Gaussian Mixture Optimization for HMM based on Efficient Cross-validation

*Takahiro Shinozaki**, *Tatsuya Kawahara*

Academic Center for Computing and Media Studies, Kyoto University, Kyoto, Japan

*staka@ar.media.kyoto-u.ac.jp

Abstract

A Gaussian mixture optimization method is explored using cross-validation likelihood as an objective function instead of the conventional training set likelihood. The optimization is based on reducing the number of mixture components by selecting and merging a pair of Gaussians step by step base on the objective function so as to remove redundant components and improve the generality of the model. Cross-validation likelihood is more appropriate for avoiding over-fitting than the conventional likelihood and can be efficiently computed using sufficient statistics. It results in a better Gaussian pair selection and provides a termination criterion that does not rely on empirical thresholds. Large-vocabulary speech recognition experiments on oral presentations show that the cross-validation method gives a smaller word error rate with an automatically determined model size than a baseline training procedure that does not perform the optimization.

Index Terms: speech recognition, HMM, Gaussian mixture, cross-validation, sufficient statistics

1. Introduction

Optimizing model size and structure in Gaussian mixture model and Gaussian mixture HMM are important to achieve higher performance with a limited training set. While the model fit to the training data monotonically increases for the number of mixtures, models with too many parameters do not work for new data because the overfitting problem become prominent. In extreme cases, Gaussian mixtures become unstable and earn exorbitantly large likelihood by assigning some of the components to particular training samples with very small variances. Given a model with large mixtures, a strategy to optimize the mixture distribution is to select and merge a pair of components based on an objective function step by step until a termination criterion is satisfied. Since the optimization requires estimation of the merging score for all the combinations of the components, the score must be efficiently estimated for the algorithm to be feasible.

The most popular choice of the objective function is likelihood. The likelihood criterion has advantages that it is consistent with the overall objective of the standard HMM training and that an efficient algorithm is known to estimate the likelihood. However, a limitation is that it does not provide termination criterion to trade-off model fit vs. complexity. Because the likelihood is estimated for the training data and optimistically biased, it always decreases for the component merging. As a result, it is difficult to determine when to stop the merging process. A threshold may be used for the change in likelihood as a termination criterion but an empirical tuning is required to determine the threshold. In addition, because the instability problem of the Gaussian mixtures is originated from the same bias effect, it is difficult to remove the outlier components with the small variances by this criterion. Information theoretic criteria have been applied to HMM training [1], but in practice, it often requires an empirical tuning factor to compensate for errors in the theoretical bias estimation.

Cross-validation (CV) is a data-driven method that can largely reduce the bias in the likelihood. For K-fold cross validation, training data is partitioned into K subsets. A likelihood of a subset is evaluated by a CV model that is estimated from $K-1$ subsets excluding that subset. The likelihood evaluation is repeated for each subset and CV likelihood is obtained as their sum. In this way, it effectively separates the data used for the model parameter estimation and the likelihood evaluation and a reliable likelihood is obtained without the bias. The outlier components can not earn large likelihood and become the subjects of the merging. However, a difficulty of the CV method is the computational cost. Because the efficient algorithm was not known to estimate the CV likelihood, the application of CV had been quite restricted in HMM training.

The efficient algorithm for the conventional training set likelihood is based on the utilization of the sufficient statistics which has been used in the context of model selection such as decision tree state clustering [2] and successive state splitting [3], and in the context of selective training [4]. Recently, we have shown that the CV likelihood of a Gaussian distribution can also be efficiently evaluated using sufficient statistics and successfully applied the technique to decision-tree state clustering [5]. In this paper, we extend the algorithm to Gaussian mixtures and explore Gaussian mixture optimization based on the CV likelihood. The main difference from the typical application of CV is that CV is integrated inside of the training algorithm by utilizing the efficient evaluation algorithm rather than comparing a few models outside of a model training.

This paper is organized as follows. The Gaussian mixture optimization algorithm is described in Section 2. Experimental conditions are shown in Section 3 and the results are presented in Section 4. Finally, a summary and future works are given in Section 5.

2. Gaussian mixture merging algorithm

In this section, a Gaussian mixture merging method using conventional likelihood is first reviewed and then its extension to CV is shown. In order to distinguish from the CV likelihood presented, we call the conventional training set likelihood as self-test likelihood. The merging method explored here is based on bottom-up clustering, but the same technique to estimate CV likelihood can be applied for top-down clustering.

