Exploitation of Unlabeled Sequences in Hidden Markov Models

Masashi Inoue, *Student Member*, *IEEE*, and Naonori Ueda, *Member*, *IEEE*

Abstract—This paper presents a method for effectively using unlabeled sequential data in the learning of hidden Markov models (HMMs). With the conventional approach, class labels for unlabeled data are assigned *deterministically* by HMMs learned from labeled data. Such labeling often becomes unreliable when the number of labeled data is small. We propose an extended Baum-Welch (EBW) algorithm in which the labeling is undertaken *probabilistically* and *iteratively* so that the labeled and unlabeled data likelihoods are improved. Unlike the conventional approach, the EBW algorithm guarantees convergence to a local maximum of the likelihood. Experimental results on gesture data and speech data show that when labeled training data are scarce, by using unlabeled data, the EBW algorithm improves the classification performance of HMMs more robustly than the conventional naive labeling (NL) approach.

Index Terms—Unlabeled data, sequential data, hidden Markov models, extended Baum-Welch algorithm.

1 Introduction

One major problem in designing classifiers is the scarcity of training data. Usually, a classifier is trained on pairs of observed feature vectors and their class labels. Such a framework is called supervised learning. In most cases, class labels are manually assigned by experts. Therefore, it is expensive and time consuming to collect large amounts of labeled data. Because of this labeling cost, data is often scarce in practice. Consequently, the designed classifier becomes unreliable and its generalization performance becomes poor [1], especially in nonlinear models.

To overcome this problem in supervised learning, a new learning scheme called *semisupervised learning* has been proposed in which unlabeled data are also used to train classifiers. Since unlabeled data can be easily collected without labeling efforts, semisupervised learning has attracted classifier designers and has been studied in various applications for both static data [2], [3], [4], [5], [6] and sequential data [7], [8]. It was reported that the classifiers learned from both labeled and unlabeled data could achieve better classification performance than those learned from small amounts of labeled data.

In this paper, we focus on semisupervised learning for hidden Markov models (HMMs). HMMs are stochastic state transition models that have been extensively used in two types of applications. The first one is concerned with classification of sequences in speech recognition (e.g., [9]), in gesture recognition (e.g., [10]), in computational biology (e.g., [11]), etc. In these tasks, given a sequence, HMMs assign a class label to the entire sequence. The second type deals with the determination of state sequences given observation sequences. Examples of this type of application include part of speech tagging in natural language processing (e.g., [12]) and named entity extraction in information extraction (e.g., [13]). In the second type of application, the term "labeled data" means the observed sequences with the state sequence information associated with them, while the term "unlabeled data" means the sequences without the state sequence information [14], [15], [16]. That is, the second one is concerned with "partially hidden data" which are not "unlabeled data" in the sense used for the semisupervised learning for static data. Such "partially hidden data" of the second application type can be processed by the standard learning framework of HMMs and, in this paper, we investigate the "unlabeled data" in the first type of application, the classification of sequences.

For the first applications, a simple semisupervised learning scheme has been used, which we refer to as the naive labeling (NL) approach [7], [8]. With the NL approach, HMMs are first trained solely on given labeled data. Then, pseudoclass labels are *deterministically* assigned to unlabeled data by classifying them using the trained HMMs. The HMMs are retrained with these newly labeled data.

The NL approach appears to be a method for classifier adaptation under the assumption that the initial model is to some extent reliable. In the above two studies where the quantities of initial labeled data were relatively large, the NL approach could improve the HMMs. This applies to the case when training HMMs used in speech recognition systems for adaptation. However, when trained on small amounts of labeled data, initial models become unreliable and, therefore, the pseudolabels also become unreliable. As a result, the addition of unlabeled data with such unreliable labels may not improve the generalization performance of the HMM.

To overcome this problem associated with the NL approach, in this paper, we present a new semisupervised learning approach that can use unlabeled data for training HMMs more effectively than the NL approach. In our

• *M. Inoue is with the Graduate School of Information Science, Nara Institute of Science and Technology (NAIST), 8916-5 Takayama-cho, Ikoma-shi, Nara, Japan. E-mail: masash-i@is.aist-nara.ac.jp.*
• *N. Ueda is with NTT Communication Science Laboratories, 2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan. E-mail: ueda@cslab.kecl.ntt.co.jp.*

Manuscript received 30 Sept. 2002; revised 11 June 2003; accepted 24 June 2003.
Recommended for acceptance by D. Forsyth.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 117477.

approach, as with [3], [4], [5], [6], the class labels are treated as missing information and the pseudoclass labels are *probabilistically* assigned to unlabeled data so that the joint likelihood function for both labeled and unlabeled data is maximized. To handle unlabeled data, we introduce extended tied-mixture HMMs (ETM-HMMs) as a mixture of tied-mixture HMMs (TM-HMMs) [17], [18]. For training ETM-HMMs, we derive an extended Baum-Welch (EBW) algorithm. Unlike the NL approach, the proposed algorithm theoretically guarantees convergence to a local maximum of the likelihood.

The EBW algorithm can be regarded as an extension of the conventional labeling approach for static data based on the EM algorithm [19] to the one for sequential data. Although the usefulness of static unlabeled data has been claimed, the usefulness of sequential unlabeled data in such approach has not been shown. We presented an outline of our approach in a previous paper [20]. In the present paper, we formally explain the EBW algorithm and empirically compare it with the NL approach.

The rest of the paper is organized as follows: After the formal definition of labeled and unlabeled data in Section 2, the conventional NL approach is explained in Section 3. Section 4 briefly reviews TM-HMMs and introduces ETM-HMMs. Next, the EBW algorithm is presented in Section 5. Section 6 provides some experimental results using gesture and speech data in which the effect of unlabeled data in our method is evaluated and compared with the NL approach. Section 7 concludes the paper.

