論文 / 著書情報
Article / Book Information

Title	Unsupervised Acoustic Model Adaptation Based on Ensemble Methods
Author	Takahiro Shinozaki, Yu Kubota, Sadaoki Furui
Journal/Book name	IEEE journal of Selected Topics in Signal Processing, Vol. 4, No. 6, pp. 1007-1015
Issue date	2010, 12
DOI	http://dx.doi.org/10.1109/JSTSP.2010.2076010
URL	http://www.ieee.org/index.html
Copyright	(c)2010 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.
Note	このファイルは著者（最終）版です。 This file is author (final) version.

Unsupervised Acoustic Model Adaptation Based on Ensemble Methods

Takahiro Shinozaki, *Member, IEEE*, Yu Kubota, and Sadaoki Furui, *Fellow, IEEE*

***Abstract*—We propose unsupervised cross-validation (CV) and aggregated (Ag) adaptation algorithms that integrate the ideas of ensemble methods, such as CV and bagging, in the iterative unsupervised batch-mode adaptation framework. These algorithms are used to reduce overtraining problems and to improve speech recognition performance. The algorithms are constructed on top of a general parameter estimation technique such as the maximum-likelihood linear regression method. The proposed algorithms are also useful for suppressing the negative effects of unsupervised adaptation, which reinforces the errors included in the hypothesis used for the adaptation. Experiments are performed using clean and noisy speech recognition tasks with several conditions. We show that both our proposed unsupervised adaptation algorithms give higher performance than the conventional batch-mode adaptation algorithm; however, the unsupervised CV adaptation algorithm is more advantageous than the unsupervised Ag adaptation algorithm in terms of computational cost. The proposed algorithms resulted in 4% to 10% relative reduction in the word error rate over the conventional batch-mode adaptation.**

***Index Terms*— Acoustic model, cross-validation, ensemble methods, speech recognition, unsupervised adaptation.**

I. Introduction

SPOKEN utterances largely vary with conditions such as speakers and environments. Therefore, the ability to adapt a speaker-independent general model to target utterances in an unsupervised manner is especially important to achieve high recognition performance in speech recognition. Batch-type unsupervised adaptation is generally performed by first running an automatic recognizer to derive a transcription of the target utterances, and then parameter estimation algorithms are applied using that transcript [1]. Based on the adapted model, this process is often iterated for lower recognition error rates [2].

Difficulties of this process are that the amount of adaptation data is usually limited, and the transcript made by the recognizer includes errors. Adaptation techniques are designed to manage these problems by effectively reducing the number of free parameters to improve the generalization ability of the adaptation. However, since the modeling flexibility is affected by the free parameter reduction, there is a tradeoff between this and the opportunity to precisely adapt the model to the target data. Therefore, though controlling the number of free parameters is effective, the problems are not completely alleviated and there is room for further improvement. We propose unsupervised cross-validation (CV) and aggregated (Ag) adaptation algorithms for improving the generalization performance of the batch-mode unsupervised adaptation technique by introducing CV [3] and bagging-like [4] ideas to the iterative model update process [5], which is similar to the CV-based gradient estimation algorithm for maximum mutual information (MMI) training [6] and to our previously proposed cross-validation expectation-maximization (CV-EM) and aggregated expectation-maximization (Ag-EM) supervised training algorithms [7], [8]. These algorithms are different from the typical applications of machine ensemble methods [9] in that multiple models are used inside an iterative training process.

The proposed algorithms are constructed on top of a conventional parameter estimation method, such as maximum-likelihood linear regression (MLLR) [10], and are basically independent from the details of how the method estimates parameters. Therefore, they have potentially broad applications in iterative unsupervised adaptation not limited to speech recognition. In this paper, however, we focus on using the proposed algorithms for MLLR and maximum *a posteriori* (MAP)-based [11] acoustic model adaptation.

The organization of this paper is as follows. In Section II, the conventional batch-mode unsupervised adaptation framework is briefly reviewed and the proposed unsupervised CV and Ag adaptation algorithms are explained. In Section III, an efficient variant of the unsupervised CV adaptation algorithm specialized for MLLR adaptation is proposed. Experimental conditions are described in Section IV and the results are shown in Section V. Conclusions and future work are given in Section VI.

II. Unsupervised Adaptation Algorithms

In this section, we briefly review the conventional batch-mode unsupervised adaptation framework, and explain our proposed unsupervised cross-validation (CV) and aggregated (Ag) adaptation algorithms. Although these algorithms are general, we assume speaker adaptation in a speech recognition system for simplicity.

A. Conventional Batch-Mode Adaptation

Fig. 1 shows a typical procedure of iterative conventional batch-mode unsupervised speaker adaptation. The first step is

Manuscript received August 21, 2009; revised November 27, 2009; accepted February 19, 2010. Date of publication September 13, 2010; date of current version November 17, 2010. This work was supported in part by the Microsoft CORE5 project by the Institute for Japanese Academic Research Collaboration (IJARC). The Drivers' Japanese Speech Corpus in a Car Environment was recorded by Asahi Kasei Corp. under the "Development of Fundamental Speech Recognition Technology" project supported by the Japanese Ministry of Economy, Trade, and Industry. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiaodong He.

The authors are with the Department of Computer Science, Tokyo Institute of Technology, Tokyo, 152-8552, Japan (e-mail: shinot@furui.cs.titech.ac.jp; kubomail@furui.cs.titech.ac.jp; furui@furui.cs.titech.ac.jp).

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.2010.2076010

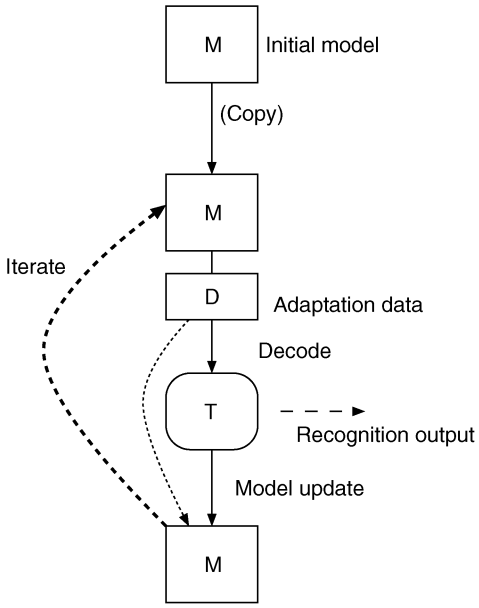


Fig. 1. Batch-mode unsupervised adaptation. M is model, T is recognition hypothesis, and D is adaptation data. Hypothesis T is made from data D using model M. Using that hypothesis, parameters of model M are updated. Updated model is used to recognize same data in next iteration.