2.1. Self-test likelihood based method

In this method, Gaussian mixture is optimized by repeatedly selecting and merging a pair of component Gaussians based on self-test likelihood. Let θ be parameters of an input M-mixture

Gaussian distribution and $\bar{\theta}$ be parameters of $M-1$ mixture Gaussians obtained by merging one of the pairs of its components. For a diagonal Gaussian distribution, the sufficient statistics are the sum of the observation count, and the first and second order sample averages. Similarly, the sufficient statistics for m-th Gaussian component are expressed as follows:

$$A^0(m) = \sum_{t\in T}\gamma_m(t), \tag{1}$$

$$\boldsymbol{A}^1(m) = \sum_{t\in T}\boldsymbol{x_t}\gamma_m(t), \tag{2}$$

$$\boldsymbol{A}^2(m) = \sum_{t\in T}\boldsymbol{x_t}^2\gamma_m(t), \tag{3}$$

where T is a training set, t is a time, m is a mixture component index, $\boldsymbol{x_t} = (x_1(t), x_2(t), \cdots, x_d(t))^T$ is a d-dimensional feature vector at time t and $\boldsymbol{x}^2 = \left(x_1^2, x_2^2, \cdots, x_d^2\right)^T$, and $\gamma_m(t) = P(m_t|T, \theta_0)$ is occupancy count of m-th mixture at time t given a proper initial model θ_0. The mean $\boldsymbol{\mu}(m)$ and variance $\boldsymbol{v}(m)$ of the m-th Gaussian component is obtained from these sufficient statistics as follows:

$$\boldsymbol{\mu}(m) = \frac{\boldsymbol{A}^1(m)}{A^0(m)}, \tag{4}$$

$$\boldsymbol{v}(m) = \frac{\boldsymbol{A}^2(m)}{A^0(m)} - \boldsymbol{\mu}(m)^2. \tag{5}$$

By assuming that the alignment does not change during the optimization, $\bar{\theta}$ is easily obtained from sufficient statistics of θ as the sufficient statistics of the merged Gaussian are the sum of the sufficient statistics of the original Gaussian pairs.

With proper assumptions such as fixed state alignments [2], the self-test likelihood of a Gaussian mixture is expressed as follows:

$$\begin{aligned} L_{self}(\theta) &\approx \sum_{m=1}^{M}\sum_{t\in T}\{\log P(x_t|m,\theta)\}\gamma_m(t) & (6)\\ &= -\frac{1}{2}\sum_{m}\Big\{\Big(\log\Big((2\pi)^d|\boldsymbol{\Sigma}(m)|\Big)+d\Big) \\ &\qquad \cdot A^0(m)\Big\}, & (7)\end{aligned}$$

where $\boldsymbol{\Sigma}$ is a diagonal covariance matrix whose main diagonal is $\boldsymbol{v}$. Mixture weights do no affect the optimization when the alignment is fixed and thus they are omitted. Equation (7) is efficiently evaluated since the summation over t, which implies to access all the training data, is pushed in the pre-computed sufficient statistics.

For M-mixture Gaussian distribution θ, there are $\frac{M(M-1)}{2}$ possible pairs of its components. Let Θ be a set of $M-1$ mixture Gaussians obtained by merging one of the pairs. Then, the Gaussian pair merging using the self-test likelihood criterion is formulated as a model selection as follow:

$$\hat{\theta} = \operatorname*{argmax}_{\bar{\theta}\in\Theta} L_{self}\left(\bar{\theta}\right). \tag{8}$$

By repeating the same procedure, the number of Gaussians is reduced one by one. For Gaussian mixture HMMs, the optimization can be independently applied for each state.

The problems of the Gaussian merging using the self-test likelihood are that the likelihood has “optimistic” bias and is not accurate especially when the amount of training samples is not large. Because of the bias, the likelihood monotonically decreases for the mixture optimization and does not provide a termination criterion.

2.2. CV likelihood based method

For K-fold CV based merging method, the training data is partitioned into K subsets,

$$T = \bigcup_{k=1}^{K} T_k, \qquad T_i \bigcap T_j = \phi \quad (i \neq j). \tag{9}$$

Let $\boldsymbol{A}_k = \left\{A_k^0, \boldsymbol{A}_k^1, \boldsymbol{A}_k^2\right\}$ be sufficient statistics for the k-th subset. The parameters of the general model θ that is to be trained from all the training set are estimated from $\sum_{i=1}^{K}\boldsymbol{A}_i$. Similarly, the parameters of the k-th CV model θ_k are obtained from $\sum_{i\neq k}^{K}\boldsymbol{A}_i$ by holding out k-th subset from the parameter estimation.