2 Labeled and Unlabeled Data

Let $X_n = \langle \mathbf{x}_{n_1}, \mathbf{x}_{n_2}, \dots, \mathbf{x}_{n_t}, \dots, \mathbf{x}_{n_{T_n}} \rangle$ be the nth observation sequence of d-dimensional feature vectors, where $\mathbf{x}_{n_t} \in \mathcal{R}^d$ is the tth feature vector in X_n and T_n is the length of the sequence X_n. Let y_n be a class label corresponding to X_n. $y_n \in \{1, \dots, y, \dots, Y\}$, where Y is the number of classes. Thus, a labeled datum is (X_n, y_n) and an unlabeled datum is X_n. Let $\mathcal{D}_l$ be a labeled data set and $\mathcal{D}_u$ be an unlabeled data set. It is assumed that we have $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$. In addition, we assume that data are mutually independent.

In the above definition, a single label is assigned to each of the observed sequences. Many types of sequence meet the above definition. For example, physiological sequences such as brain waves, biological sequences such as gene expression profiles, and economic time series data such as the trend of unemployment. For some types of sequence, on the other hand, another definition of labeled sequences is sometimes used: An observed sequence corresponds to several concatenated labels. For example, continuous speech recognition systems that regard each phoneme as a class deal with spoken sentences as such sequences. For these settings, however, together with segmentation algorithms such as [21], the concatenated sequences may be dealt with as the basic sequences defined above. In this paper, with the aim of evaluating the proposed algorithm, we consider only the basic sequences where there is a one-to-one correspondence between a sequence of feature vectors and a class label.

3 Naive Labeling Approach

First, we review the conventional NL approach that utilizes unlabeled data straightforwardly. Assume that relatively small amounts of data have been manually labeled and vast amounts of unlabeled data are accessible. In the NL approach, using the hand-labeled data, a partially correct initial model is trained. The remaining unlabeled data are labeled based on the initial model. Once unlabeled data have been given pseudolabels, they can be regarded as labeled data. Then, the model can be retrained by using conventional supervised learning algorithms.

Let $\mathcal{D}'_l$ be a pseudolabeled data set whose labels are generated by the initial model. Then, the NL approach can be summarized as follows:

```
Step 1: Initialization
  1-1. Set $\mathcal{D} \leftarrow \mathcal{D}_l$.
  1-2. Train a model (classifier) using $\mathcal{D}$.

Step 2: Retraining
  Repeat the following several times:
  2-1. Based on the current model, assign a pseudolabel to
       each datum in $\mathcal{D}_u$ and generate $\mathcal{D}'_l$.
  2-2. Set $\mathcal{D} \leftarrow \mathcal{D}_l \cup \mathcal{D}'_l$.
  2-3. Retrain the model using $\mathcal{D}$.
```

The above algorithm gives the most general form of the NL approach. However, since the NL approach has been developed independently for various applications, some variants and extensions exist. For example, in [2], $\mathcal{D}'_l$ was used and $\mathcal{D}_l$ was not used in retraining, and in [7], Step 2 was executed just once. However, such differences do not seem to be essential. Therefore, in this paper, we use the general algorithm given above.

Although the NL approach has been reported to be effective in practice, it has two fundamental drawbacks. First, the convergence of Step 2 is not guaranteed. Therefore, should the retraining procedure not converge, we should stop the retraining cycle based on some heuristic criterion such as the maximum number of retraining cycles. Second, when the initial model is unreliable, the unlabeled data cannot be effectively used. Since $\mathcal{D}_l$ in Step 1-1 is often small, the initial model may be poorly trained; thus, a substantial percentage of pseudolabels assigned by such models may be wrong. If $\mathcal{D}'_l$ contains many erroneous data, their addition may deteriorate the performance of the classifier.

Confidence measures for labeling have been introduced to cope with the second problem [7], [8] so that unreliable pseudolabeled data whose confidence measures are below a certain threshold are not included in $\mathcal{D}$. These confidence measures are defined for individual applications based on domain knowledge and have been reported to be beneficial in improving classification. Such measures are, however, not always available or effective. That is, the success of the NL approach basically depends on the quality of the initial model.

4 TM-HMMs AND ETM-HMMs

4.1 Proposed Algorithm and Model Structure

To overcome the problems associated with the NL approach, we propose a new algorithm which uses unlabeled data directly without explicit labeling. By so doing, we can use both labeled and unlabeled data simultaneously and expect a better initial estimate of the model parameters based on the larger amount of training data. Such methods have been already presented for static models [3], [4], [5], [6]. However, the structure of conventional HMMs prevents the direct use of these methods. The static models used in the above researchs include all classes in a single model and unlabeled data can be used in those models. In contrast, HMMs are constructed for each class and all training data must be allocated to the classes before learning. Therefore, unlabeled data cannot be used unless pseudolabels are given by a method such as the NL approach. In Section 4.2, to clarify why unlabeled data cannot be used in conventional HMMs, we detail the model structure of conventional HMMs, especially tied-mixture HMMs (TM-HMMs). In Section 4.3, as an extension of TM-HMMs, we introduce a model structure named extended tied-mixture HMMs (ETM-HMMs) that can handle unlabeled data.

4.2 TM-HMMs

An HMM consists of several states and the probabilistic transitions between them. With continuous HMMs, each HMM state outputs a continuous (vector) value according to the distribution of a mixture of Gaussians. In a TM-HMM shown in Fig. 1a, each state has a mixture of Gaussians with shared underlying Gaussian components over all classes, but different mixing parameters. TM-HMMs are frequently used because they can reduce the number of model parameters without losing flexibility [17], [18].

Let TM-HMM(y) be a TM-HMM of class y. Let U^y be the number of states in TM-HMM(y) and K be the number of Gaussian components in the feature space. Let $s_t \in \{1,\dots,i,\dots,j,\dots,U^y\}$ be the index of the state at time t.[1] Let $m_t \in \{1,\dots,k,\dots,K\}$ be the index of the component at time t. Let $\Theta_y = \{\pi_i^y, a_{ij}^y, c_{jk}^y, \boldsymbol{\mu}_k, \Sigma_k\}$ be the set of parameters for TM-HMM(y). The definitions of parameters in Θ_y are listed below.