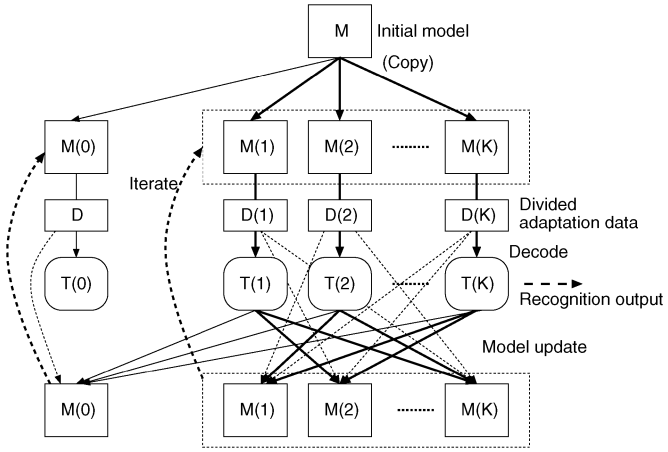


Fig. 2. Unsupervised CV adaptation. M is model and D is target data. $M(k)$ is kth CV model, $D(k)$ is kth exclusive data subset, and $T(k)$ is recognition hypothesis of that subset using $M(k)$. $M(0)$ denotes global CV model and $T(0)$ denotes hypothesis by $M(0)$.

to decode a set of target utterances from a speaker using either an initial model, if it is the first iteration, or otherwise using an adapted model made in the previous step. By running the decoder, a hypothesized transcript is obtained. The second step is to perform model parameter updates based on the hypothesis. The details of how to update the parameters depend on underlying adaptation techniques, such as MLLR. This process is iterated several times to achieve higher adaptation performance. The final recognition result is obtained by outputting the hypothesis made in the last decoding step.

In this procedure, the over-fitting problem cannot be avoided. Once a model parameter is biased to a specific data sample, the bias is reinforced in the subsequent adaptation iterations since the same data is used in the decoding and model update steps. Moreover, it is unavoidable to have recognition errors in the recognition hypothesis. These errors are also reinforced during the iteration. These problems decrease the efficiency of the adaptation.

B. Cross-Validation (CV) Adaptation

The proposed unsupervised CV adaptation algorithm reduces the problems of batch-mode adaptation by effectively separating the data used in the decoding and model update steps based on the K-fold CV technique. Fig. 2 shows the procedure of the unsupervised CV adaptation algorithm. In this procedure, the target utterances are divided into K exclusive subsets $(D(1), D(2), \cdots, D(K))$ so that each subset has roughly the same size. The first decoding step is basically the same as the batch-mode adaptation, and the K subsets are processed using the same initial model. Then, given K subsets $(T(1), T(2) \cdots, T(K))$ of the recognition hypotheses for the utterances, K CV models $(M(1), M(2), \cdots, M(K))$ are made by excluding one of these subsets, instead of making a single model. As an initial model for estimating the kth CV model, the kth CV model of the previous stage is used. Each model is used in the next decoding step to make a new hypothesis for the data subset that has been excluded from the parameter estimation of that model. The decoding and the model update steps are repeated as in the conventional batch-mode adaptation, and the final recognition hypothesis is obtained by gathering the hypotheses of the K subsets made in the last decoding step. With this procedure, the data used for the decoding and for the model parameter estimation are effectively separated, minimizing the undesired effect of reinforcing the bias. Because the utterances used for model estimation are not decoded with that model, there is no chance that the same recognition error is repeated with that model. Each CV model is estimated from a union of $K-1$ exclusive subsets, each of which includes $1/K$ of the original adaptation data. Therefore, the amount of data used to estimate a CV model is $(K-1)/K$ of the original data, and the data fragmentation problem is minimal for large K. For example, if K is 20, then 95% of the original adaptation data is used to estimate each CV model. Optionally, a global CV model $(M(0))$ can be made in the update step together with the CV models by using all recognition hypotheses. The global CV model is useful when a single adapted model is required as an output of the adaptation process.

The unsupervised CV adaptation algorithm is similar to the CV-based gradient estimation algorithm for MMI training [6] and to our previously proposed CV-EM algorithm [7], which extends EM [12], in that CV is introduced in an iterative parameter estimation process. Compared to this CV-EM algorithm, the decoding step corresponds to the E-step, and the parameter update step corresponds to the M-step. The differences are that the proposed unsupervised CV adaptation algorithm is an unsupervised training algorithm, and the CV framework is used to obtain recognition hypotheses rather than estimating gradients, as in the MMI training, or sufficient statistics for true transcripts as in CV-EM. The model update step can be performed using any kind of parameter estimation method. The computational

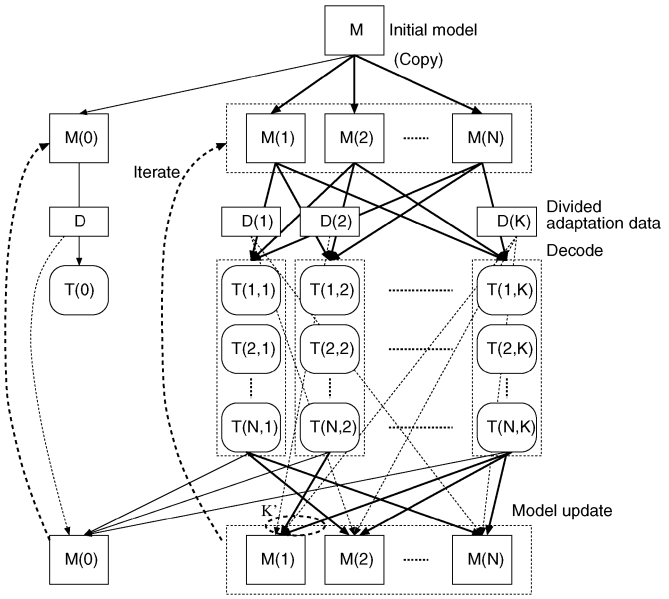


Fig. 3. Ag adaptation. M is model and D is target data. $M(n)$ is nth Ag model, $D(k)$ is kth exclusive data subset, and T(n,k) is recognition hypothesis of kth subset made using nth Ag model. $M(0)$ denotes global Ag model and $T(0)$ denotes hypothesis by $M(0)$.

cost for the decoding step is constant for K except for the overhead due to reading K different models. The computational cost for the update step is proportional to K.