With the same assumptions as the self-test likelihood method, the CV likelihood of θ is expressed as follow:

$$L_{cv}(\theta) = \sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{t\in T_k}\{\log P(x_t|m,\theta_k)\}\gamma_m(t). \tag{10}$$

In the equation, the k-th CV model θ_k is used to estimate the likelihood of the k-th subset T_k. Because T_k is excluded from the estimation of θ_k, this makes the data mutually independent for the model estimation and the likelihood evaluation. By substituting Gaussian distribution for $P(x_t|m,\theta_k)$ and by putting the summation over t inside, Equation (10) is rewritten as (12) that can be efficiently evaluated using the pre-computed sufficient statistics. This is the main contribution of this paper that makes it possible to apply CV to the mixture optimization with feasible computational cost. By using Equation (12) as the objective function for the component selection, CV version of the Gaussian merging algorithm is obtained. Equation (12) is a CV counterpart of the likelihood evaluation function (7). In fact, if the CV index k is omitted, Equation (12) is farther simplified and become identical to (7).

While the CV method divides the training data for the model estimation and the likelihood evaluation, the data fragmentation problem is minimum for large K since each CV model is trained from $\frac{K-1}{K}$ of the training set. Because it separates data used for the parameter estimation and the likelihood evaluation, the CV likelihood is less biased than the self-test likelihood. As the result, the likelihood behaves as if it is estimated for test data and is not monotonic for the number of mixtures. Therefore, the optimal termination point for the merging process is easily determined as a maximum point of the likelihood.

2.3. Preliminary likelihood results

Fig. 1 shows an example of the likelihood that is estimated during the Gaussian merging optimization for a certain HMM state. The initial model had 256 Gaussians as components. The components were merged using the self-test and the CV likelihood criteria with 40 subsets, respectively. The horizontal axis is the number of mixtures that decreases for the optimization and the vertical axis is the total likelihood of the mixture distribution for the training set. As can be seen, due to the optimistic bias, self-test likelihood takes a larger value than the CV likelihood. Because the self-test likelihood is monotonic for the number of mixtures, it is difficult to know when to stop the merging. On the other hand, the CV likelihood has a peak. The increase in likelihood indicates that the generality of the model is improved by reducing excessive components. As the merging process proceeds, the CV likelihood takes a maximum at some point,

$$L_{cv}\left(\bar{\theta}\right) = \sum_{k}^{K}\sum_{t\in T_k}\sum_{m}^{M}\log\left\{\frac{1}{\sqrt{(2\pi)^d\left|\boldsymbol{\Sigma}_k(m)\right|}}\exp\left(-\frac{1}{2}\left(\boldsymbol{x}_t-\boldsymbol{\mu}_k(m)\right)^T\boldsymbol{\Sigma}_k(m)^{-1}\left(\boldsymbol{x}_t-\boldsymbol{\mu}_k(m)\right)\right)\right\}\gamma_m(t) \tag{11}$$

$$= -\frac{1}{2}\sum_{k}^{K}\sum_{m}^{M}\Big\{\log\left((2\pi)^d\left|\boldsymbol{\Sigma}_k(m)\right|\right)A_k^0(m) + \left(\boldsymbol{v}_k(m)^{-1}\right)^T\boldsymbol{A}_k^2(m) - 2\left(\boldsymbol{\Sigma}_k(m)^{-1}\boldsymbol{\mu}_k(m)\right)^T\boldsymbol{A}_k^1(m) + \left(\boldsymbol{v}_k(m)^{-1}\right)^T\boldsymbol{\mu}_k(m)^2A_k^0(m)\Big\}. \tag{12}$$