- Initial state probabilities for $1 \le s_1 \le U^y$:
$$\pi_i^y = P(s_1 = i \mid y), \tag{1}$$
where $\pi_i^y \ge 0$ and $\sum_i \pi_i^y = 1$.
- Transition probabilities for $1 \le s_t, s_{t+1} \le U^y$:
$$a_{ij}^y = P(s_{t+1} = j \mid s_t = i, y), \tag{2}$$
where $a_{ij}^y \ge 0$ and $\sum_j a_{ij}^y = 1$.
- Mixture coefficients for $1 \le s_t \le U^y,\ 1 \le m_t \le K$:
$$c_{jk}^y = P(m_t = k \mid s_t = j, y), \tag{3}$$
where $c_{jk}^y \ge 0$ and $\sum_k c_{jk}^y = 1$.

1. Since indices i and j represent a state of the HMM for a particular class (y), they should be written as i^y and j^y. However, for simplicity of notation, we omit the superscript y.

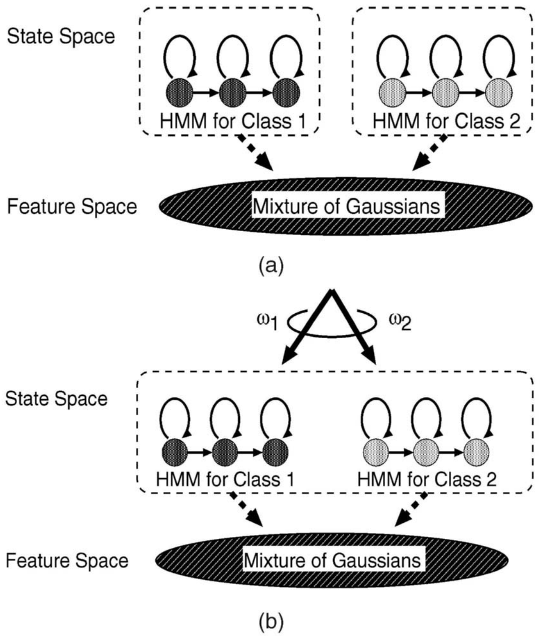

Fig. 1. This figure shows a two-class example of (a) a Tied-Mixture HMM (TM-HMM) and (b) an Extended Tied-Mixture HMM (ETM-HMM). The circles represent HMM states and the solid arrows represent state transitions. The black ovals denote feature spaces represented by a mixture of Gaussians. In (b), ω_1 and ω_2 denote class priors for class 1 and class 2, respectively.

- Gaussian parameters (mean vectors and covariance matrices) for $1 \le m_t \le K$:
$$\boldsymbol{\mu}_k \text{ and } \Sigma_k, \tag{4}$$
where $m_t = k$.

Note that, since all Gaussians are common to all states and classes, $\boldsymbol{\mu}_k$ and Σ_k depend neither on i, j nor on y.

Let $S_n = \{s_t \mid t = 1,\dots,T_n\}$ be the sequence of states, $M_n = \{m_t \mid t = 1,\dots,T_n\}$ be the sequence of Gaussian components both of which correspond to X_n. In TM-HMM(y), X_n is observable and S_n and M_n are unobservable; hence, they are called *hidden* variables. According to the definition of Θ_y, when $s_1 = h$, $s_t = i$, $s_{t+1} = j$, $m_t = k$, the complete data likelihood of TM-HMM(y) is given by:

$$p(X_n, S_n, M_n \mid \Theta_y) = \pi_h^y \prod_{t=1}^{T_n-1} a_{ij}^y \prod_{t=1}^{T_n} c_{ik}^y \mathcal{N}(\mathbf{x}_{n_t} \mid \boldsymbol{\mu}_k, \Sigma_k). \tag{5}$$

In TM-HMMs, as shown in Fig. 2a, the feature space is tied over classes and any feature vector can be placed there. In contrast, since state spaces are defined separately for each TM-HMM(y), unlabeled data without class labels cannot be placed in state spaces. Therefore, TM-HMMs cannot use unlabeled data directly.

4.3 ETM-HMMs

To deal with unlabeled data in state space, we use another model structure named an ETM-HMM. Let ω_y be a class prior. Then, an ETM-HMM can be defined by:

$$\mathrm{ETM-HMM} = \sum_{y=1}^{Y} \omega_y \mathrm{TM-HMM}(y). \tag{6}$$

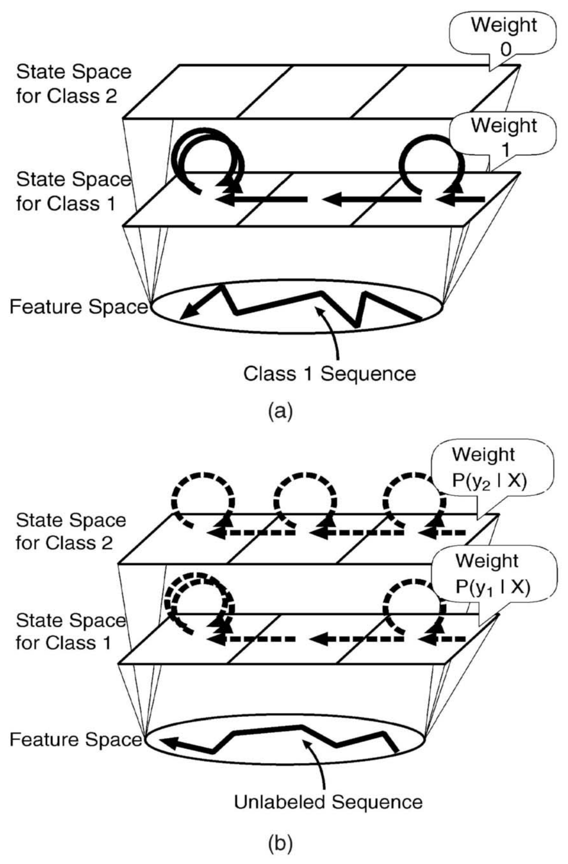

Fig. 2. Allocations of (a) labeled and (b) unlabeled sequences in the state spaces of two classes. Labeled sequences can be located in either TM-HMMs or an ETM-HMM. By contrast, unlabeled sequences can be located only in an ETM-HMM.

That is, an ETM-HMM is defined as a mixture of TM-HMMs of different classes as shown in Fig. 1b. In an ETM-HMM, as well as a TM-HMM, a feature space represented by a mixture of Gaussians is tied over all classes. As shown in Fig. 2b, in ETM-HMMs, unlabeled data can be located in multiple state spaces with probabilistic weights.