If $K = 2$, CV adaptation is also similar to cross-adaptation [13]. The difference is that CV adaptation is performed on a single recognition system, whereas cross-adaptation requires two. While cross-adaptation uses transcripts from two systems representing different views of the same data based on different features and/or decoding algorithms, CV adaptation uses different data of the same view.

C. Aggregated (Ag) Adaptation

Fig. 3 describes the unsupervised Ag adaptation algorithm. Unlike CV adaptation, Ag adaptation allows overlap between the data used in the decoding and the update steps. Instead, the generalization ability is obtained through aggregating N models as in the bagging method. More specifically, the target utterances are first divided into K exclusive subsets $(D(1), D(2), \cdots, D(K))$. Then, each data subset is repeatedly decoded using N models $(M(1), M(2), \cdots, M(N))$. Initially, these N models are prepared just by copying an initial model. In the update step, N models are made using NK' hypotheses from $K'(< K)$ subsets that are randomly selected without replacement. Depending on the underlying adaptation method, the observation counts may be normalized by N in the parameter estimation since N hypotheses from the same utterance are used simultaneously. The N models are then used in the next decoding step.

Ag adaptation is an extension of Ag-EM [7] and has the same similarities and dissimilarities as in CV adaptation and CV-EM. Another difference is that, a single recognition hypothesis with the adapted models needs to be output in Ag adaptation. For this, the N different hypotheses from the last decoding step of the same subset can be integrated using recognizer output voting for error reduction (ROVER) [14] or confusion network combination (CNC) [15]. Another option is to make a special model that integrates all the transcripts from all the subsets along with the N models and output a recognition result using that model. We adopted the latter strategy and refer to the model as a global Ag model. The computational cost for the update step is $O(N^2K'/K)$.

III. Variant for Efficient MLLR Adaptation

While the unsupervised CV and Ag adaptation algorithms are independent from an underlying parameter estimation method, their computational cost for the model update step can be reduced in some cases by using the details of the estimation method with a small modification to their algorithms. In this section, we first overview the MLLR algorithm for mean transformation [10], which is widely used in speech recognition, and then propose an efficient variant of the unsupervised CV adaptation algorithm specialized for the MLLR method.

A. MLLR Algorithm

In MLLR adaptation, mean vectors of a set of Gaussian mixture hidden Markov models (HMMs) are classified into M classes, and the transformation shown in (1) is estimated for each class so as to maximize the likelihood of adaptation data with the adapted HMM, where m is a class index, m_r is a Gaussian component index belonging to mth class, $\boldsymbol{W_m}$ is a transformation matrix, $\boldsymbol{\xi_{m_r}} = [1, \boldsymbol{\mu}_{\boldsymbol{m_r}}^T]^T$ is an extended mean vector consisting of a constant term and an original mean vector $\boldsymbol{\mu_{m_r}}$, and $\boldsymbol{\mu'_{m_r}}$ is a transformed mean vector:

$$\boldsymbol{\mu'_{m_r}} = \boldsymbol{W_m}\boldsymbol{\xi_{m_r}}. \tag{1}$$

Given a set of adaptation utterances, the optimal transformation $\boldsymbol{W_m}$ is obtained by solving (2) as follows:

$$\sum_t \sum_{m_r} \gamma_{m_r}(t)\boldsymbol{\Sigma_{m_r}^{-1}}\boldsymbol{o}(t)\boldsymbol{\xi_{m_r}^T} = \sum_t \sum_{m_r} \gamma_{m_r}(t)\boldsymbol{\Sigma_{m_r}^{-1}}\boldsymbol{W_m}\boldsymbol{\xi_{m_r}}\boldsymbol{\xi_{m_r}^T} \tag{2}$$

where $\boldsymbol{\Sigma_{m_r}}$ is a covariance matrix of the m_rth Gaussian component of the original model, $\boldsymbol{o}(t)$ is an observation vector at time t, and $\gamma_{m_r}(t) = P(q_{m_r}(t)|\lambda, \boldsymbol{O})$ is an occupation count of being at the Gaussian mixture component q_{m_r} at time t given HMM model parameters λ and the observation sequence $\boldsymbol{O}$.

Transformation estimation using (2) can be divided into two steps. The first step is an accumulation step expressed in (3), and the second is a transformation estimation step that solves (4) as follows:

$$\begin{aligned} A^0_{m_r} &= \sum_t \gamma_{m_r}(t) \\ \boldsymbol{A^1_{m_r}} &= \sum_t \{\gamma_{m_r}(t)\boldsymbol{o}(t)\}. \end{aligned} \tag{3}$$

$$\sum_{m_r} \boldsymbol{\Sigma_{m_r}^{-1}} \boldsymbol{A^1_{m_r}} \boldsymbol{\xi_{m_r}^T} = \sum_{m_r} A^0_{m_r} \boldsymbol{\Sigma_{m_r}^{-1}} \boldsymbol{W_m} \boldsymbol{\xi_{m_r}} \boldsymbol{\xi_{m_r}^T}. \tag{4}$$

While the accumulation step requires summation over observation sequences, and the computational cost is linear to the

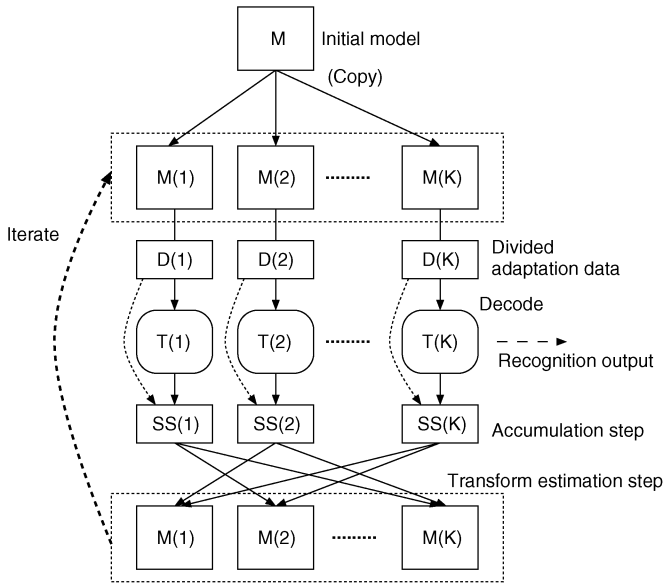


Fig. 4. Efficient variant of unsupervised CV adaptation specialized for MLLR. $M(k)$ is kth CV model, $D(k)$ is kth exclusive data subset, and $T(k)$ is recognition hypothesis of that subset using $M(k)$. $SS(k)$ is set of MLLR sufficient statistics for $D(k)$ using $M(k)$ and $T(k)$.

amount of data, the cost for the transformation estimation step is constant. Therefore, for a large amount of data, the computational cost is dominated by the accumulation step.