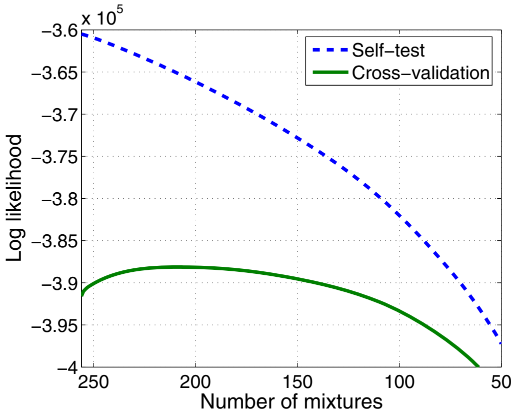

Figure 1: *Gaussian component merging and GMM likelihood.*

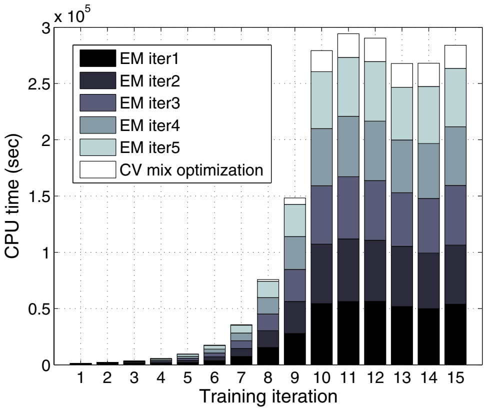

Figure 2: *Computational cost.*

which is around 210 in this case, and then it begins to decrease. The decrease in likelihood indicates that the model size is becoming too small and the Gaussian mixture is losing modeling accuracy. Therefore, in this case, the CV likelihood indicates that around 210 mixtures is appropriate to balance the modeling accuracy and the data sparseness problem.

3. Training paradigm and experimental setups

There are several possibilities of how to apply the Gaussian mixture optimization method in the HMM training. For example, it can be applied only once using a HMM with large mixtures as an input model. A problem with this strategy is that it is not obvious how to choose the number of mixtures for the initial model. The other strategy is to repeat the merging process along with mixture splitting. In this way, the initial mixture size problem is avoided. In addition, a positive effect is expected in finding better local optima as it kneads the mixtures by repeatedly absorbing unnecessary components and increasing the survived Gaussians. In this work, the latter training procedure is adopted. The HMMs were trained with the following procedure:

1. Input 1-mixture tied-state HMM as an initial model.
2. Randomize and uniformly partition the training data. Iterate EM for five times. Compute sufficient statistics for each data subset for the CV based mixture optimization method.
3. Optimize Gaussian mixtures with the CV merging method using the sufficient statistics. The number of mixture is reduced until the CV likelihood is maximized. Output HMM.
4. Split and double the number of the mixtures by duplicating the parameters with small deviation. Go to step 2.

In the following, we count step 2 through step 4 as one training iteration. The random partitioning was performed for each training iteration. If the Gaussian merging in step 3 is not performed, then the number of Gaussians in the HMM is simply doubled for each training iteration. We refer this procedure as a baseline.

The acoustic models were tied-state Gaussian mixture HMM with 1000 states. They were trained from 30 hours of a subset of the Corpus of Spontaneous Japanese (CSJ) [6]. The utterances were from academic presentations. Feature vectors had 39 elements comprising of 12 MFCC and log energy, their delta, and delta delta. The HTK toolkit [7] was used for the EM training. The language model was a trigram model trained from 6.8M words of academic and extemporaneous presentations from the CSJ. Test set was the CSJ evaluation set that consisted of 10 academic presentations given by male speakers. Speech recognition was performed using the Julius decoder [8].

4. Results

Fig. 2 shows the computational cost of the EM iterations and the mixture merging with 40-fold CV. In the figure, an training iteration consists of the five EM steps and the mixture merging optimization. The ratio of the computational cost of the merging optimization for the total training procedure was about 7%. This result shows that the proposed CV merging algorithm is efficient and well practical.

Fig. 3 plots the averaged number of mixtures per state for the training iteration. The CV based optimization methods was performed with 40 and 80 CV subsets. The baseline is the result

