論文 / 著書情報
Article / Book Information

Title	Aggregated Cross-validation and Its Efficient Application to Gaussian Mixture Optimization
Authors	Takahiro Shinozaki, Sadaoki Furui, Tatsuya Kawahara
Citation	Interspeech2008, , , pp. 2382-2385,
Pub. date	2008, 9
Copyright	(c) 2008 International Speech Communication Association, ISCA
DOI	http://dx.doi.org/

interspeech 2008 incorporating SST 08

Aggregated Cross-validation and Its Efficient Application to Gaussian Mixture Optimization

Takahiro Shinozaki[1], *Sadaoki Furui*[1], *Tatsuya Kawahara*[2]

[1]Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan
[2]Academic Center for Computing and Media Studies, Kyoto University, Kyoto, Japan
shinot@furui.cs.titech.ac.jp

Abstract

We have previously proposed a cross-validation (CV) based Gaussian mixture optimization method that efficiently optimizes the model structure based on CV likelihood. In this study, we propose aggregated cross-validation (AgCV) that introduces a bagging-like approach in the CV framework to reinforce the model selection ability. While a single model is used in CV to evaluate a held-out subset, AgCV uses multiple models to reduce the variance in the score estimation. By integrating AgCV instead of CV in the Gaussian mixture optimization algorithm, an AgCV likelihood based Gaussian mixture optimization algorithm is obtained. The algorithm works efficiently by using sufficient statistics and can be applied to large models such as Gaussian mixture HMM. The proposed algorithm is evaluated by speech recognition experiments on oral presentations and it is shown that lower word error rates are obtained by the AgCV optimization method when compared to CV and MDL based methods.

1. Introduction

Gaussian mixture distribution is used as Gaussian mixture model (GMM) and Gaussian mixture HMM, and these models have wide applications in speaker recognition, speech recognition, etc. One of the general problems of Gaussian mixture estimation is how to decide the number of mixtures for a given training data so as to maximize the model performance by balancing the model preciseness and parameter estimation accuracy. Since a Gaussian mixture has hidden variables in the form of mixture weights and has many local optima, not only optimizing the mixture size, but also how to arrange the components is important.

Given a large mixture model, a strategy to optimize the mixture distribution is to reduce the components by iteratively selecting and merging pairs of components based on an objective function 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 to make the algorithm feasible.

The most popular choice for the objective function is likelihood. However, a limitation is that it does not provide a termination criterion to balance model fit vs. parameter estimation accuracy. Because the likelihood is estimated for the training data and optimistically biased, it is monotonic to the number of model parameters. A threshold may be used for the change in likelihood as a termination criterion but an empirical tuning is required. Information theoretic criteria provide a termination criterion, but in practice, it often requires an empirical tuning factor to compensate for errors in the theoretical bias estimation [1].

Cross-validation (CV) is a data-driven method that can largely reduce the bias by effectively separating the data used for model parameter estimation and likelihood evaluation. As it is less biased, the optimal model size is easily found as the maximum point of the score. While the traditional use of CV likelihood to structure optimization had been limited to comparing a small number of models or semi-continuous HMMs due to infeasible computational cost [2], we recently showed that the CV likelihood of Gaussian distributions can be efficiently evaluated using sufficient statistics. The CV likelihood evaluation algorithm is an extension of the self-test likelihood evaluation method used in [3] and [1], and is similar to those used in successive state splitting [4] and selective training [5] in that the likelihood is evaluated for a data set that is different from the one used for the model estimation. We have applied the CV likelihood evaluation technique to Gaussian mixture structure optimization, and have shown that it improves speech recognition performance [6].

However, a concern when using CV in the structure optimization algorithm is that the number of models subject to the comparison is much larger than that in the traditional use of CV. While CV can mostly remove the bias, the CV score statistically varies depending on data distribution, CV partitioning, etc. Among the large number of models, there may be a model that gives a higher CV score just by chance regardless of its true performance on new data. This effect increases with the number of models and degrades the model selection performance.