B. Efficient Unsupervised CV Adaptation

Fig. 4 shows the procedure of the proposed efficient variant of the unsupervised CV adaptation algorithm for MLLR. The differences from the original CV algorithm are that the MLLR model update procedure is split into two steps, and the data exchange for the CV operation is performed between the two steps. That is, the MLLR statistics defined by (3) are accumulated in the accumulation step for each CV subset using the recognition hypothesis of that subset and a corresponding CV model. Then, the MLLR transforms for the kth CV model are estimated in the estimation step described by (4) by gathering all the statistics excluding the one for the kth subset. The new kth CV model is made by applying the estimated transforms to the kth CV model of the previous epoch.

The accumulation of the MLLR statistics is based on an assumption that the alignments using the CV models correspond. In the proposed CV adaptation algorithm, this assumption holds for large K. Because an arbitrary pair of the CV models $(M(i), M(j))$ share $K-2$ sets of the MLLR statistics out of $K-1$ in their parameter estimation, the models are quite similar, except that they are open to a different data subset.

In this procedure, the computational cost for the MLLR accumulation step is constant for the number of CV folds K, except the overhead of reading multiple models, since each input utterance is processed only once while in the original version it is processed $K-1$ times using different CV models. Therefore, when the computational cost of MLLR is dominated by the accumulation step, the model update step of the efficient version works with only $1/(K-1)$ of the original cost.

IV. Experimental Setup

A. Corpora and Initial Acoustic Models

To evaluate the proposed algorithm in various conditions, two test sets and three training conditions were used. The first test set was the official evaluation set of the Corpus of Spontaneous Japanese (CSJ) [16]. This test set consists of ten academic oral presentations given by different male speakers. The length of each presentation was about 10 to 20 minutes and the total duration was 2.3 hours. This data is referred to as CSJ test set. To evaluate this test data with domain-matched as well as unmatched recognition conditions, spontaneous, and read speech training data were used. The spontaneous training data was from the training set portion of the CSJ corpus containing 254 hours of academic oral presentations. As a speaker-independent baseline acoustic model, a tied-state Gaussian mixture triphone HMM set with 3000 states and 32 mixtures per state was estimated using this data. The read speech training data was from the Japanese News Article Sentences (JNAS) corpus [17] and consisted of 52 hours of gender-independent data. Using the data, a speaker-independent initial triphone HMM set with 2000 states and 32 mixtures per state was constructed. These models were estimated using the EM and minimum phone error (MPE) [18] training methods. Both Mel-frequency cepstral coefficient (MFCC) [19] and perceptual linear prediction (PLP)-based [20] acoustic models were constructed. The MFCC-based features had 39 elements comprising 12 MFCCs and log energy, and their delta [21] and delta-delta values. Similarly, the PLP-based features consisted of 12 PLPs and log energy, and their delta and delta-delta values.

The second test set was from the "Drivers' Japanese Speech Corpus in a Car Environment" corpus [22] and consisted of 20 male speakers and 20 female speakers who were professional drivers. The utterances were voice commands to a car navigation system, and there were a total of 108 utterances per speaker. They were recorded inside a car in idling mode, driven in a city, or driven on a highway. Depending on the driving conditions, the signal-to-noise ratio (SNR) varied from -10 to 20 dB. The total amount of data per speaker was about six minutes. This data is referred to as a noisy test set. The speaker-independent initial model used to recognize this noisy test data was a tied-state Gaussian mixture triphone HMM. It was first trained on 52 hours of clean speech data from the JNAS corpus and then adapted to noisy speech conditions using 1795 CSJ utterances that were randomly mixed with 28 noises from the JEIDA-NOISE corpus [23], including car noises at seven different SNRs. The HMM had 2000 states and 16 mixtures per state. The feature vectors had 38 elements consisting of 12 MFCCs, their delta plus delta log energy, and delta-delta values. Spectral subtraction [24] was performed both in the estimation of the initial speaker-independent noisy speech model and recognition of the evaluation data. The noise vector for the spectral subtraction was estimated using the first ten frames of each speech segment.

B. Recognition Systems

Most of the recognition experiments were performed using the weighted finite-state transducer (WFST)-based T^3 decoder

[25] with the MFCC-based features. An exception was an evaluation of cross-adaptation in which the Julius decoder [26] and PLP-based features were also used together with an MFCC-based T^3 system. Julius is a two-pass decoder with stack decoding, while T^3 is one pass.

The hidden Markov model toolkit (HTK) [27] was used for the HMM training and for adaptation using the MLLR and MAP methods. MLLR adaptation was performed using regression class trees with 32 leaf nodes with the default settings of the toolkit except Ag adaptation, in which the occupancy threshold to determine the number of MLLR adaptation classes was multiplied by N for normalization because N hypotheses from the same data are used in Ag adaptation. For the efficient variant of the CV MLLR adaptation, a modified version of the toolkit was used to support the algorithm. Other than that, the original version was used without any modification in the source code.

The recognition system for the large vocabulary CSJ test set used a trigram language model trained from 6.8 million words of academic and extemporaneous presentations from the CSJ. The dictionary size was 30 k. The recognition system for the noisy test set was based on a network grammar with a vocabulary of 300 words.

V. Experimental Results

The proposed unsupervised ensemble adaptation algorithms were first evaluated in the context of unsupervised MLLR speaker adaptation using the CSJ test set and the CSJ trained acoustic model to investigate basic characteristics. Since the adaptation algorithms rely on a recognition hypothesis, there was a question of how their adaptation performance would be affected by the accuracy of the initial recognition results. To answer this question, domain-mismatched experiments were performed, where the read speech JNAS model was used as an initial model to recognize the spontaneous speech CSJ test set. Next, the proposed algorithms were evaluated for an unsupervised speaker-independent domain adaptation task using the JNAS model as an initial model. In addition to the experiments using the large-vocabulary clean CSJ task, small vocabulary noisy-speech recognition experiments were performed using the noisy test set that was recorded in actual car environments to see how the proposed algorithms worked in different tasks. Finally, the proposed unsupervised CV adaptation algorithm was compared with cross-adaptation, which is known to be useful in improving adaptation performance.

A. Speaker Adaptation

The proposed unsupervised ensemble-based MLLR speaker adaptation was performed using the CSJ test set and the EM-based CSJ initial model with MFCC-based features. Fig. 5 shows the relationship between the number of CV folds K of the CV adaptation algorithm and the word error rates averaged over the speakers. CV adaptation gave lower word error rates than the baseline conventional batch-mode adaptation for all the CV conditions. The best results were obtained when K was greater than ten. This is because when K is small, the amount of effective adaptation data is reduced for model parameter estimation. As the value of K increases, stable results are obtained since $(K-1)/K$ of the data is used in the model parameter estimation.

