論文 / 著書情報
Article / Book Information

Title	Gaussian Mixture Optimization Based on Efficient Cross-Validation
Author	Takahiro Shinozaki, Sadaoki Furui, Tatsuya Kawahara
Journal/Book name	IEEE Journal of Selected Topics in Signal Processing, Vol. 4, No. 3, pp. 540-547
発行日 / Issue date	2010, 6
権利情報 / Copyright	(c)2010 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Gaussian Mixture Optimization Based on Efficient Cross-Validation

Takahiro Shinozaki, *Member, IEEE*, Sadaoki Furui, *Fellow, IEEE*, and Tatsuya Kawahara, *Senior Member, IEEE*

***Abstract*—A Gaussian mixture optimization method is developed by using the cross-validation (CV) 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 pairs of Gaussians step by step according to the objective function so as to remove redundant components and improve the generality of the model. The CV likelihood is more effective for avoiding over-fitting than is the conventional likelihood, and it provides a termination criterion that does not rely on empirical thresholds. While the idea is simple, one problem is its infeasible computational cost. To make such optimization practical, an efficient evaluation algorithm using sufficient statistics is proposed. In addition, aggregated CV (AgCV) is developed to further improve the generalization performance of CV. Large-vocabulary speech recognition experiments on oral presentations show that the proposed methods improve speech recognition performance with automatically determined model complexity. The AgCV-based optimization is computationally more expensive than the CV-based method but gives better recognition performance.**

***Index Terms*—Cross-validation, Gaussian mixture, hidden Markov model (HMM), speech recognition, sufficient statistics.**

I. Introduction

GAUSSIAN mixture distributions are used in Gaussian mixture models (GMMs) and Gaussian mixture hidden Markov models (HMMs), both of which have wide applications in speech segmentation, speech recognition, image processing, and so forth [1]–[3]. In these applications, model parameters are estimated from training data, and one general problem is how to determine the number of Gaussians for a given training data set so as to maximize model performance by balancing the model precision and parameter estimation accuracy. Since a Gaussian mixture has a hidden variable in the form of mixture weights and has many local optima, optimizing the mixture size and arranging the components are both important.

Given a Gaussian mixture with a large number of components, a strategy to optimize the mixture distribution is to reduce the number of components by iteratively selecting and merging pairs of components according to an objective function until a termination criterion is satisfied. Since optimization requires evaluating the merging scores for all combinations of components, the scores must be efficiently estimated to make the algorithm feasible.

The most popular choice for the objective function is the likelihood. Using the likelihood in model structure optimization has the advantages of being consistent with the overall objective of standard model training and of having an efficient algorithm [4] for evaluation. A limitation, however, is that the likelihood does not provide a termination criterion for balancing model fit and parameter estimation accuracy. Because the likelihood is estimated for training data and is optimistically biased, it is monotonic with respect to the number of model parameters. A threshold for the change in likelihood can be specified as a termination criterion, but empirical tuning is required to determine the threshold value. Information-theoretic criteria such as the minimum description length (MDL) provide possible termination criteria [5], [6], but in practice, they often require an empirical tuning factor to compensate for errors in the theoretical bias estimation [7]. Moreover, as an extreme case of the bias effect, a Gaussian mixture sometimes becomes unstable and earns an exorbitantly large likelihood when some of its components are assigned to particular training samples with very small variances. Because such likelihood inflation occurs without increasing the number of model parameters, an information-theoretic criterion defined as the sum of likelihood and model-size-based penalty terms loses its meaning in such a situation. Variance flooring mitigates the problem, but finding the optimal flooring threshold is not trivial [8].

These problems result from using the same data for model parameter estimation and likelihood evaluation. Cross-validation (CV) is a data-driven method that separates these data and can significantly reduce the bias in score evaluation [9]–[11]. Outlier Gaussian components covering particular training samples cannot earn large likelihoods, because the same samples do not appear in the likelihood evaluation. Therefore, by using the CV likelihood as the objective function for Gaussian mixture optimization, such abnormal components should be efficiently removed. A difficulty of the CV method, however, is its computational cost. Because no efficient algorithm to estimate the CV likelihood for Gaussian distributions is known, the application of CV has been quite restricted in GMM and HMM training [12].

An efficient algorithm to estimate the conventional training set likelihood of a Gaussian distribution for model structure optimization [4] is based on using sufficient statistics. In this paper, we show that this algorithm can be extended to estimate CV likelihood, and we apply the extended algorithm to Gaussian mix-

Manuscript received April 01, 2009; revised November 04, 2009; accepted December 24, 2009. First published April 15, 2010; current version published May 14, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Hsiao-Chun Wu.

T. Shinozaki and S. Furui are with the Department of Computer Science, Tokyo Institute of Technology, Tokyo 152-8552, Japan (e-mail: shinot@furui.cs.titech.ac.jp; furui@cs.titech.ac.jp).

T. Kawahara is with the Academic Center for Computing and Media Studies, Kyoto University, Kyoto 606-8501, Japan.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2010.2048235

ture optimization [13], [14]. The resulting CV likelihood estimation algorithm is also similar to those used as part of the successive state splitting procedure [15] and in selective training [16]. The similarity is that the likelihood is evaluated for a data set that differ from the one used for model parameter estimation, and the proposed algorithm can be regarded as an extension of those algorithms. In addition to the CV-based Gaussian mixture optimization, we also propose aggregated cross-validation (AgCV) and AgCV-based Gaussian mixture optimization [17]. AgCV is an extension of CV with higher generalization ability, making it more advantageous when larger numbers of models are compared.