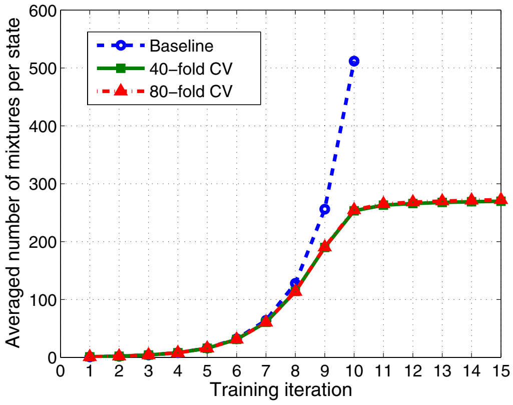

Figure 3: *Number of iterations and average number of mixtures per state.*

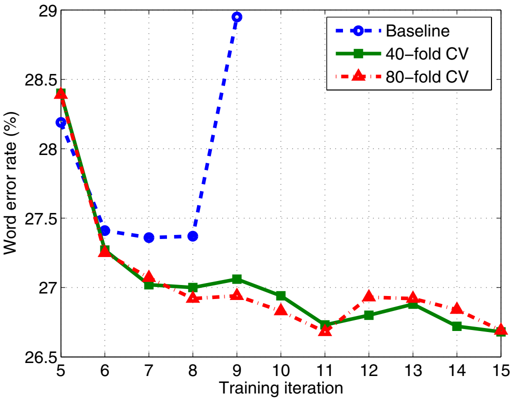

Figure 4: *Number of training iterations and test set word error rate.*

when no merging optimization was performed and the number of mixtures increased exponentially. The two CV settings gave the mostly same results. In these cases, the number of mixtures first increased exponentially as most of the components were not merged. As the number of mixtures increased to over 100, the merging process effectively started to work. After sufficient iterations, the number of merged components became equal to the number of splits and a balance in the total number of mixtures was reached.

Fig.4 shows word error rates for the training iterations. For the baseline training, the lowest word error rate of 27.4% was obtained at seventh iteration and then the performance began to decrease for the training iteration. This is because the sparseness problem arose as the model size got large. The word error rates with the CV optimization were more stable for the training iteration because it controls the number of mixtures. Both the 40-fold and the 80-fold CV gave similar results with lower word error rate than the baseline. After 15 iterations, the word error rates were 26.7% in both cases. Compared to the best result of the baseline training, the relative word error rate reduction was 2.5% and the difference was statistically significant. This improvement was obtained by optimally merging Gaussian components and determining the number of mixtures for each HMM state using the CV likelihood.

5. Summary and future works

Gaussian mixture optimization method using cross-validation (CV) likelihood was proposed. The CV likelihood is efficiently estimated by using sufficient statistics. The CV likelihood is more reliable than the conventional self-test likelihood and gives a clear termination criterion. Large-vocabulary speech recognition experiments on oral presentations showed that the CV method gave lower word error rate than the baseline.

Future work includes the comparison with information-theoretic criteria, and the combinations with the cross-validation decision-tree state clustering (CV-DTC) method and the cross-validation EM (CV-EM) algorithm [9]. The CV-EM algorithm is another application of CV for the HMM training that we had proposed. The CV-EM training is somewhat different from CV-DTC and the CV mixture optimization in that it does not use CV for model selection but for reducing the bias of the sufficient statistics. By combining these algorithms, all the likelihoods used in the basic HMM training procedures can be substituted by the CV likelihood and more robust parameter estimation is expected. While we have evaluated the proposed method on speech recognition experiments, the optimization algorithm itself is general and can be widely applicable for model estimation problem with Gaussian mixtures.

6. Acknowledgments

This study was supported by the Special Coordination Funds for Promoting Science and Technology, Ministry of Education, Culture, Sports, Science and Technology, Japan.

7. References

[1] K. Shinoda and T. Watanabe, “Acoustic modeling based on the MDL criterion for speech recognition,” Proc. EUROSPEECH, pp. 99–102, 1997.

[2] S. Young, J. Odell, and P. Woodland, “Tree-based state tying for high accuracy acoustic modelling,” Proc. ARPA Workshop on Human Language Technology, pp. 307–312, 1994.

[3] M. Ostendorf and H. Singer, “HMM topology design using maximum likelihood successive state splitting,” Computer Speech and Language, vol. 11 pp. 17–41, 1997.

[4] T. Cincarek, T. Tomoki, H Saruwatari, and K. Shikano. “Utterance-based selective training for the automatic creation of task-dependent acoustic models,” IEEE Trans. Audio, Speech, and Language Processing, Vol. 15, No. 1, pp. 150–161, 2007.

[5] T. Shinozaki, “HMM state clustering based on efficient cross-validation,” Proc. ICASSP, Toulouse, Vol. 1, pp. 1157–1160, 2006.

[6] T. Kawahara, H. Nanjo, T. Shinozaki and S. Furui, “Benchmark test for speech recognition using the corpus of spontaneous Japanese,” Proc. SSPR2003, pp. 135–138, 2003.

[7] S. Young *et al.*, “The HTK Book,” Cambridge University Engineering Department, 2005.

[8] A. Lee, T. Kawahara and S. Doshita, “An efficient two-pass search algorithm using word trellis index,” Proc. ICSLP, pp. 1831–1834, 1998.

[9] T. Shinozaki and M. Ostendorf, “Cross-validation EM training for robust parameter estimation,” Proc. ICASSP, pp. 437–440, 2007.