To reduce the variance, we propose aggregated cross-validation (AgCV) that introduces a bagging-like [7] idea to the cross-validation framework. We then apply AgCV to Gaussian mixture structure optimization and evaluate the optimization algorithm by speech recognition experiments on oral presentations. While the idea of using the bagging like approach in AgCV is similar to our previously proposed AgEM [8, 9], they are largely different in that AgCV is a model selection method that extends CV whereas AgEM is a parameter estimation algorithm that extends EM. In the following sections, we refer to conventional training set likelihood as self-test likelihood to distinguish it from likelihood estimated by the proposed method.

This paper is organized as follows. In Section 2, the AgCV algorithm is proposed that extends CV. In Section 3, AgCV is applied to Gaussian mixture structure optimization and efficient evaluation algorithm is shown. Experimental conditions are shown in Section 4 and the results are presented in Section 5. Finally, a summary and future works are given in Section 6.

Accepted after peer review of full paper

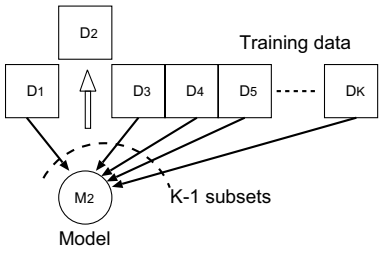

Figure 1: *K-fold cross-validation (K-fold CV).*

2. Aggregated cross-validation (AgCV) algorithm

Aggregated cross-validation (AgCV) is an extension of the K-fold cross-validation (CV) method. K-fold CV works by first dividing the training data into K subsets as shown in Figure 1. Then, it holds out one of the subsets, estimates a model using the rest of the $K-1$ subsets, and evaluates a score of the held-out subset using the estimated model. The CV score is obtained by repeating this process K times changing the held-out subset and accumulating the evaluation score. The fragmentation problem is minimum with large K, since $\frac{K-1}{K}$ of the training data is used for the model estimation. Since the overlap is avoided between the data used for the model estimation and the evaluation, a fair evaluation score is obtained without the optimistic bias in the self-test score. By using the CV score as the criterion for model structure optimization, it is possible to select a model that generalizes well to new data.

AgCV introduces a bagging-like idea into the K-fold CV to reinforce the model selection performance by reducing the variance in the score evaluation. Bagging [7] is one of the ensemble training methods to improve classification performance by integrating outputs from multiple classifiers. The multiple classifiers are trained on mutually overlapped subsets of the original training data obtained by sampling with replacement.

In the proposed AgCV algorithm, a held-out subset is repeatedly processed by N models and their scores are averaged as shown in Figure 2, while a single model is used in the conventional CV method. The N models are trained from mutually overlapped subsets defined by sub-sampling of the original training set as in bagging. However, unlike the original bagging, a coarse sampling strategy is adopted using the CV subsets as a unit for sampling. That is, K' subsets out of $K-1$ of the CV partitioning excluding the held-out subset are randomly selected without replacement for N times to obtain the subsets for the model estimation. The coarse sampling approach is useful to reduce the storage cost when applying the algorithm to sufficient statistics based structure optimization.

The similarity between the N models is controlled by $\frac{K'}{K}$ which decides the amount of shared data between the models. In this study, we experimentally fixed K' to $\frac{K}{2}$. If $K' = K-1$ and $N=1$, AgCV reduces to conventional CV.