This paper is organized as follows. In Section II, we briefly review CV and add some theoretical analysis. The proposed Gaussian mixture optimization algorithms are described in Section III. Our experimental setting is described in Section IV, and the results are presented in Section V. Finally, a summary and discussion of our future work are given in Section VI.

II. K-Fold Cross-Validation

K-fold cross-validation (K-fold CV) is a widely used data-driven model selection method that can assess how the performance of a statistical model will generalize to a data set independent of the training data. It works by partitioning the original training data into K subsets. Of the K subsets, a single subset is retained as the validation data for evaluating the model, and the remaining $K-1$ subsets are used to estimate the model parameters. This process is repeated K times by changing the subset used for evaluation. The K evaluation results are then averaged to generate the overall evaluation score. Since this process separates the data used for model parameter estimation and evaluation, the CV evaluation score is mostly unbiased. The data fragmentation problem is minimized by choosing a large K, since each CV model is estimated using $(K-1)/K$ of the original training data. When K is equal to the number of training samples, the method is called leave-one-out CV.

Let $T = \{x_1, x_2, \cdots, x_N\}$ be a training set consisting of N samples taken independently. Let M be a statistical model with a specific model structure, and let $\Upsilon_M(x|T)$ be the log likelihood or more generally any kind of score for a data sample x evaluated by M, whose parameters are estimated using T. Then, the expected error e_{cv} of a leave-one-out CV score $(1/N)\sum_{x_t \in T} \Upsilon_M(x_t|T \setminus \{x_t\})$ with respect to a model score based on the true data distribution $\mathop{\mathrm{E}}_{x}[\Upsilon_M(x|T)]$ is expressed by

$$e_{cv}(N) = \mathop{\mathrm{E}}_{T,|T|=N}\left[\left(\frac{1}{N}\sum_{x_t \in T}\Upsilon_M\left(x_t|T \setminus \{x_t\}\right) - \mathop{\mathrm{E}}_{x}\left[\Upsilon_M(x|T)\right]\right)^2\right] \quad (1)$$

$$\approx \frac{1}{N}\mathop{\mathrm{E}}_{T,|T|=N}\left[\mathop{\mathrm{E}}_{x}\left[\Upsilon_M^2(x|T)\right] - \left(\mathop{\mathrm{E}}_{x}\left[\Upsilon_M(x|T)\right]\right)^2\right] \quad (2)$$

$$= \mathop{\mathrm{E}}_{T,|T|=N}\left[\mathop{\mathrm{E}}_{D,|D|=N}\left[\left\{\frac{1}{N}\sum_{x_d \in D}\Upsilon_M(x_d|T) - \mathop{\mathrm{E}}_{x}\Upsilon_M(x|T)\right\}^2\right]\right] \quad (3)$$

$$= e_{\mathrm{dev}}(N, N). \quad (4)$$

By assuming that eliminating up to two training samples from T does not significantly change the model score for x, i.e., that $\Upsilon_M(x|T) \approx \Upsilon_M(x|T \setminus \{x_i\}) \approx \Upsilon_M(x|T \setminus \{x_i, x_j\})$ for arbitrary x_i and x_j in T, (1) can be rewritten as (2). Then, e_{cv} is equal to the expected error e_{dev} of an evaluation score based on an independent development set $D = \{x'_1, x'_2, \cdots, x'_N\}$, as shown by (3) and (4). Therefore, leave-one-out CV has about the same generalization ability in model evaluation as evaluation using a development set of the same size as the training set. The advantage of the leave-one-out CV method is that it does not actually require such extra data. A similar argument holds for K-fold CV with a large K.

III. Gaussian Mixture Optimization

This section first describes a framework of Gaussian mixture structure optimization that uses an objective function to select a pair of components to merge. Then, we describe efficient likelihood evaluation algorithms to formulate the objective function. For the likelihood evaluation algorithms, we first review the conventional algorithm to estimate training set likelihood and then explain the proposed cross-validation (CV) and aggregated cross-validation (AgCV) likelihood estimation algorithms. We refer to the conventional training set likelihood as the self-test likelihood to distinguish it from the proposed CV and AgCV likelihoods.

A. Gaussian Mixture Structure Optimization Framework

While the proposed CV and AgCV likelihood estimation algorithms are of general application, we specifically targeted bottom-up clustering for Gaussian mixture optimization in this study. The optimization works by taking a Gaussian mixture with a large mixture size as an input and iteratively selecting and merging pairs of its components according to an objective function until a termination criterion is satisfied, as illustrated in Fig. 1. In this optimization, $M(M-1)/2$ component pairs are subject to comparison to reduce the mixture size from M to $M-1$, and the process is iterated. Therefore, the objective function must be efficiently computable to make the optimization practical.

B. Self-Test Likelihood Method

An efficient self-test likelihood evaluation algorithm for Gaussians is based on sufficient statistics. This type of algorithm has been used in HMM state clustering [4], which assumes a single Gaussian per state. In this case, a set of HMM states corresponds to a set of Gaussians and is the subject of the optimization. It is assumed that the state alignment does not change for the training data during clustering. Similarly, a set of M Gaussians is the subject of optimization for Gaussian

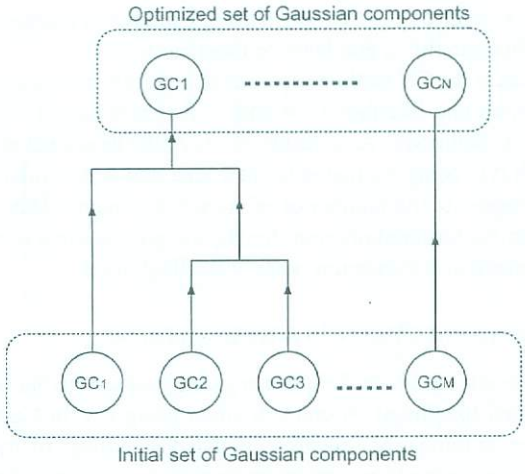


Fig. 1. Gaussian mixture optimization. Pairs of Gaussian components are merged step by step according to an objective function until a termination criterion is met.

