achieved under the same number of mixture components per mixture density was compared for baseline and GEM derived models (for GEM model, this number is an average over all states). The last row of the table also gives the relative rate of error reduction (RER).

Mix. size	8	10	12	15	16	17
Baseline	88.37	88.66	88.59	88.84	89.31	89.33
GEM	88.96	88.92	89.14	89.54	89.86	89.77
RER	5.1%	2.3%	4.8%	6.3%	5.1%	4.1%

Table 1. Word accuracy of conventional EM and GEM

Over the range of studied model complexity, GEM trained models consistently gave lower word error rate than EM trained models, confirming the superior performance of GEM training over EM. For many states, GEM produced models with smaller number of Gaussian components than EM. For example, when mixture size was 16, nearly 50% GMDs have less than 16 Gaussians. This result confirms the fact that different phonetic units need models with different complexities under typical speech model training conditions.

Figure 2 shows a histogram of number of Gaussians in each tied state for the GEM case. We can see that there were about 20% GMDs containing 32 components. These states had large training data samples, and the likelihood of the training data in general increased as the model structure became more complex. Therefore, BIC criterion needs to be used to avoid overfitting. For comparison, BIC-based model selection was performed on EM and GEM derived GMD models on a state-by-state basis, with the maximum number of Gaussians in each state fixed to be 32. Again, speech recognition experiments were performed to evaluate the BIC selected models, and the test results are given in Table 2.

After BIC model selection, the average numbers of Gaussians per state for GEM and baseline models were reduced to 15 and 13 Gaussians, respectively. In both cases, the resulting models performed better than their without BIC counterparts. For the GEM models, the number of states containing 32 components was reduced by 40%. For the baseline models, although the maximum possible number of Gaussians was also 32, the resulting model performed worse than GEM model of similar complexity without BIC.

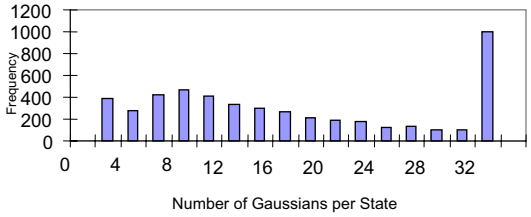

Figure 2. Histogram of number of Gaussians per state

	Mix. Size	without BIC	with BIC
Baseline	13	88.84	89.10
GEM	15	89.54	89.74

Table 2. Comparison of word accuracy from BIC results of EM and GEM derived models

6. CONCLUSION

We propose an approximate greedy EM algorithm for acoustic model training. Partial EM and sparse EM are used in GEM to enable better modeling of local data patterns and to speedup training process. Experiments conducted on WSJ 20K Nov 92 task demonstrated that the proposed algorithm consistently outperformed the conventional EM, and that BIC criterion can be naturally integrated into the sequential procedure of model growth. In future study, we will explore the possibility of using spatial information of data distribution to enhance GEM's capability of modeling detailed structures of class boundary.

REFERENCES

[1] A. P. Dempster, N. M. Laird, D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Stat.* B, Vol. 39, No. 1, 1-38, 1977.

[2] J. Q. Li and A. R. Barron "Mixture density estimation," *Advances in Neural Information processing Systems* 12, the MIT Press, 2000.

[3] J. J. Verbeek and N. Vlassis "Efficient Greedy Learning of Gaussian Mixture Models," *Neural Comp.* 5(2): 469-485, 2003.

[4] S. S. Chen and P.S. Gopalakrishnan, "Clustering via the Bayesian Information Criterion with Applications in Speech Recognition," *Proc. ICASSP,* vol. 2, 645-648, 1998.

[5] L. R. Rabiner and B.H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[6] B. H. Juang and L. R. Rabiner, "The Segmental K-Means Algorithm for Estimating Parameters of Hidden Markov Models," *IEEE Trans. ASSP*, Vol. 38. No. 9, 1639-1641 1990.

[7] R. M. Neal and G. E. Hinton, "A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants," *Learning in Graphic Models*, 355-368, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[8] The HTK Toolkit, *http://htk.eng.cam.ac.uk/.*