The hidden variables of an ETM-HMM for labeled data are that for a TM-HMM, and for unlabeled data, class label y_n is also a hidden variable. A set of parameters for an ETM-HMM $\Theta = \{\omega_y, \Theta_y \mid y = 1, \dots, Y\}$, where Θ_y has been defined for TM-HMMs in Section 4.2. When $s_1 = h$, $s_t = i$, $s_{t+1} = j$, and $m_t = k$, the complete data likelihood for an ETM-HMM is given by:

$$p(X_n, y_n, S_n, M_n \mid \Theta) = \omega_y \pi_h^y \prod_{t=1}^{T_n-1} a_{ij}^y \prod_{t=1}^{T_n} c_{ik}^y \mathcal{N}(\mathbf{x}_{n_t} \mid \boldsymbol{\mu}_k, \Sigma_k). \tag{7}$$

Equation (7) differs from (5) in that class prior ω_y is introduced and class labels are regarded as random variables.

5 Extended Baum-Welch Algorithm

5.1 Q-Function for Mixed Data

This section describes the learning algorithm for the ETM-HMMs, named the extended Baum-Welch (EBW) algorithm. The EBW algorithm is an extension of the Baum-Welch (BW) algorithm [22], which is widely used to train HMMs. The BW algorithm can be regarded as an application of the expectation-maximization (EM) algorithm [19] to HMMs. The EM algorithm is an iterative procedure for computing maximum likelihood estimates from incomplete data. It alternates two steps: The Estep computes the expected complete data log-likelihood called Q-function and the M-step maximizes the Q-function with respect to unknown parameters. Since the ETM-HMM learns from both labeled and unlabeled data, the Q-function for the conventional HMM needs to be redefined.

First, we derive the Q-function for labeled and unlabeled mixed data in a general form. Let $\mathcal{Z}_l$ and $\mathcal{Z}_u$ be sets of hidden variables that correspond to $\mathcal{D}_l$ and $\mathcal{D}_u$, respectively. Then, a set of hidden variables $\mathcal{Z} = \mathcal{Z}_l \cup \mathcal{Z}_u$ corresponds to $\mathcal{D}$. Let θ be a set of unknown model parameters. Assuming that data are independently and identically distributed (i.i.d.), complete data likelihood can be decomposed into the complete data likelihood for labeled data and that for unlabeled data:

$$p(\mathcal{D}, \mathcal{Z} \mid \theta) = p(\mathcal{D}_l, \mathcal{Z}_l \mid \theta) \cdot p(\mathcal{D}_u, \mathcal{Z}_u \mid \theta). \tag{8}$$

Similarly, since $p(\mathcal{D}) = p(\mathcal{D}_l) \cdot p(\mathcal{D}_u)$ holds based on the assumption of the independence between data, the distribution of posterior probabilities for the hidden variable $\mathcal{Z}$ given current parameter estimates θ^{old} can be decomposed as shown below:

$$\begin{aligned} P(\mathcal{Z} \mid \mathcal{D}, \theta^{\text{old}}) &= \frac{p(\mathcal{D}, \mathcal{Z} \mid \theta^{\text{old}})}{p(\mathcal{D})} \\ &= \frac{p(\mathcal{D}_l, \mathcal{Z}_l \mid \theta^{\text{old}})}{p(\mathcal{D}_l)} \cdot \frac{p(\mathcal{D}_u, \mathcal{Z}_u \mid \theta^{\text{old}})}{p(\mathcal{D}_u)} \\ &\equiv P(\mathcal{Z}_l \mid \mathcal{D}_l, \theta^{\text{old}}) \cdot P(\mathcal{Z}_u \mid \mathcal{D}_u, \theta^{\text{old}}). \end{aligned} \tag{9}$$

By definition, the general formulation of the Q-function is given by:

$$\begin{aligned} Q(\theta \mid \theta^{\text{old}}) &= \mathrm{E}\Big[\log p(\mathcal{D}, \mathcal{Z} \mid \theta) \mid \mathcal{D}, \theta^{\text{old}}\Big] \\ &= \sum_{\mathcal{Z}} P(\mathcal{Z} \mid \mathcal{D}, \theta^{\text{old}}) \log p(\mathcal{D}, \mathcal{Z} \mid \theta). \end{aligned} \tag{10}$$

Therefore, by substituting (8) and (9) into (10), the Q-function for the mixed data can be obtained:

$$Q(\theta \mid \theta^{\text{old}}) = Q_l(\theta \mid \theta^{\text{old}}) + Q_u(\theta \mid \theta^{\text{old}}), \tag{11}$$

where

$$\begin{aligned} Q_l(\theta \mid \theta^{\text{old}}) &= \mathrm{E}\Big[\log p(\mathcal{D}_l, \mathcal{Z}_l \mid \theta) \mid \mathcal{D}_l, \theta^{\text{old}}\Big] \\ Q_u(\theta \mid \theta^{\text{old}}) &= \mathrm{E}\Big[\log p(\mathcal{D}_u, \mathcal{Z}_u \mid \theta) \mid \mathcal{D}_u, \theta^{\text{old}}\Big]. \end{aligned}$$

Therefore, the Q-function for the mixed data is the direct sum of the Q-functions for labeled and unlabeled data.