mixture optimization, and we assume a fixed mixture assignment during optimization. These optimizations require two steps: merging multiple Gaussian distributions into a single distribution, and evaluating the likelihoods of Gaussians for the training data in order to select the Gaussians to merge. The fixed-alignment assumption enables efficient computation of both steps by using sufficient statistics. The algorithms for clustering and Gaussian merging are basically the same, but for simplicity in the following discussion, we consider Gaussian merging.

Let there be M Gaussians in a Gaussian mixture Θ, and let $\theta_m = \{\mu_m, v_m\}$ be the parameters of the mth Gaussian consisting of the mean μ_m and variance v_m. For a diagonal covariance Gaussian distribution, the sufficient statistics are the sum of the observation count, and the first and second order sample averages

$$A^0(m) = \sum_{t \in T} \gamma_m(t) \tag{5}$$

$$A^1(m) = \sum_{t \in T} x_t \gamma_m(t) \tag{6}$$

$$A^2(m) = \sum_{t \in T} x_t^2 \gamma_m(t) \tag{7}$$

where T is a training set, t is a time or frame index, $x_t = (x_1(t), x_2(t), \cdots, x_d(t))^T$ is a d-dimensional feature vector at time t, $x^2 = (x_1^2, x_2^2, \cdots, x_d^2)^T$, and $\gamma_m(t) = P(m_t|T, \Theta_0)$ is the occupancy probability of the mth mixture at time t given a proper initial model Θ_0. These sufficient statistics are computed for all the Gaussians from the training data only once before the optimization process.

Let $\theta_{\langle i,j \rangle}$ be a Gaussian formed by merging the ith and jth Gaussians. That is, $\theta_{\langle i,j \rangle}$ is a Gaussian that is estimated using a union of data assigned to either the ith or jth Gaussians. Since the sufficient statistics are weighted averages, those for the merged Gaussian are easily obtained by adding the corresponding statistics of the original Gaussians:

$$A^0(\langle i,j \rangle) = A^0(i) + A^0(j) \tag{8}$$

$$A^1(\langle i,j \rangle) = A^1(i) + A^1(j) \tag{9}$$

$$A^2(\langle i,j \rangle) = A^2(i) + A^2(j). \tag{10}$$

From these statistics, the parameters of the merged Gaussian $\theta_{\langle i,j \rangle}$ are estimated as follows:

$$\mu(\langle i,j \rangle) = \frac{A^1(\langle i,j \rangle)}{A^0(\langle i,j \rangle)} \tag{11}$$

$$v(\langle i,j \rangle) = \frac{A^2(\langle i,j \rangle)}{A^0(\langle i,j \rangle)} - \mu(\langle i,j \rangle)^2. \tag{12}$$

The mixture weight of the merged Gaussian is the sum of the weights of the original Gaussians.

The self-test log likelihood of the Gaussian mixture Θ approximated as an occupancy weighted log likelihood [4] is expressed as follows:

$$L_{\text{self}}(\Theta) \approx \sum_{m=1}^{M} \sum_{t \in T} \{\log P(x_t|\theta_m)\} \gamma_m(t) \tag{13}$$

$$= -\frac{1}{2} \sum_m \left\{ \left(\log \left((2\pi)^d |\Sigma(m)| \right) + d \right) \cdot A^0(m) \right\} \tag{14}$$

where $\Sigma(m)$ is a diagonal covariance matrix whose main diagonal is $v(m)$. When component Gaussians are merged, they are removed from the summation and the merged Gaussian is added instead. Mixture weights do not affect the optimization and are thus omitted.

Equations (11), (12), and (14) can be efficiently evaluated without directly accessing the original training data since the summation over t is pushed in the precomputed sufficient statistics. For the M-mixture Gaussian distribution Θ, there are $M(M-1)/2$ possible pairs of components. Let Ψ be a set of $M-1$ mixture Gaussian distributions obtained by merging one of the pairs. Selecting a pair of Gaussians for merging corresponds to selecting an element of Ψ. Therefore, by using the self-test likelihood, the selection is formulated as

$$\hat{\Theta}' = \underset{\Theta' \in \Psi}{\operatorname{argmax}} L_{\text{self}}(\Theta'). \tag{15}$$

By repeating the same procedure, the number of Gaussians is reduced one by one. To optimize a Gaussian-mixture HMM, the optimization can be independently applied for each HMM state.

As mentioned in the introduction, the problems of Gaussian merging using the self-test likelihood are that the likelihood has an "optimistic" bias and is especially inaccurate when the number of training samples is not large. Because of this bias, the likelihood monotonically decreases for mixture optimization and does not provide a termination criterion.