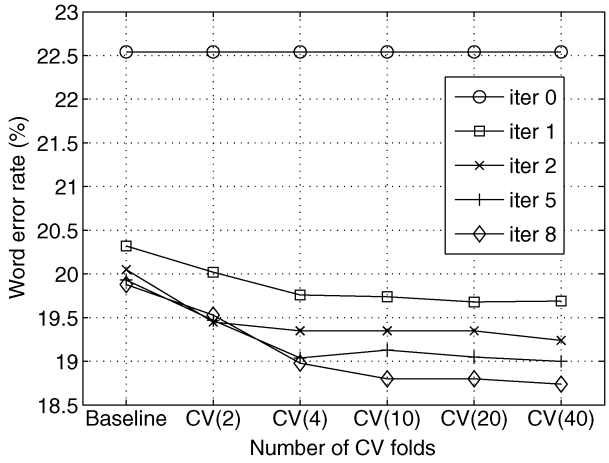


Fig. 5. Number of cross-validation folds K of CV adaptation and recognition performance. K-fold CV adaptation is denoted as CV(K). "iter" is number of adaptation iterations. Zeroth iteration is result of speaker-independent model. Baseline conventional batch-mode adaptation result is denoted as Baseline.

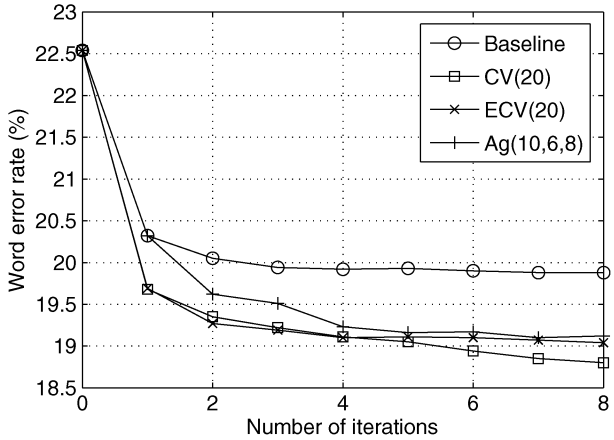


Fig. 6. Number of adaptation iterations and recognition performance of CV and Ag adaptation. Zeroth iteration is result of speaker-independent model. Batch-mode baseline adaptation result is denoted as baseline, CV ($K = 20$), ECV ($K = 20$), and Ag ($K = 10$, $K' = 6$, $N = 8$) adaptations are denoted as CV(20), ECV(20), and Ag(10,6,8).

Fig. 6 shows the number of iterations and word error rates of the CV and Ag adaptation algorithms. CV adaptation was performed with $K = 20$, and Ag adaptation was performed with $K = 10$, $K' = 6$, and $N = 8$. Ag adaptation gives the same result as baseline batch-mode adaptation for the first iteration. This is because the recognition hypothesis from Ag adaptation is obtained from the global Ag model. After the first iteration, Ag adaptation gave lower word error rates than the baseline, and the CV algorithm gave a lower error rate than the baseline from the first iteration. The efficient variant of CV adaptation, referred to as ECV, performed almost the same as the original version for the first few iterations. However, it gave a slightly higher error rate than the original CV adaptation for iterations larger than five. This is probably because of the modification of the CV algorithm to compute MLLR-sufficient statistics and because of the different treatment of the transformation classes in the current implementation in which common classes are used over

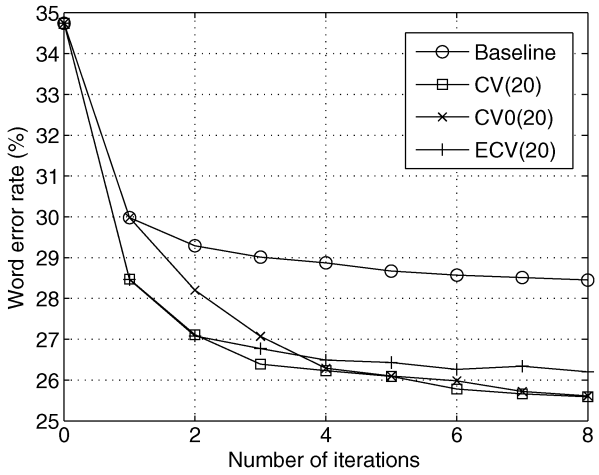


Fig. 7. Number of adaptation iterations and recognition performance of CV and ECV adaptations in unmatched training and test conditions. Zeroth iteration is result of speaker-independent model. Baseline conventional batch-mode adaptation result is denoted as Baseline, CV ($K = 20$), and ECV ($K = 20$) adaptations are denoted as CV(20), and ECV(20). CV0(20) is result of CV(20) with global model.

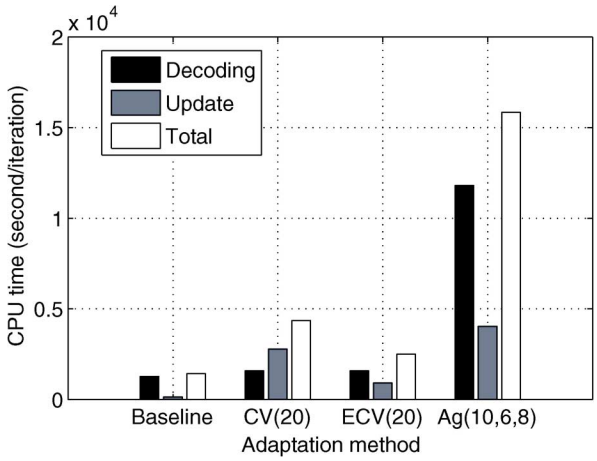


Fig. 8. CPU time of adaptation algorithms. CPU time is measured for decoding and update steps at each epoch and averaged over adaptation iterations.

the different CV subsets. After eight iterations, baseline, CV, ECV, and Ag adaptations gave 12%, 17%, 16%, and 15% relative reductions in the word error rate from 22.5% error rate of the speaker-independent initial model, respectively. Compared to the batch-mode baseline, CV, ECV, and Ag adaptations gave 6%, 4%, and 4% relative reductions, which were statistically significant with the matched pair sentence segment word error (MAPSSWE) test [28].