3. Gaussian mixture structure optimization based on AgCV likelihood

The Gaussian mixture structure optimization is based on reducing extra components of an input Gaussian mixture distribution with a large number of mixtures. During the optimization, a pair of Gaussians is repeatedly selected and merged based on scores

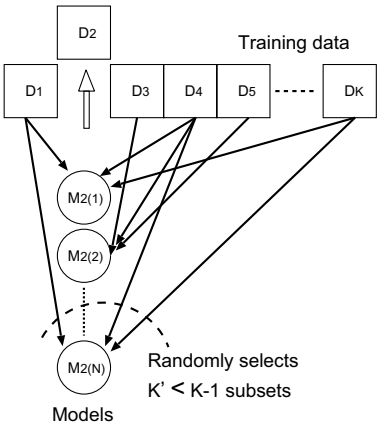

Figure 2: *Aggregated cross-validation (AgCV).*

of an objective function. At each stage, there are $\frac{M(M-1)}{2}$ possible combinations of the components for M-mixture distribution and a pair of components that gives the largest score gain for their merging is selected.

Since the number of combinations is large, the score needs to be efficiently evaluated. The efficient evaluation algorithms for the conventional self-test likelihood, previously proposed CV likelihood, and proposed AgCV likelihood are all based on sufficient statistics of Gaussian distributions, which is a set of statistics shown in Equations (1), (2), and (3).

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

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

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

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, $\boldsymbol{x}^2 = \left(x_1^2, x_2^2, \cdots, x_d^2\right)^T$, 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.

Assuming that alignments are fixed during the optimization [3], the self-test likelihood of a Gaussian mixture θ is expressed as follows:

$$L_{self}(\theta) \approx \sum_{m=1}^{M}\sum_{t\in T} \log\left(P(x_t|m,\theta)\right)\gamma_m(t) \quad (4)$$

$$= -\frac{1}{2}\sum_m \left\{\left(\log\left((2\pi)^d|\boldsymbol{\Sigma}(m)|\right) + d\right) \cdot A^0(m)\right\}, \quad (5)$$

where $\boldsymbol{\Sigma}(\boldsymbol{m})$ is a diagonal covariance matrix of m-th Gaussian component. Since the variance is obtained from the precomputed sufficient statistics as shown in Equations (6) and (7), the score can be efficiently evaluated without directly accessing the original training data.

$$\boldsymbol{\mu}(m) = \frac{\boldsymbol{A}^1(m)}{A^0(m)}, \quad (6)$$

$$diag(\Sigma(m)) = \boldsymbol{v}(m) = \frac{\boldsymbol{A}^2(m)}{A^0(m)} - \boldsymbol{\mu}(m)^2. \quad (7)$$

For the CV and AgCV based Gaussian mixture structure optimization methods, the training set is partitioned into K subsets and the sufficient statistics are estimated for each subset. Here, we denote the sufficient statistics estimated for k-th subset as $\boldsymbol{A}_k = \left\{A_k^0, \boldsymbol{A}_k^1, \boldsymbol{A}_k^2\right\}$.

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

$$L_{cv}(\theta) = \sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{t\in T_k} \log\left(P\left(x_t|m,\theta_k\right)\right)\gamma_m(t), \quad (8)$$

where θ_k is the k-th CV Gaussian mixture distribution that is estimated excluding the k-th subset. The parameters of θ_k is easily obtained by accumulating the sufficient statistics excluding $\boldsymbol{A}_k$ as shown in Equations (9) and (10).

$$\boldsymbol{\mu}_k(m) = \frac{\sum_{k\neq k}\boldsymbol{A}_k^1(m)}{\sum_{k\neq k}A_k^0(m)}, \quad (9)$$

$$\boldsymbol{v}_k(m) = \frac{\sum_{k\neq k}\boldsymbol{A}_k^2(m)}{\sum_{k\neq k}A_k^0(m)} - \boldsymbol{\mu}_k(m)^2. \quad (10)$$

By re-writing Equation (8) with the means and variances obtained by Equations (9) and (10), the CV likelihood can be efficiently evaluated using the pre-computed sufficient statistics [6].