5.2 E-Step: Calculation of Q-Function

Applying (11) to ETM-HMMs, we derive the Q-function for the EBW algorithm. Let N_l be the number of labeled sequential data and $\mathcal{I}_y$ be the set of data indices $\{n \mid y_n = y\}$. Labeled data are $\{(X_n, y_n) \in \mathcal{D}_l\}$ and the corresponding hidden variables are $\{(S_n, M_n) \in \mathcal{Z}_l\}$. By taking the conditional expectation of the complete log-likelihood (log of (7)) over the hidden variables given the data and the current estimates of parameters Θ^{old}, the Q-function for labeled data, Q_l, is derived as follows:

$$\begin{aligned}
&Q_l(\Theta \mid \Theta^{\text{old}})\\
&=\sum_{n=1}^{N_l} \mathrm{E}\Big[\log p(X_n, y_n, S_n, M_n \mid \Theta) \mid X_n, y_n, \Theta^{\text{old}}\Big]\\
&=\sum_{y=1}^{Y}\sum_{n\in\mathcal{I}_y}\Bigg\{\log\omega_y\\
&+\sum_j \gamma_{n_0}(j)\log\pi_j^y\\
&+\sum_{i,j}\sum_{t=1}^{T_n-1}\gamma_{n_t}(i,j)\log a_{ij}^y\\
&+\sum_{j,k}\sum_{t=1}^{T_n}\zeta_{n_t}(j,k)\log c_{ik}^y\\
&+\sum_{k}\sum_{t=1}^{T_n}\kappa_{n_t}(k)\log\mathcal{N}(\mathbf{x}_{n_t} \mid \boldsymbol{\mu}_k, \Sigma_k)\Bigg\}.
\end{aligned} \tag{12}$$

Next, the posterior probabilities of hidden variables defined as (9) are specified. In (12), γ, ζ, and κ represent transition posteriors, staying and emission posteriors, and emission posteriors given below for $t \geq 1$:

$$\gamma_{n_t}(i,j) = P(s_t = i, s_{t+1} = j \mid X_n, y_n, \Theta^{\text{old}}), \tag{13}$$

$$\zeta_{n_t}(j,k) = P(s_t = j, m_t = k \mid X_n, y_n, \Theta^{\text{old}}), \tag{14}$$

$$\kappa_{n_t}(k) = P(m_t = k \mid X_n, y_n, \Theta^{\text{old}}). \tag{15}$$

Note that, in (13), $\gamma_{n_0}(j) = P(s_1 = j \mid X_n, y_n, \Theta^{\text{old}})$.

Let N_u be the number of unlabeled sequential data. Unlabeled data are $\{X_n \in \mathcal{D}_u\}$ and the corresponding hidden variables are $\{(y_n, S_n, M_n) \in \mathcal{Z}_u\}$. By taking the conditional expectation of the complete log-likelihood (log of (7)) over the hidden variables given the data and the current estimates of parameters Θ^{old}, the Q-function for unlabeled data, Q_u, is derived as follows:

$$\begin{aligned}
&Q_u(\Theta \mid \Theta^{\text{old}})\\
&=\sum_{n=1}^{N_u} \mathrm{E}\Big[\log p(X_n, y_n, S_n, M_n \mid \Theta) \mid X_n, \Theta^{\text{old}}\Big]\\
&=\sum_{n=1}^{N_u}\sum_{y=1}^{Y}\Bigg\{P(y \mid X_n, \Theta^{\text{old}})\log\omega_y\\
&+\sum_j \lambda_{n_0}(y,j)\log\pi_j^y\\
&+\sum_{i,j}\sum_{t=1}^{T_n-1}\lambda_{n_t}(y,i,j)\log a_{ij}^y\\
&+\sum_{j,k}\sum_{t=1}^{T_n}\eta_{n_t}(y,j,k)\log c_{ik}^y\\
&+\sum_{k}\sum_{t=1}^{T_n}\xi_{n_t}(y,k)\log\mathcal{N}(\mathbf{x}_{n_t} \mid \boldsymbol{\mu}_k, \Sigma_k)\Bigg\}.
\end{aligned} \tag{16}$$

In (16), λ, η, and ξ represent transition posteriors, staying and emission posteriors, and emission posteriors given below for $t \geq 1$:

$$\lambda_{n_t}(y,i,j) = P(y_n = y, s_t = i, s_{t+1} = j \mid X_n, \Theta^{\text{old}}), \tag{17}$$

$$\eta_{n_t}(y,j,k) = P(y_n = y, s_t = j, m_t = k \mid X_n, \Theta^{\text{old}}), \tag{18}$$

$$\xi_{n_t}(y,k) = P(y_n = y, m_t = k \mid X_n, \Theta^{\text{old}}). \tag{19}$$

Note that, in (17), $\lambda_{n_0}(y,j) = P(y_n = y, s_1 = j \mid X_n, \Theta^{\text{old}})$. Equations (17)-(19) correspond to (13)-(15), respectively, but differ in that the class label y is regarded as the value of the random variable. The class posterior given below should also be calculated.

$$P(y \mid X_n, \Theta^{\text{old}}). \tag{20}$$

Either for labeled data or for unlabeled data, the forward-backward algorithm [23] efficiently calculates the above posteriors. For unlabeled data, however, due to computational problems, a modified scaling technique needs to be applied in practice (See Appendix A).

The Q-function of the ETM-HMM is given by the sum of (12) and (16). It is different from that of the TM-HMM in the following two respects: First, Q_u does not exist in the Q-function for TM-HMMs since TM-HMMs cannot handle unlabeled data. Second, the term for ω_y does not exist in Q_l for TM-HMMs in which class priors are not taken into account.

5.3 M-Step: Parameter Reestimation

In the M-step, the Q-function that was derived in Section 5.2 is maximized with respect to each model parameter. For example, the reestimation formula for class prior ω_y can be obtained by maximizing the objective function $J = Q(\Theta \mid \Theta^{\text{old}}) + \tau(\sum_{y=1}^{Y}\omega_y - 1)$ with the constraint $\sum_{y=1}^{Y}\omega_y = 1$, where τ is a Lagrange multiplier. By solving the two equations, $\partial J/\partial\omega_y = 0$ and $\partial J/\partial\tau = 0$, the following reestimation formula is obtained:

$$\hat{\omega}_y = \frac{N_y + \sum_{n=1}^{N_u} P(y \mid X_n, \Theta^{\text{old}})}{N_l + N_u}, \tag{21}$$

where $\hat{\omega}_y$ denotes newly estimated ω_y and N_y represents the number of labeled data belonging to class y.