C. CV Likelihood Method

To solve the problems with using the self-test likelihood, we propose an efficient algorithm to compute the CV likelihood, and we apply it to Gaussian mixture optimization. For K-fold CV-likelihood-based Gaussian merging optimization, the training data is partitioned into K subsets of about the same size:

$$T = \bigcup_{k=1}^{K} T_k, \quad T_i \bigcap T_j = \phi \quad (i \neq j). \tag{16}$$

Let Θ be an M-mixture Gaussian distribution, and let $A_k(m) = \{A_k^0(m), A_k^1(m), A_k^2(m)\}$ be the set of sufficient statistics of the mth Gaussian component θ_m computed for the kth subset. The parameters of θ_m to be trained using all training data are estimated from $\sum_{k=1}^{K} A_k(m)$. Let Θ_k be the kth CV Gaussian mixture model corresponding to Θ, and let $\theta_{m,k}$ be the mth Gaussian of Θ_k. Similarly to the case of θ_m, the parameters $\theta_{m,k}$ are obtained from $\sum_{i \neq k} A_i(m)$ by excluding the kth subset from the parameter estimation.

With the same assumptions used for the self-test likelihood method, the CV likelihood of Θ is expressed as follows:

$$L_{cv}(\Theta) = \sum_{k=1}^{K} \sum_{m=1}^{M} \sum_{t \in T_k} \{\log P(x_t|\theta_{m,k})\} \gamma_m(t). \tag{17}$$

In the equation, the kth CV model $\theta_{m,k}$ is used to estimate the likelihood of the kth subset T_k. Because T_k is excluded from the estimation of $\theta_{m,k}$, this makes the data for model estimation and likelihood evaluation mutually independent.

By substituting a Gaussian distribution for $P(x_t|\theta_{m,k})$ and moving the summation over t inside, (17) can be rewritten as (19), which can be efficiently evaluated using the precomputed sufficient statistics:

$$\begin{aligned} & L_{cv}(\Theta) \\ & = \sum_{k}^{K} \sum_{t \in T_k} \sum_{m}^{M} \log \left\{ \frac{1}{\sqrt{(2\pi)^d |\Sigma_k(m)|}} \right. \\ & \qquad \left. \times \exp\left(-\frac{1}{2} (x_t - \mu_k(m))^T \Sigma_k(m)^{-1} \right. \right. \\ & \qquad \left. \left. \times (x_t - \mu_k(m)) \right) \right\} \gamma_m(t) \end{aligned} \tag{18}$$

$$\begin{aligned} & = -\frac{1}{2} \sum_{k}^{K} \sum_{m}^{M} \left\{ \log \left((2\pi)^d |\Sigma_k(m)| \right) A_k^0(m) \right. \\ & \qquad + \left(v_k(m)^{-1} \right)^T A_k^2(m) \\ & \qquad - 2 \left(\Sigma_k(m)^{-1} \mu_k(m) \right)^T A_k^1(m) \\ & \qquad \left. + \left(v_k(m)^{-1} \right)^T \mu_k(m)^2 A_k^0(m) \right\}. \end{aligned} \tag{19}$$

This is the main aspect of the proposed optimization method making it possible to apply CV to mixture optimization with a feasible computational cost. By using (19) as the objective function, we obtain a CV version of the Gaussian merging algorithm. Equation (19) is the CV counterpart of the likelihood evaluation function given by (14). In fact, if the CV index k is omitted, (19) is further simplified and become identical to (14).

Because the CV method separates the data used for parameter estimation and likelihood evaluation, it is less biased than the self-test likelihood. As a result, the CV likelihood behaves as though it is being estimated for new data and is not monotonic with respect to the number of mixture components. Therefore, the optimal termination point for the merging process is easily determined as a maximum point of the likelihood.

D. Aggregated CV (AgCV) Likelihood Method

Applying the CV technique can greatly reduce the bias in the estimated likelihood. A concern when using CV in Gaussian mixture optimization, however, is that the number of models subject to comparison is much larger than that in the traditional use of CV. As we showed in Section II, the generalization ability of CV is about the same as that of evaluation using a development set of the same size as the training data set. Therefore, if the number of models to compare is too large for the training set, over-fitting can still occur, depending on the specific CV configuration such as that for data distribution and partitioning. To address this problem, we propose aggregated cross-validation (AgCV), which introduces a bagging-like [18] idea into the CV framework, and we apply AgCV to Gaussian mixture optimization.

Bagging is an ensemble method to improve classification performance by integrating outputs from multiple classifiers. The multiple classifiers are trained on mutually overlapping subsets of the original training data, obtained by sampling with replacement. In the proposed AgCV algorithm, each excluded subset in K-fold CV is repeatedly processed by N models, and the resulting scores are averaged, as shown in Fig. 2. As in bagging, the N models are trained from mutually overlapping subsets defined by sub-sampling the original training set. Unlike bagging, however, a coarse sampling strategy is adopted by using the CV subsets as the unit for sampling. That is, K' subsets out of the $K-1$ subsets in the CV partitioning (i.e., those other than the excluded subset) are randomly selected without replacement N times to obtain the subsets for model estimation. The similarity among the N models is controlled by the value K'/K, which determines the amount of shared data. In this study, we specified K' as $K/2$. If $K' = K-1$ and $N = 1$, AgCV reduces to conventional CV.

The AgCV likelihood can also be efficiently computed for a Gaussian mixture Θ by using the precomputed sufficient statistics as follows:

$$\begin{aligned} L_{AgCV}(\Theta) = \frac{1}{N} \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{t \in T_k} & \{\log (P(x_t|\theta_{m,k,n})) \\ & \cdot \gamma_m(t)\} \end{aligned} \tag{20}$$

$$\mu_{k,n}(m) = \frac{\sum_{i \in \Omega_{k,n}} A_i^1(m)}{\sum_{i \in \Omega_{k,n}} A_i^0(m)} \tag{21}$$

$$v_{k,n}(m) = \frac{\sum_{i \in \Omega_{k,n}} A_i^2(m)}{\sum_{i \in \Omega_{k,n}} A_i^0(m)} - \mu_{k,n}(m)^2 \tag{22}$$

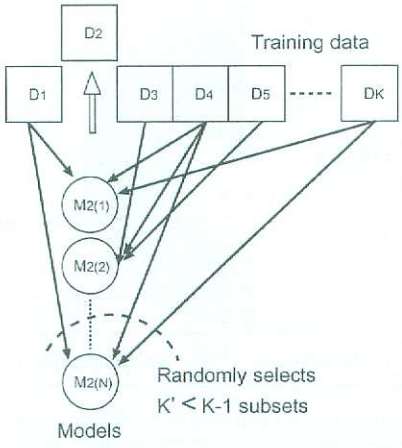


Fig. 2. Aggregated cross-validation (AgCV). Multiple models are used to evaluate an excluded subset in order to improve the generalization performance of CV.