Because unsupervised adaptation is based on a recognition hypothesis, adaptation performance is affected by the accuracy of initial recognition results. To observe the performance of the proposed algorithms in an unmatched training and test condition, the JNAS read speech model was used as the initial model to recognize the CSJ test set. Fig. 7 shows the results. While the same CSJ test set was used as for the iterations shown in Fig. 6, the initial word error rate was much higher because of the mismatch of the training and test domains. As can be seen in Fig. 7, CV adaptation again gave significantly larger improvements than with the conventional batch-mode adaptation. Compared to the baseline adaptation, CV, CV with a global model, and ECV adaptations gave 10%, 10%, and 8% relative reductions, respectively.

Fig. 8 shows the observed CPU time in recognizing the CSJ test set using the CSJ initial model. The CPU times are for the decoding and the update steps averaged over the adaptation iterations and the speakers. As mentioned in Section II, CV adaptation has roughly the same computational cost for the decoding step as that of baseline conventional batch-mode adaptation. The cost for the update step is proportional to K, but because adaptation is generally cheaper than decoding, the total cost of the 20-fold CV adaptation was about three times higher than that of baseline adaptation. On the other hand, the computational cost of Ag adaptation is generally higher than that of baseline adaptation in both the decoding and update steps. The total computational cost of Ag adaptation with $K = 10$, $K' = 6$, and $N = 8$ was 11 times higher than that of baseline adaptation. For the 20-fold ECV adaptation, the cost for the update step was about one-third that of the original version. As a result, the total cost of ECV was only 1.8 times higher than that of baseline conventional batch-mode adaptation.

B. Domain Adaptation

The proposed ensemble-based unsupervised adaptation algorithms can be applied not only to speaker adaptation but also to domain adaptation. Unsupervised domain adaptation is useful when domain-dependent speech data is available but transcription is not.

Unsupervised domain adaptation experiments were performed using the CSJ test set and the JNAS model as the domain-independent initial model. Speaker-independent domain adaptation data consisted of 1 to 10 hours of academic presentations from the CSJ training set. The domain-independent initial model was adapted using the domain adaptation data using the MAP method with five iterations. Since a single model is required as the output of the domain adaptation, CV adaptation made a global CV model in the last model update step. Fig. 9 compares the adaptation performance of the conventional batch-mode and the proposed CV adaptation algorithms by recognizing the CSJ test set using the domain-adapted speaker-independent models. For comparison, supervised adaptation results are also shown in this figure. As can be seen, unsupervised CV domain adaptation gave a lower word error rate than the baseline conventional batch-mode adaptation. Unsupervised CV adaptation with 10 hours of unlabeled data performed better than supervised adaptation with two hours of labeled data.

Unsupervised domain adaptation can be combined with unsupervised speaker adaptation. Fig. 10 shows the word error rates when the domain-adapted speaker-independent model was used as the initial model for unsupervised MLLR speaker adaptation. The zeroth iteration is the result from the domain-adapted models using 10 hours of domain adaptation data. The figure plots four lines representing the combinations of the conventional baseline and CV adaptation algorithms for domain and speaker adaptation. This suggests that the improvement by CV domain adaptation compared to the baseline conventional batch-mode adaptation remains after unsupervised speaker adaptation.

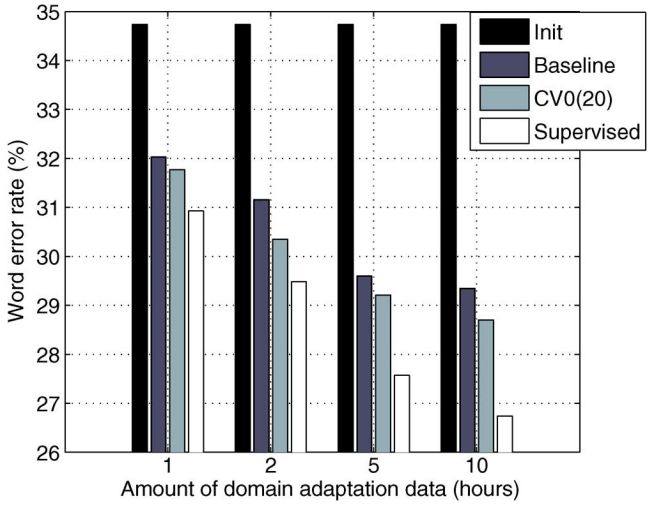


Fig. 9. Unsupervised domain adaptation using conventional batch-mode baseline adaptation and proposed CV adaptation algorithms. "Init" is result of initial domain-independent model. Supervised adaptation results are also shown for comparison.

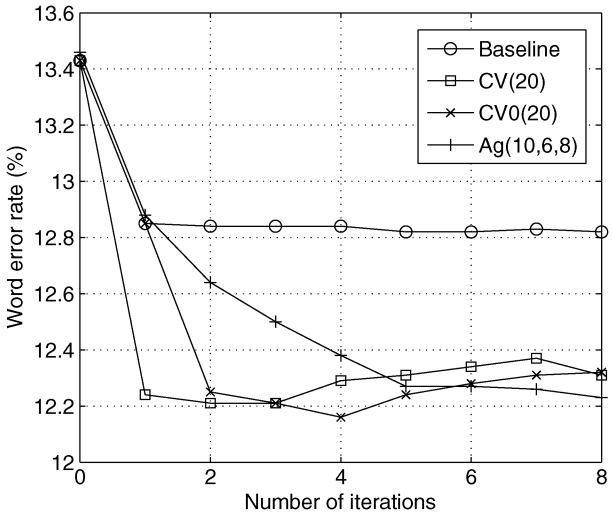


Fig. 11. Unsupervised speaker adaptation using noisy speech data from actual car environments as recognition task.

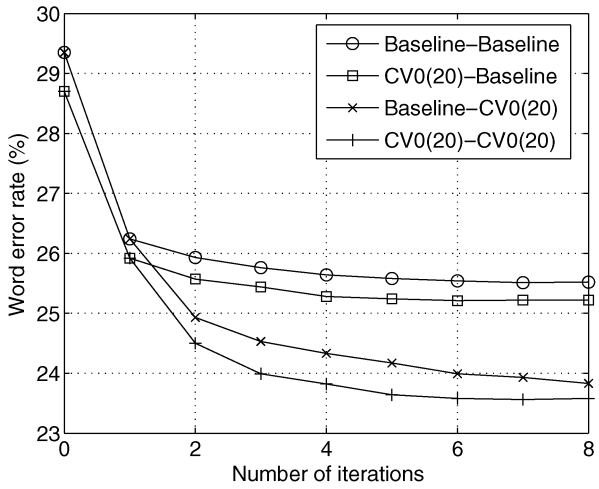


Fig. 10. Unsupervised speaker adaptation after domain adaptation. Unsupervised speaker adaptation was performed using domain-adapted speaker-independent model as initial model. In legend, "algorithm1-algorithm2" denotes domain adaptation with algorithm 1 and speaker adaptation with algorithm 2.