In a similar manner, the reestimation formulae for transition probabilities and mixture coefficients are obtained as follows:[2]

$$\hat{a}_{ij}^y = \frac{\sum_{n\in\mathcal{I}_y}\sum_{t=1}^{T_n-1}\gamma_{n_t}^y(i,j) + \sum_{n=1}^{N_u}\sum_{t=1}^{T_n-1}\lambda_{n_t}(y,i,j)}{\sum_{n\in\mathcal{I}_y}\sum_{t=1}^{T_n-1}\sum_j\gamma_{n_t}^y(i,j) + \sum_{n=1}^{N_u}\sum_{t=1}^{T_n-1}\sum_j\lambda_{n_t}(y,i,j)}, \tag{22}$$

2. The reestimation formula for π_i^y is not given here because we used the left-to-right models in which π_i^y is unchanged by reestimation.

$$\hat{c}^y_{jk} = \frac{\sum_{n\in\mathcal{I}_y}\sum_{t=1}^{T_n}\zeta^y_{n_t}(j,k) + \sum_{n=1}^{N_u}\sum_{t=1}^{T_n}\eta_{n_t}(y,j,k)}{\sum_{n\in\mathcal{I}_y}\sum_{t=1}^{T_n}\sum_k\zeta^y_{n_t}(j,k) + \sum_{n=1}^{N_u}\sum_{t=1}^{T_n}\sum_k\eta_{n_t}(y,j,k)}. \quad (23)$$

The reestimation formulae for mixture components, which are Gaussians, are given as follows:

$$\hat{\boldsymbol{\mu}}_k = \frac{\sum_{y=1}^{Y}\left\{\sum_{n\in\mathcal{I}_y}\sum_{t=1}^{T_n}\kappa^y_{n_t}(k)\mathbf{x}_{n_t} + \sum_{n=1}^{N_u}\sum_{t=1}^{T_n}\xi_{n_t}(y,k)\mathbf{x}_{n_t}\right\}}{\sum_{y=1}^{Y}\left\{\sum_{n\in\mathcal{I}_y}\sum_{t=1}^{T_n}\kappa^y_{n_t}(k) + \sum_{n=1}^{N_u}\sum_{t=1}^{T_n}\xi_{n_t}(y,k)\right\}}, \quad (24)$$

$$\hat{\Sigma}_k = \frac{\sum_{y=1}^{Y}\left\{\sum_{n\in\mathcal{I}_y}\sum_{t=1}^{T_n}\kappa^y_{n_t}(k)\mathbf{v}_{kt} + \sum_{n=1}^{N_u}\sum_{t=1}^{T_n}\xi_{n_t}(y,k)\mathbf{v}_{kt}\right\}}{\sum_{y=1}^{Y}\left\{\sum_{n\in\mathcal{I}_y}\sum_{t=1}^{T_n}\kappa^y_{n_t}(k) + \sum_{n=1}^{N_u}\sum_{t=1}^{T_n}\xi_{n_t}(y,k)\right\}}, \quad (25)$$

where $\mathbf{v}_{kt} = (\mathbf{x}_{n_t} - \boldsymbol{\mu}_k)(\mathbf{x}_{n_t} - \boldsymbol{\mu}_k)^t$ (the superscript t denotes transpose).

At each stage of parameter reestimation, the increase in the likelihood is guaranteed. We stop the EM cycle when the change in the log-likelihood value is below a specified threshold. However, when mixed data are used, accurate calculation of the log-likelihood is computationally difficult; therefore, we use another measure as a convergence criterion (See Appendix B).

When only labeled data are used, the reestimation formula for the class priors becomes $\hat{\omega}_y = N_y/N_l$. It is a constant defined by the number of training data for each class. Other parameter reestimation formulae become those constituted from the first term of both numerators and denominators. Such formulae are the same as those in the BW algorithm. Therefore, if the class priors are the same among classes, in other words, if we assume all classes have the same number of data, the EBW algorithm without unlabeled data is reduced to the BW algorithm. That is, the BW algorithm can be viewed as a special case of the EBW algorithm.

5.4 Selective Posterior Calculation

As for the practical issue, a few remarks should be made concerning the computational cost of the EBW algorithm. The computational complexity of calculating posteriors in the EBW algorithm is, for labeled data, $O(N_l)$; while, for unlabeled data, it is $O(N_u \cdot Y)$. In the reestimation formula for c^y_{ij} (23), for example, the posterior (18) must be calculated for all combinations of classes, states, and Gaussian components (Y, U^y, and K) for each sequence. In the case of the experiment in Section 6.3, $Y = 48$, $U^y = 3$, and $K = 500$; thus, the number of combinations is $72,000$. Since such a large amount of computation is sometimes impractical, we introduce the following approximate calculation, which we call selective posterior calculation.

First, class posteriors (20) are calculated for all classes. Next, according to their values, the classes are sorted in descending order (e.g., If $Y = 4$, $P(1 \mid X) = 0.3$, $P(2 \mid X) = 0.7$, $P(3 \mid X) = 0.1$, and $P(4 \mid X) = 0.5$, we have an ordered class index set as $\{2, 4, 1, 3\}$). Posteriors (13)-(19) are calculated only for the top $M(\ll Y)$ classes and the posterior values for the remaining classes are set at zero. For instance, if we set $M = 3$, there are $4,500$ parameter combinations for (18), which is $1/16$ of the original number of combinations.

5.5 Classification

Once the ETM-HMM has been trained based on the maximum likelihood principle, unknown sequential data are classified to the class with the largest posterior probability. The class y^* of an unseen sequence X^* is determined by the following formula:

$$y^* = \arg\max_y P(y \mid X^*, \hat{\Theta}), \quad (26)$$

where $\hat{\Theta}$ is the estimate of a set of parameters obtained by the EBW algorithm.

6 Experiments

6.1 Experimental Conditions

We experimentally validated the proposed algorithm on two data sets: gesture data and speech data. Our goal is to improve the classifiers that learned poorly due to the scarcity of labeled data by adding unlabeled data. The classification error rate (CER) was used to evaluate the performance of the learned classifiers. The data in the original data sets were all labeled; thus, unlabeled data were created by hiding their class labels for experimental purposes. In both experiments, in addition to the few initial labeled training data $\mathcal{D}_l^{\text{ini}}$, either labeled data $\mathcal{D}_l$ or unlabeled data $\mathcal{D}_u$ were added to the training data set. Here, we say that the initial data are "few" when the addition of *labeled* data to the initial training data set decreases the CER on the test data set. This situation implies that the initial labeled training data are insufficient relative to the number of model parameters and the model parameters are not reliably estimated.