Similar to the CV likelihood, the proposed AgCV likelihood is defined by the following equations.

$$L_{AgCV}(\theta) = \frac{1}{N}\sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{t\in T_k} \left\{\log\left(P\left(x_t|m,\theta_{k,n}\right)\right)\cdot\gamma_m(t)\right\}, \quad (11)$$

$$\boldsymbol{\mu}_{k,n}(m) = \frac{\sum_{i\in\Omega_{k,n}}\boldsymbol{A}_i^1(m)}{\sum_{i\in\Omega_{k,n}}A_i^0(m)}, \quad (12)$$

$$\boldsymbol{v}_{k,n}(m) = \frac{\sum_{i\in\Omega_{k,n}}\boldsymbol{A}_i^2(m)}{\sum_{i\in\Omega_{k,n}}A_i^0(m)} - \boldsymbol{\mu}_{k,n}(m)^2, \quad (13)$$

where $\Omega_{k,n}$ is a set of K' integers randomly selected from $\{1,2,\cdots K\}\setminus\{k\}$ without replacement. The maximum possible value for N is $C(K-1,K')$ as we require $\Omega_{k,s}\neq\Omega_{k,t}$ if $s\neq t$. The Equation (11) is again able to be efficiently evaluated using the pre-computed sufficient statistics without directly accessing the original training data with the computational cost linear in N. A Gaussian mixture HMM can be optimized by applying the optimization method independently at each state.

Fig. 3 shows an example of the likelihood that is estimated during the optimization for a certain HMM state. The initial model had 200 Gaussians as components. The components were merged step by step using the self-test and the AgCV likelihood criteria with $K=6, N=10$. The horizontal axis is the number of mixtures that decreases by the optimization and the vertical axis is the total likelihood of the mixture distribution for the training set.

As can be seen, self-test likelihood takes a larger value than the AgCV likelihood due to the optimistic bias and it monotonically decreases during the optimization. On the other hand, the AgCV likelihood is a good approximation of the model performance to new data with smaller bias. The optimal model size is easily found as the peak of the likelihood. The AgCV likelihood indicates that around 100 mixtures is appropriate to balance the model estimation accuracy and the model preciseness.

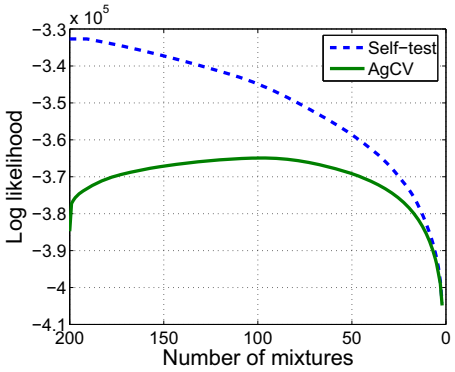

Figure 3: *An example of the objective scores estimated for training data by the Gaussian mixture structure optimization methods.*

4. Training paradigm and experimental setups

The proposed AgCV based Gaussian mixture structure optimization algorithm was applied to HMM training and evaluated by speech recognition experiments. In this study, it was integrated in the HMM training process as follows.

1. Input 1-mixture tied-state HMM as an initial model.
2. Randomize and uniformly partition the training data. Run five EM iterations to update model parameters. Compute sufficient statistics for the AgCV method.
3. Optimize Gaussian mixtures with the AgCV structure optimization method. The number of mixtures is reduced until the AgCV likelihood is maximized. Output the HMM or continue to step 4.
4. Split and double the number of the mixture components by duplicating the parameters with small deviation. Go to step 2.

The random partitioning was performed for each training iteration, consisting of step 2 through step 4, to avoid unnecessary dependencies between the consecutive CV-based structure optimizations. If the Gaussian merging in step 3 is not performed, the number of Gaussians in the HMM is simply doubled for each training iteration. We refer this procedure as a baseline. For the purpose of comparison, MDL information theoretic criterion and previously proposed CV based structure optimizations were also performed. The tuning factor for the MDL based method was set to 1.0 based on preliminary experiments so as to maximize the test set word error rate. The CV optimization method was performed with $K=30$ and the AgCV optimization used $K=6$, $N=10$.