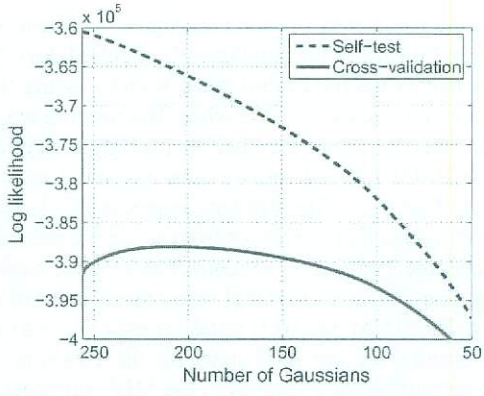


Fig. 3. Gaussian mixture optimization and the variation in the estimated GMM likelihood. Through optimization, the mixture size is decreased one by one. The self-test likelihood takes larger values than does the CV likelihood because of its optimistic bias. Unlike the self-test likelihood, the CV likelihood has a peak indicating the optimal mixture size.

where $\Omega_{k,n}$ is a set of K' integers randomly selected from $\{1, 2, \cdots K\} \setminus \{k\}$ without replacement, $\theta_{m,k,n}$ is a Gaussian component, m is a mixture index, k is a CV subset index, and n is an aggregated model index. Since (20) includes one extra summation as compared to (17), AgCV-based optimization is more computationally expensive than CV-based optimization. The coarse sampling strategy is essential for this algorithm in order to suppress the storage cost for the set of sufficient statistics. While the idea of using a bagging-like approach in AgCV is similar to our previously proposed AgEM [19], [20], these concepts differ significantly in that AgCV is a model selection method extending CV, whereas AgEM is a parameter estimation algorithm extending expectation maximization (EM) [21].

E. Preliminary Likelihood Results

Fig. 3 shows an example of the likelihood estimated during Gaussian merging optimization for a certain HMM state. The initial model had 256 Gaussian components, which were merged using both the self-test and 40-fold CV likelihood criteria. The horizontal axis represents the number of Gaussians and the vertical axis represents the total likelihood of the mixture distribution for the training set. As the graph shows, because of the optimistic bias, the self-test likelihood takes larger values than does the CV likelihood. Because the self-test likelihood is monotonic with respect to the number of Gaussians, it is difficult to know when to stop the merging. On the other hand, the CV likelihood has a peak at around 210. The increased likelihood to the left of the peak indicates that the generality of the model is improved by reducing the number of excessive components. As the merging process proceeds, the subsequent 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.

IV. Training Paradigm and Experimental Setup

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

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

In the following discussion, we count step 2 through step 4 as one 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 to that procedure as a baseline.

The random partitioning for CV was performed for each training iteration by using an utterance as an unit. In a preliminary experiment, we also evaluated partitioning with a speaker-independent constraint in which utterances from the same speaker belonged to the same subset, so that the CV score was evaluated in a speaker-independent manner. This gave a similar but slightly ($\approx 0.1\%$) worse word error rate than did partitioning without this constraint. Hence, all partitioning described here was performed without the constraint. For mixture

splitting, the parameters of each Gaussian component were duplicated, and 0.1 times the component's standard deviation was added to one of the duplicated mean vectors, while the same amount was subtracted from the other. The mixture weights for the duplicated Gaussians were half their original weights.

For the HMM training set, we used 30- and 100-hour subsets of the Corpus of Spontaneous Japanese (CSJ) [22]. The training set consists of utterances from academic presentations, with an average length of 6.7 seconds. The acoustic model was a tied-state Gaussian mixture HMM with a three-state left-to-right topology. The HMM had 1000 states for experiments using the 30-hour training set and 3000 states for the 100-hour training set. For the purpose of comparison, the MDL information-theoretic criterion was also applied.

The feature vectors had 39 elements comprised of 12 mel-frequency cepstral coefficients (MFCCs) and the log energy, their deltas, and their delta-delta. The HTK toolkit [23] was used for the EM training. The language model was a trigram model trained with 6.8 million words from academic and extemporaneous presentations in CSJ. The test set was a CSJ evaluation set consisting of ten academic presentations given by male speakers who were not included in the training set. The lengths of the presentations were about 10 to 20 minutes, and the total duration was 2.3 hours. Speech recognition was performed using the Julius decoder [24].

V. Results

Fig. 4 shows the computational costs of 15 training iterations with the CV- and AgCV-based Gaussian mixture optimizations and the 30-hour training set. Since the training process was parallelized, the cost was measured as the total user CPU time for all the parallelized processes. The value of K for the CV-based optimization was 40, while the AgCV-based optimization used $K = 6$, $K' = 3$, and $N = 10$. For the CV case, the ratio of the mixture structure optimization cost to the total training procedure cost was about 13%. This result shows that the proposed CV Gaussian merging algorithm is efficient and highly practical. For the AgCV case, the computational cost increased compared to the CV case. It was about the same, however, as the cost for the EM iterations and thus still affordable.