Improvements by CV adaptation for unsupervised speaker adaptation were larger than for domain adaptation. This is probably because the adaptation and the test data are identical in unsupervised speaker adaptation, whereas they are different in domain adaptation. Therefore, the effect of using CV was more direct in unsupervised speaker adaptation than in domain adaptation. When the baseline adaptation algorithm was used for domain and speaker adaptation, the word error rate was 25.5% after eight iterations of speaker adaptation. On the other hand, it was 23.6% when the CV adaptation algorithm was used for domain and speaker adaptation. The relative reduction in the word error rate by CV-based domain and speaker adaptation was 7.6% compared to that of baseline adaptation result.

C. Adaptation in Noisy Speech Condition

Fig. 11 shows the results of speaker adaptation using the noisy test set. The initial model was the noisy speech model from the multi-condition training with spectral subtraction. CV adaptation was performed with $K = 20$, and Ag adaptation was performed with $K = 10$, $K' = 6$, and $N = 8$. It can be seen that both CV and Ag adaptation showed larger improvements than conventional batch-mode baseline adaptation. A slight increase in the error rate was observed for CV adaptation when the number of iterations was more than three, while the error rates were still smaller than those of the baseline. After eight iterations, the relative reductions in the word error rate from the initial model by the baseline, CV, CV with a global model, and Ag adaptations were 5%, 8%, 8%, and 9%, respectively. Compared to the baseline adaptation result, the relative reductions in the word error rate by the CV, CV with a global model, and Ag adaptations were 4%, 4%, and 5%, respectively.

D. Comparisons With Cross-Adaptation

When there are two base systems, cross-adaptation [13] is known to be useful in improving adaptation performance in which recognition hypotheses are exchanged between the two systems used as transcriptions for model parameter updates. The success of cross-adaptation depends on how the two systems are different. Apparently, cross-adaptation gives the same result as conventional batch-mode adaptation if the two systems are identical. At the same time, the two systems need to have comparable performance. If one of the component systems has significantly lower performance than the other, it could harm overall performance.

To run cross-adaptation, we trained a PLP-based acoustic model using the CSJ training set and developed a Julius decoder-based recognition system in addition to the T^3-based recognition system as the base systems. Table I lists the word error rates of these systems for the CSJ test set when the speaker-independent initial models were used. The PLP-based T^3 system gave a lower error rate with a smaller computational cost than the PLP-based Julius system. This was probably because T^3 was based on WFST, and the search strategy was mathematically more organized than conventional decoders. Note that the T^3 system with MFCC features used in this

TABLE I
BASE SYSTEMS FOR CROSS-ADAPTATION

Decoder	Feature	WER
T^3	MFCC	19.3
T^3	PLP	19.7
Julius	PLP	21.1

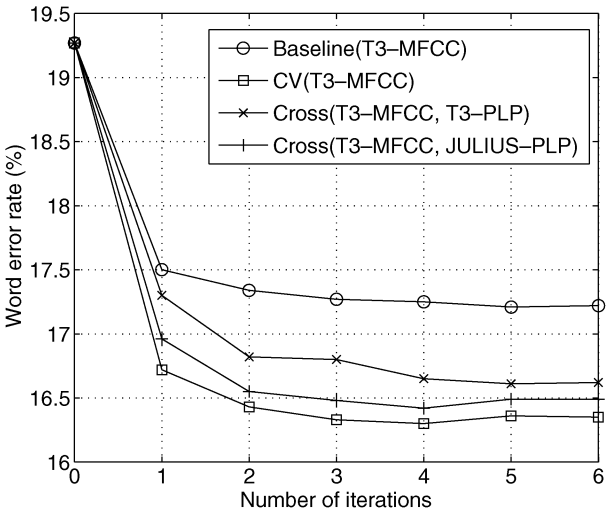


Fig. 12. Unsupervised speaker adaptation by CV adaptation and cross-adaptation.

experiment was a newer version than the system used in the previous sections and had lower initial word error rates. The changes included the use of the MPE trained model and tunings of the decoder. The PLP model was also estimated using the MPE method.

Two composite recognition systems were investigated for cross-adaptation. The first one combines T^3-based systems using MFCC and PLP features and the latter one combines a T^3-based system with MFCC features and a Julius-based system with PLP features. Fig. 12 shows the word error rates for the number of iterations of unsupervised speaker adaptation. Baseline conventional batch-mode adaptation and CV adaptation were based on the T^3 system with MFCC features, and CV adaptation was based on a 20-fold CV. The results of the cross-adaptations were derived from the T^3 system with MFCC features as it gave the lowest initial word error rate. The two cross-adaptation systems gave lower error rates than the conventional batch-mode adaptation. Among them, the combination of T^3 with MFCC features and Julius with PLP features gave better results than the combination of T^3 decoders with MFCC and PLP features, which was expected since the two component systems of the former system were more different than the latter. Unsupervised CV adaptation gave slightly better performance than both of these systems. More significantly, unsupervised CV adaptation improved using only a single base system in a systematic manner without requiring two heuristically different base systems, which is advantageous in reducing development cost. The relative reductions in the word error rate by baseline, CV, and cross-adaptation using T^3 with MFCC and PLP features, and cross-adaptation with T^3 and Julius were 11%, 15%, 14%, and 14%, respectively.

VI. CONCLUSION

We proposed unsupervised CV and Ag adaptation algorithms. The CV and Ag adaptation algorithms respectively introduce CV and bagging-like approaches into the conventional iterative unsupervised batch-mode adaptation framework to reinforce generalization and robustness against errors in a recognition hypothesis used for a model parameter update. The proposed algorithms are simple and general and can be combined with various parameter estimation methods. Experiments were conducted with varying conditions using clean and noisy speech recognition tasks. Experimental results showed that both CV and Ag adaptation algorithms gave significantly lower word error rates than the conventional batch-mode adaptation algorithm. Among the proposed algorithms, the CV adaptation algorithm was more advantageous than the Ag one in terms of computational cost. Compared to the conventional batch-mode unsupervised speaker adaptation results, the relative reductions in the word error rate with the proposed algorithms ranged from 4% to 10%, depending on the tasks.

Future work includes using a confidence measure in adaptation [29], improving Ag adaptation algorithm by introducing the ROVER or CNS method, and applying the CV and Ag adaptation frameworks to lightly supervised training [30] and other adaptation problems not limited to speech recognition.