Once we found that there were few initial labeled data, such ETM-HMM was trained on the larger quantity of labeled data ($\mathcal{D}_l^{\text{ini}} \cup \mathcal{D}_l$) or on the mixed data ($\mathcal{D}_l^{\text{ini}} \cup \mathcal{D}_u$) by using the EBW algorithm varying the quantity of additional data. The classification performance of learned ETM-HMMs was compared for two types of additional data, labeled or unlabeled, with respect to their quantity. In general, as the quantity of labeled training data increases, the generalization performance improves [24]. Therefore, the addition of labeled data can be regarded as the ideal setting for performance improvement; we can examine how close the performance with the addition of unlabeled data is to the performance with the addition of labeled data.

In addition to the quantity of training data, the classification performance is influenced by two other factors: variances in the initialization of the model parameters and those in the training data selection. Although both kinds of variance should be averaged, we only averaged the effect of data selection over 10 trials and used fixed initial model parameters for all trials since the amount of computation required was too large. For each quantity of additional labeled and unlabeled data, the training data were drawn 10 times randomly from the whole available training data set. Then, 10 different ETM-HMMs were trained on these 10 data subsets. The median of their CERs for the independent test data set was calculated.

Throughout the experiments in this article, we used left-to-right HMMs. Fixed model parameters of those HMMs were set as follows: Class priors, transition probabilities, and mixture coefficients were initialized to uniform distributions. Gaussians means were determined by the k-means algorithm for the whole available training data of all classes. Gaussian covariances were determined by the Voronoi partitions of the data based on the result of the k-means algorithm. The covariance matrices were diagonal. Note that, in ETM-HMMs, since the feature spaces are tied, all data can be used for estimating the Gaussian parameters. By undertaking the initialization with large quantities of data, we avoided the effect of poor parameterization so that we could focus our attention on the effect of the data amount.

6.2 Gesture Classification

6.2.1 Sign Language Data Set

The first experiment was on gesture data. Each gesture was one of the 15 Japanese sign language (JSL) signs. With magnetic sensors attached to both hands of the experimental subjects, the positions of the hands in three dimensional space and the rotation angles around three axes were measured at a 30 Hz sampling rate. The collected sequences were, therefore, 15 classes ($Y = 15$) and 12-dimensional. Each sign was performed 40 times (30 for training data and 10 for test data) by 20 nonnative JSL signers. The total amount of training data was $9,000$ and the total amount of test data was $3,000$. All 15 classes have the same amount of data (600 training data and 200 test data for each class). The mean, maximum, and minimum lengths of the sequences were, respectively, 25.6, 44, and 15 for the training data and 24.6, 41, and 16 for the test data.

6.2.2 Preliminary Experiment

Unless the addition of labeled data reduces the CER, unlabeled data cannot reduce the errors, either. Therefore, in our preliminary experiments, we first searched for a situation where the training data were insufficient. Let $N_y^{\rm ini}$ be the amount of initial labeled data for class y. We found that, when $N_y^{\rm ini} = 2$ for all classes, the number of states $U^y = 5$ for all classes and the number of components $K = 50$, the median of CERs decreased more than 40 points when labeled sequences were added to the training data. This implies that $N_y^{\rm ini}$ is too small relative to the number of model parameters to be estimated.

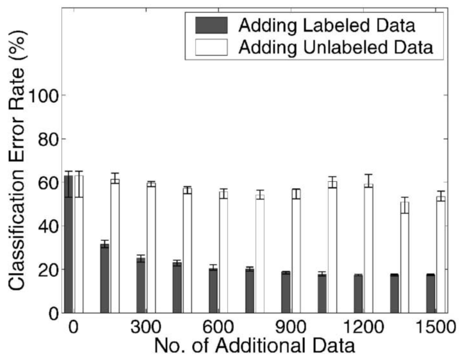

Fig. 3. The change in the classification error rates for JSL data by the number of additional labeled or unlabeled data. The thick bars represent medians of CERs and the thin lines represent upper and lower quartiles. The initial training data were two labeled data for each class (i.e., $N_l^{\rm ini} = 30$), the number of states $U^y = 5$ for each class, and the number of Gaussians $K = 50$. Either 150 labeled or unlabeled data were added at a time.

Using the above case as an example, we evaluated the EBW algorithm when unlabeled data were added.

6.2.3 Experimental Results

We trained ETM-HMMs of the structure specified by $U^y = 5$ and $K = 50$. The initial labeled data, $N_y^{\rm ini} = 2$ for all ys, comprised about 0.33 percent of the total available training data. Either 150 labeled or unlabeled sequences were added at a time to $\mathcal{D}_l^{\rm ini}$. That is, the number of additional data, N_l or N_u, was 150. For each number, N_l or N_u, we created 10 training data sets by random sampling and, for the 10 ETM-HMMs learned from those training data subsets, the CERs on the test data with a size $N_t = 3,000$ were computed.

The result of the experiment is shown in Fig. 3. Each bar in the graph represents the median of the CERs of 10 ETM-HMMs. When no data were added, the median of the CERs was 63.1 percent as shown by the leftmost bars. Black bars show the change in the CERs caused by the addition of labeled data. The median of CERs decreased to 17.2 percent at their lowest. White bars show the change in the CERs caused by the addition of unlabeled data. The median of CERs decreased to 50.8 percent at their lowest. As Gaussian parameters change more than other parameters by adding unlabeled data, we presume that the improved estimation of Gaussian parameters is the most important source of performance improvement.

It should be noted that the addition of labeled data lowered the CERs dramatically, whereas the addition of unlabeled data lowered the CERs gradually. That is, in terms of reducing errors, the labeled data were clearly superior to the unlabeled data. However, we do not usually have additional expensive labeled data and without adding unlabeled data, the median of CERs remains at 63.1 percent. In this regard, we can say that the addition of unlabeled data by the EBW algorithm was beneficial in improving the classifier for this gesture data set when the amount of labeled data was limited.