Table I lists the word error rates for various values of CV folds K. The table shows that stable results were obtained when K was larger than 30. Fig. 5 shows the word error rates for the training iterations when the 30-hour training set was used. Four types of experiments were performed: "EM," "EM+MDL," "EM+CV," and "EM+AgCV." "EM" was the baseline result with no Gaussian merging optimization. "EM+MDL," "EM+CV," and "EM+AgCV" were the results when the Gaussian optimization was performed using the MDL, CV, and AgCV criteria, respectively. For the MDL criterion, the tuning factor was set to 1.0 according to preliminary experiments, so as to minimize the test set word error rate. The value of K for the CV-based optimization was 30, while the AgCV-based optimization used $K = 6$, $K' = 3$, and $N = 10$. For the baseline training, the lowest word error rate of 27.4% was obtained at the seventh training iteration, and then the error rate began to increase with each additional training iteration. This is because the mixture size increases exponentially with the number of training iterations, as shown in Fig. 6, and the sparseness problem arose as the model size became large. When the structure optimization methods were applied, the model sizes were automatically controlled, and the error rates gradually stabilized with the increasing number of training iterations. The proposed CV-based method produced lower word error rates than those of baseline and MDL methods. Further improvement was obtained with the AgCV method. The lowest word error rates with MDL-, CV-, and AgCV-based optimization were 26.9%, 26.8%, and 26.4%, respectively. The relative word error rate reductions from the baseline with the CV and AgCV methods were 2.2% and 3.5%, respectively, and they were both statistically significant according to the MAPSSWE test [25]. The difference between the MDL and AgCV results was also statistically significant.

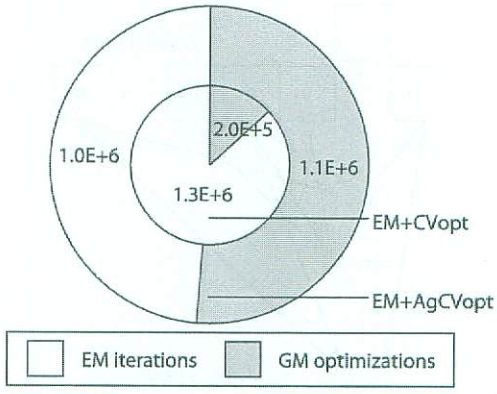


Fig. 4. Computational costs for training with EM iterations and CV Gaussian mixture optimization (inner circle), and for training with EM iterations and AgCV optimization (outer circle). Both trainings used the 30-hour data set. The CV optimization used $K = 40$, and the AgCV optimization used $K = 6$, $K' = 3$., and $N = 10$. The costs are the totals for 15 training iterations, measured in seconds.

TABLE I
Value of CV Folds K and Word Error Rate

CV folds K	5	10	20	30	40	80
Word error rate (%)	27.1	27.1	27.1	26.8	26.7	26.7

In terms of model size, EM training combined with the proposed CV- or AgCV-based Gaussian mixture optimization had a larger optimal point than those of the baseline and MDL methods. Table II lists the lowest word error rates and average number of Gaussians per state of the HMMs exhibiting those error rates. While a larger model size is a consequence of superior generalization ability in estimating model parameters from a limited amount of training data, this property has both pros and cons. The advantage is that it enables more precise model estimation from limited training data, while the disadvantage is the increased computational cost for training and decoding.

After 15 training iterations, CV optimization produced a larger model than that with AgCV optimization. This was probably because of over-fitting. To further analyze this point, different threshold values were investigated as termination criteria. That is, the Gaussian merging optimization was terminated when the increase of CV- or AgCV-likelihood by merging a pair of components was below a predefined threshold value. A zero threshold value corresponds to maximizing the

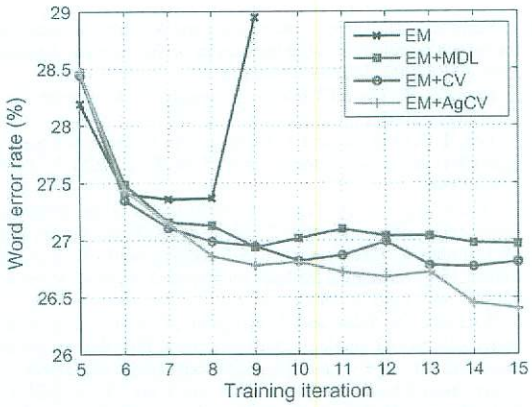


Fig. 5. Number of training iterations and test set word error rate. "EM" indicates the results when no mixture structure optimization was applied. "EM+MDL," "EM+CV," and "EM+AgCV" indicates the results with MDL-, CV- and AgCV-based optimization, respectively. The proposed CV and AgCV methods gave better results than did the conventional methods.

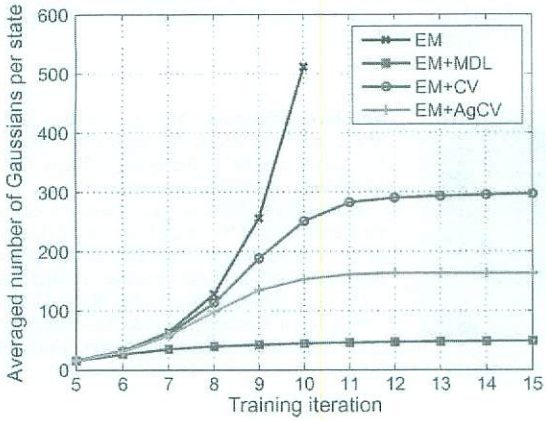


Fig. 6. Number of iterations and average number of Gaussians per state with the 30-hour training set. The CV case used $K = 30$, and the AgCV case used $K = 6, K' = 3, N = 10$.