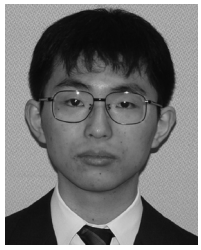
ACKNOWLEDGMENT

The authors would like to thank Asahi Kasei Corp. for letting them use their corpus.

REFERENCES

[1] M. Gales and S. Young, "The application of hidden Markov models in speech recognition," *Foundat. Trends Signal Process.*, vol. 1, no. 3, pp. 195–304, 2008.
[2] S. Nakagawa, T. Watanabe, H. nishizaki, and T. Utsuro, "An unsupervised speaker adaptation method for lecture-style spontaneous speech recognition using multiple recognition systems," *IEICE Trans. Inf. Syst.*, vol. E88-D, no. 3, pp. 463–471, 2005.
[3] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*. London, U.K.: Prentice-Hall, 1982.
[4] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[5] T. Shinozaki, Y. Kubota, and S. Furui, "Unsupervised cross-validation adaptation algorithms for improved adaptation performance," in *Proc. ICASSP*, 2009, pp. 4377–4380.
[6] N. S. Kim and C. K. Un, "Deleted strategy for MMI-based HMM training," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 3, pp. 299–303, May 1998.
[7] 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.
[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] 'Machine Learning Ensemble,' [Online]. Available: http://en.wikipedia.org/wiki/Machine_learning_ensemble Jun. 2009
[10] C. J. Leggetter and P. C. Woodland, "Flexible speaker adaptation using maximum likelihood linear regression," in *Proc. Eurospeech*, 1995, pp. 1155–1158.
[11] J. Gauvain and C. Lee, "Maximum *a posteriori* estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 2, pp. 291–298, Apr. 1994.

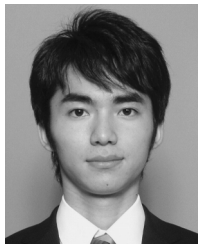
[12] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc.*, no. 1, pp. 1–38, 1977, Series B 39.
[13] H. Soltau, B. Kingsbury, L. Mangu, D. Povery, G. Saon, and G. Zweig, "The IBM 2004 conversational telephony system for rich transcription," in *Proc. ICASSP*, 2005, vol. I, pp. 205–208.
[14] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction rover," in *Proc. IEEE ASRU*, 1997, pp. 347–352.
[15] G. Evermann and P. C. Woodland, "Posterior probability decoding, confidence estimation and system combination," in *Proc. Speech Transcript. Workshop*, 2000.
[16] 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.
[17] K. Itou, M. Yamamoto, K. Takeda, T. Takezawa, T. Matsuoka, T. Kobayashi, K. Shikano, and S. Itahashi, "JNAS: Japanese speech corpus for large vocabulary continuous speech recognition research," *Acoust Soc. Jpn. E*, vol. 20, no. 3, pp. 199–206, 1999.
[18] D. Povey and P. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. ICASSP*, 2002, vol. I, pp. 105–108.
[19] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust. Speech, Signal Process.*, vol. 28, no. 4, pp. 357–366, Aug. 1980.
[20] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *J. Acoust. Soc. Amer.*, vol. 87, no. 4, pp. 1738–1752, 1990.
[21] S. Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP–34, no. 1, pp. 52–59, Feb. 1986.
[22] T. Kato, J. Okamoto, and M. Shozakai, "Physical parameter analysis of Japanese speech at automobile driving act, and analysis of correlation between the parameter and accuracy," in *Autumn Meeting Acoust. Soc. Jpn.* (in Japanese), 2007, vol. 3-Q-25, pp. 267–268.
[23] S. Itahashi, "On recent speech corpora activities in japan," *J. Acoust. Soc. Jpn. (E)*, vol. 20, no. 3, pp. 163–169, 1999.
[24] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-27, no. 2, pp. 113–120, Apr. 1979.
[25] P. R. Dixon, D. A. Caseiro, T. Oonishi, and S. Furui, "The TITECH large vocabulary WFST speech recognition system," in *Proc. IEEE ASRU*, 2007, pp. 443–448.
[26] A. Lee, T. Kawahara, and S. Doshita, "An efficient two-pass search algorithm using word trellis index," in *Proc. ICSLP*, 1998, pp. 1831–1834.
[27] S. Young *et al., The HTK Book*. Cambridge, U.K.: Cambridge Univ. Eng. Dept., 2005.
[28] L. Gillick and S. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proc. ICASSP*, vol. 89, pp. 532–535.
[29] H. Jiang, "Confidence measures for speech recognition: A survey," *Speech Commun.*, vol. 45, no. 4, pp. 455–470, 2005.
[30] L. Lamel, J. Gauvain, and G. Adda, "Investigating lightly supervised acoustic model training," in *Proc. ICASSP*, 2001, vol. 1, pp. 477–480.



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.

Dr. Shinozaki received the Awaya Prize from the Acoustical Society of Japan (ASJ) in 2008 and the Yamashita SIG Research Award from the Information Processing Society of Japan (IPSJ) in 2009.



Yu Kubota received the B.E. degree in computer science from the Tokyo Institute of Technology, Tokyo, Japan, in 2009. He is currently working toward the M.E. degree in the Department of Computer Science, Tokyo Institute of Technology.

Sadaoki Furui (M'79–SM'88–F'93) is currently a Professor in the Department of Computer Science, Tokyo Institute of Technology. He is engaged in a wide range of research on speech analysis, speech recognition, speaker recognition, speech synthesis, and multimodal human–computer interaction and has authored or coauthored over 800 published articles.

Prof. Furui is a Fellow of the International Speech Communication Association (ISCA), the Institute of Electronics, Information, and Communication Engineers of Japan (IEICE), and the Acoustical Society of America. He served as President of the Acoustical Society of Japan (ASJ) and the ISCA. He served as a member of the Board of Governors of the IEEE Signal Processing (SP) Society and Editor-in-Chief of both the Transaction of the IEICE and the *Journal of Speech Communication*. He received the Yonezawa Prize, the Paper Award, and the Achievement Award from IEICE (1975, 1988, 1993, 2003, 2003, 2008), and the Sato Paper Award from ASJ (1985, 1987). He received the Senior Award and Society Award from the IEEE SP Society (1989, 2006), the Achievement Award from the Minister of Science and Technology, and the Minister of Education, Japan (1989, 2006), the Purple Ribbon Medal from the Japanese Emperor (2006), and the ISCA Medal for Scientific Achievement (2009). In 1993, he served as an IEEE SPS Distinguished Lecturer.