The HMMs were tied-state Gaussian mixture HMM with 1000 states. They were trained from 30 hours of academic presentations which was a subset of the Corpus of Spontaneous Japanese (CSJ) [10]. Feature vectors had 39 elements comprising of 12 MFCCs and log energy, and their delta, and delta delta values. The HTK toolkit [11] 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. The length of each presentation is about 10 to 20 minutes and the total duration is 2.3 hour. Speech recognition was performed using the Julius decoder [12].

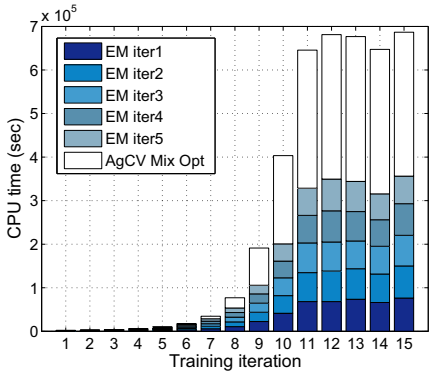

Figure 4: *Computational cost.*

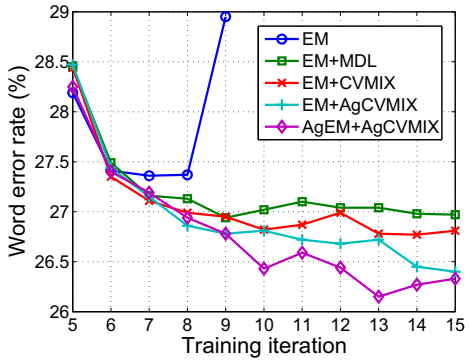

Figure 5: *Number of training iterations and test set word error rates.*

5. Experimental results

Fig. 4 shows the CPU time spent for the 5 EM iterations and the AgCV mixture structure optimization at each training iteration. While AgCV structure optimization is heavier than the CV method, the computational cost is still affordable as can be seen in the figure.

Fig.5 shows word error rates for the training iterations. In the figure, "EM" is the baseline result without the Gaussian mixture structure optimization. "EM+MDL", "EM+CVMIX", "EM+AgCVMIX" are the results with the Gaussian mixture structure optimization by the MDL, CV, and AgCV methods, respectively. "AgEM+AgCVMIX" is the result when AgEM [8, 9] was used instead of EM in combination with the AgCV structure optimization method. The AgEM setting was $K = 12, K' = 6, N = 12$. 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 each additional training iteration. This is because the sparseness problem arose as the model size got large. When the structure optimization methods are used, the model sizes are automatically controlled and the error rates gradually settle with the increase of the training iterations. Among the structure optimization methods, the proposed AgCV gave the lowest error rate demonstrating the superiority as the structure optimization method. Further improvement was obtained by combining AgCV with AgEM. The lowest error rates by the EM+AgCVMIX and AgEM+AgCVMIX training was 26.4% and 26.2%, respectively. Compared to the lowest word error rate of the baseline training, the relative word error rate reduction was 3.5% for EM+AgCVMIX and 4.4% for AgEM+AgCVMIX.

6. Conclusions

We have proposed Aggregated CV method that extends conventional CV and successfully applied the algorithm to Gaussian mixture structure optimization. The proposed Gaussian mixture structure optimization method works efficiently using sufficient statistics. In the experiments, it has been shown that lower word error rates than conventional methods are obtained by the proposed AgCV optimization method with automatically determined model sizes. Further improvement was obtained by combining the AgCV Gaussian mixture structure optimization method with the AgEM parameter estimation algorithm. Future works include evaluating the proposed algorithm with larger training data and applying AgCV to other structure optimization methods.

7. Acknowledgments

This work was supported by KAKENHI (19700167).

8. References

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

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

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

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

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

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

[7] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[8] 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.

[9] T. Shinozaki and M. Ostendorf, "Cross-validation and aggregated EM training for robust parameter estimation," *Computer speech and language*, vol. 22, no. 2, pp. 185–195, 2008.

[10] 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.

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

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