TABLE II
LOWEST WORD ERROR RATE AND AVERAGE NUMBER OF GAUSSIANS PER STATE WITH THE 30-HOUR TRAINING SET. AN ASTERISK INDICATES THAT THE DIFFERENCE IN WORD ERROR RATE FROM THE BASELINE EM WAS STATISTICALLY SIGNIFICANT

	EM	EM+MDL	EM+CV	EM+AgCV
WER	27.4	26.9	26.8*	26.4*
# Gaussians	64	42.4	295.1	163.1

CV- or AgCV-likelihood. The smaller the threshold value, the more aggressive the Gaussian merging process becomes. Table III lists the results. CV optimization gave the lowest word error rate when the threshold value was negative. This makes sense because the negative threshold suppresses the over-fitting problem. This means, however, that it again requires empirical threshold tuning to achieve the best result. On the other hand, AgCV was more robust against over-fitting, and the best result was obtained without tuning.

Table IV lists the lowest word error rates when the 100-hour training set was used. The value of K for the CV-based optimization was 30, while the AgCV case used $K = 6$, $K' = 3$, and $N = 10$. Both optimization methods gave lower error rates

TABLE III
GAIN THRESHOLD (GTh) TO TERMINATE CV-BASED GAUSSIAN MERGING AND WORD ERROR RATE AFTER 15 TRAINING ITERATIONS. THE VALUES IN PARENTHESES ARE THE AVERAGE NUMBERS OF GAUSSIANS PER HMM STATE

GTh	-120	-60	0	30
CV	26.9 (115.4)	26.5 (178.6)	26.8 (296.8)	27.3 (380.2)
AgCV	26.9 (90.2)	26.8 (121.6)	26.4 (163.1)	26.8 (196.5)

TABLE IV
LOWEST WORD ERROR RATE AND AVERAGE NUMBER OF GAUSSIANS PER STATE WITH 100-HOUR TRAINING. AN ASTERISK INDICATES THAT THE DIFFERENCE IN WORD ERROR RATE FROM THE BASELINE EM WAS STATISTICALLY SIGNIFICANT

	EM	EM+MDL	EM+CV	EM+AgCV
WER	23.1	22.6*	22.8	22.2*
# Gaussians	32	54.4	395.2	218.1

than those of the baseline EM where no Gaussian mixture structure optimization was performed. Under this condition, however, the word error rate of the CV-based method was higher than that of MDL. This was probably because of the increased optimal mixture size. Since a larger mixture size increases the number of model comparisons in Gaussian mixture optimization, it increases the risk of choosing an over-fitted model. On the other hand, AgCV was more robust against over-fitting than was CV, giving the lowest word error rate. The relative word error rate reduction from the baseline was 2.1%, 1.4%, and 3.6% for the MDL, CV, and AgCV methods, respectively. The improvement from the baseline with AgCV was statistically significant.

VI. CONCLUSION

We have proposed a Gaussian mixture optimization method using the CV likelihood. The CV likelihood can be efficiently estimated by using sufficient statistics. It is more reliable than the conventional self-test likelihood and gives a clear termination criterion for model structure optimization. In addition, we have proposed AgCV to introduce a bagging-like idea into the CV framework to further improve the generalization ability of CV. The AgCV likelihood for Gaussian mixtures can also be efficiently computed using sufficient statistics. Large-vocabulary speech recognition experiments on oral presentations with 30 hours of training data showed that the CV- and AgCV-based methods gave lower word error rates than did MDL-based optimization. When the amount of training data was increased to 100 hours, CV-based optimization gave higher word error rates than did MDL. This was probably due to the over-fitting problem, since the number of models subject to comparison increases in optimization as the mixture size increases. On the other hand, AgCV-based optimization gave lower word error rates than did MDL, demonstrating its superior generalization ability in model selection.

Future work includes theoretical and experimental comparisons with the variational Bayesian (VB) approach [26], [27]. The VB approach has similar benefits to those of the proposed CV methods in that it has the ability to deal with the overtraining

problem and can compare models with different numbers of parameters. One advantage of the proposed CV methods with respect to VB method is that they do not require a prior distribution. Since CV is a data-driven approach, the proposed methods can be applied to evaluate other objective functions if sufficient statistics are available. In fact, a combination of our CV method with VB was recently proposed [28] in order to cross-validate prior distributions. It would also be interesting to extend our CV methods to discriminative training [29], [30]. While we have evaluated the proposed methods in terms of speech recognition, the optimization algorithms are general and should be widely applicable.

REFERENCES

[1] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Process.*, pp. 19–41, 2000.

[2] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proc. ICPR*, 2004, vol. 2, pp. 23–26.

[3] X. Anguera, T. Shinozaki, C. Wooters, and J. Hernando, "Model complexity selection and cross-validation EM training for robust speaker diarization," in *Proc. ICASSP*, 2007, vol. IV, pp. 273–276.

[4] S. Young, J. Odell, and P. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proc. ARPA Workshop Human Lang. Technol.*, 1994, pp. 307–312.

[5] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 716–723, Dec. 1974.

[6] J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Trans. Inf. Theory*, vol. IT-30, no. 4, pp. 629–638, Jul. 1984.

[7] K. Shinoda and T. Watanabe, "Acoustic modeling based on the MDL principle for speech recognition," in *Proc. Eurospeech*, 1997, vol. 1, pp. 99–102.

[8] H. Melin, J. W. Koolwaaij, J. Lindberg, and F. Bimbot, "A comparative evaluation of variance flooring techniques in HMM-based speaker verification," in *Proc. ICSLP*, Sydney, Australia, 1998, pp. 2379–2382.

[9] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*. London, U.K.: Prentice-Hall, 1982.

[10] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York: Springer-Verlag, 2001.

[11] M. Stone, "Asymptotics for and against cross-validation," *Biometrika*, vol. 64, no. 1, pp. 29–35, Apr. 1977.

[12] I. Rogina, "Automatic architecture design by likelihood-based context clustering with crossvalidation," in *Proc. Eurospeech*, Rhodes, Greece, 1997, pp. 1223–1226.

[13] T. Shinozaki, "HMM state clustering based on efficient cross-validation," in *Proc. ICASSP*, Toulouse, France, 2006, vol. I, pp. 1157–1160.

[14] T. Shinozaki and T. Kawahara, "Gaussian mixture optimization for HMM based on efficient cross-validation," in *Proc. Interspeech*, 2007, pp. 2061–2064.

[15] M. Ostendorf and H. Singer, "HMM topology design using maximum likelihood successive state splitting," *Comput. Speech Lang.*, vol. 11, pp. 17–41, 1997.

[16] T. Cincarek, T. Tomoki, H. Saruwatari, and K. Shikano, "Utterance-based selective training for the automatic creation of task-dependent acoustic models," *IEICE Trans. Inf. Syst.*, vol. E89-D, no. 3, pp. 962–969, 2006.

[17] T. Shinozaki, S. Furui, and T. Kawahara, "Aggregated cross-validation and its efficient application to Gaussian mixture optimization," in *Proc. Interspeech*, 2008, pp. 2382–2385.

[18] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[19] T. Shinozaki and M. Ostendorf, "Cross-validation and aggregated EM training for robust parameter estimation," *Comput. Speech Lang.*, vol. 22, no. 2, pp. 185–195, 2008.

[20] T. Shinozaki and T. Kawahara, "GMM and HMM training by aggregated EM algorithm with increased ensemble sizes for robust parameter estimation," in *Proc. ICASSP*, 2008, pp. 4405–4408.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc.*, no. 1, pp. 1–38, 1977, no. Series B 39.

[22] T. Kawahara, H. Nanjo, T. Shinozaki, and S. Furui, "Benchmark test for speech recognition using the Corpus of Spontaneous Japanese," in *Proc. SSPR2003*, 2003, pp. 135–138.

[23] S. Young *et al.*, *The HTK Book*. Cambridge, U.K.: Cambridge Univ. Eng. Dept., 2005.

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

[25] L. Gillick and S. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proc. ICASSP*, vol. 89, pp. 532–535.

[26] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, pp. 183–233, 1999.

[27] S. Watanabe, A. Sako, and A. Nakamura, "Automatic determination of acoustic model topology using variational Bayesian estimation and clustering for large vocabulary continuous speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 14, no. 3, pp. 855–872, May 2006.

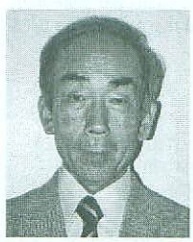
[28] K. Hashimoto, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "Bayesian context clustering using cross valid prior distribution for HMM-based speech recognition," in *Proc. Interspeech*, 2008, pp. 936–939.

[29] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. ICASSP*, 1986, pp. 49–52.

[30] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden Markov models for speech recognition," *Comput. Speech Lang.*, vol. 16, pp. 25–47, 2002.

Takahiro Shinozaki (M'07) received the B.E., M.E., and Ph.D. degrees in computer science from the Tokyo Institute of Technology, Tokyo, Japan, in 1999, 2001, and 2004, respectively.

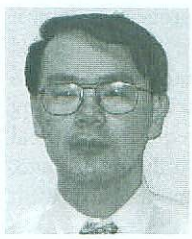
From 2004 to 2006, he was a Research Scholar in the Department of Electrical Engineering, University of Washington, Seattle. From 2006 to 2007, he was a Research Assistant Professor in the Academic Center for Computing and Media Studies, Kyoto University, Kyoto, Japan. Currently, he is an Assistant Professor in the Department of Computer Science, Tokyo Institute of Technology.



Sadaoki Furui (M'79–SM'88–F'93) received the B.S., M.S., and Ph.D. degrees in mathematical engineering and instrumentation physics from the Tokyo University, Tokyo, Japan, in 1968, 1970, and 1978, respectively.

He is engaged in a wide range of research on speech analysis, speech recognition, speaker recognition, speech synthesis, and multimodal human–computer interaction authoring or coauthoring over 800 published articles.

Dr. Furui has received Paper Awards and Achievement Awards from the IEEE, IEICE, ASJ, ISCA, and from Japan's Minister of Science and Technology and Minister of Education, as well as the Purple Ribbon Medal from the Japanese Emperor.



Tatsuya Kawahara (M'91–SM'08) received the B.E., M.E., and Ph.D. degrees in information science from Kyoto University, Kyoto, Japan, in 1987, 1989, and 1995, respectively.

Currently, he is a Professor in the Academic Center for Computing and Media Studies and an Adjunct Professor in the School of Informatics, Kyoto University. He has published more than 200 technical papers on speech recognition, spoken language processing, and spoken dialogue systems.

Prof. Kawahara was a member of the IEEE Signal Processing Society Speech Technical Committee from 2003 to 2006. He was a general chair of the IEEE Automatic Speech Recognition and Understanding workshop (ASRU-2007).