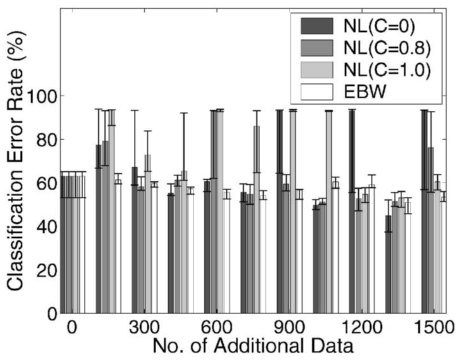

Fig. 4. The change in the classification error rates for JSL data by the EBW algorithm and by the NL approach for the same amount of unlabeled data. The initial training data were two labeled data for each class (i.e., $N_l^{\text{ini}} = 30$), the number of states $U^y = 5$ for each class, and the number of Gaussians $K = 50$. Either 150 labeled or unlabeled data were added at a time. The thick bars represent medians of CERs and the thin lines represent upper and lower quartiles. In the NL approach, the threshold of the confidence measure was changed ($C = 0, 0.8, 1.0$).

Note that, since all classes were equal as regards the number of initial labeled data, the class priors were uniform. Therefore, the initial ETM-HMMs learned by the EBW algorithm were the same as the initial TM-HMMs learned by the BW algorithm.

6.2.4 Comparison with Naive Labeling Approach

We compare our EBW algorithm with the NL approach explained in Section 3 in terms of the degree of improvement. Varying the confidence threshold C among $\{0, 0.8, 1.0\}$, we computed the changes in the CERs caused by the NL approach. Here, $C = 0$ indicates the NL approach without a confidence measure. The result is shown in Fig. 4 in which the CERs obtained by the EBW algorithm are cited from Fig. 3. Although the CERs did not decrease monotonically, the EBW algorithm was able to improve the classification performance in general. In contrast, the change in the CERs caused by the NL approach was unstable. For some C and N_u, the CERs became worse than that of the initial model. From the results, it can be said that the EBW algorithm was superior to the NL approach for the JSL data.

6.3 Phoneme Classification

6.3.1 Speech Data Set

As an example of larger data and uneven class distributions, we used the TIMIT corpus [25] of read speech. The phoneme classification tasks on TIMIT have frequently been used to evaluate classifiers for sequential data.

In our experiment, for the training data, we used the standard data sets SX and SI defined in TIMIT. There were $140,099$ sequences or phonemes in this training data set, and they were all used for model initialization. For the test data, we used the core test data set defined in TIMIT. There were $50,754$ sequences or phonemes in this test data set. The mean, maximum, and minimum lengths of the sequences were, respectively, 8.9, 238, and 3 for the training data and 9, 465, and 3 for the test data. In contrast to the gesture data, each class contained different numbers of sequences: from the smallest (149 sequences) to the largest ($12,516$ sequences). As in [26], for training, we grouped the original 64 phoneme categories into 48 as follows: $\{\text{q} \rightarrow '\text{remove}'\}$, $\{\text{ux} \rightarrow \text{uw}\}$, $\{\text{axr} \rightarrow \text{er}\}$, $\{\text{ax-h} \rightarrow \text{ah}\}$, $\{\text{em} \rightarrow \text{m}\}$, $\{\text{nx} \rightarrow \text{n}\}$, $\{\text{eng} \rightarrow \text{ng}\}$, $\{\text{hv} \rightarrow \text{hh}\}$, $\{\text{pcl}, \text{tcl}, \text{kcl} \rightarrow \text{cl}\}$, $\{\text{bcl}, \text{dcl}, \text{gcl} \rightarrow \text{vcl}\}$, $\{\text{h\#}, \text{pau} \rightarrow \text{sil}\}$. As in [26], for testing, we grouped the above 48 phoneme categories into 39 as follows: $\{\text{cl}, \text{vcl}, \text{ep} \rightarrow \text{sil}\}$, $\{\text{el} \rightarrow \text{l}\}$, $\{\text{en} \rightarrow \text{n}\}$, $\{\text{zh} \rightarrow \text{sh}\}$, $\{\text{ao} \rightarrow \text{aa}\}$, $\{\text{ix} \rightarrow \text{ih}\}$, $\{\text{ax} \rightarrow \text{ah}\}$. Thirty-nine dimensional feature vectors were extracted as in [27]: 12 MFCC coefficients, log-energy, and the corresponding delta and delta-delta coefficients were computed at a 10 ms frame rate, using a 25 ms Hamming window.

6.3.2 Preliminary Experiment

In the preliminary experiment, as well as the previous experiment on the gesture data, we searched for a situation where there was little training data relative to the number of free model parameters. As a result, we found that the median of CERs decreased more than 20 points when we added labeled data, when $N_y^{\text{ini}} = 5$, $U^y = 3$ for all classes, and $K = 500$. We focused on this case as an example and evaluated the effect of unlabeled data utilized by the EBW algorithm.

We also examined the effect of M which is introduced in Section 5.4. Varying M among $\{1, 3, 5\}$, we compared the CERs of the learned ETM-HMMs. Since we did not observe any clear improvement by increasing M, we concluded that, as far as the TIMIT corpus is concerned, the choice of M does not affect the performance. Therefore, to minimize the amount of computation, we chose $M = 1$ for the rest of the experiments.

6.3.3 Experimental Results

For $\mathcal{D}_l^{\text{ini}}$, although the class distributions of TIMIT were inhomogeneous, we sampled the data uniformly from all classes. This assumed that we knew there to be 48 classes, but we had no prior knowledge of their distributions. The number of initial labeled data $N_y^{\text{ini}} = 5$ for all y, which comprised about 0.17 percent of all the available training data. In contrast to the labeled data, the unlabeled data were randomly drawn from the real distribution of the whole training data since we usually collect unlabeled data without knowing their true classes. The sampled unlabeled data might reflect the true distribution of the labeled data if the amount were large enough. Either 480 labeled or unlabeled data were added at a time until the total reached $4,800$. For each additional amount, 10 different data sets were created as above. Then, ETM-HMMs ($U^y = 3, K = 500$) were trained on these data sets and tested on the same test data set.

The results of these experiments are shown in Fig. 5. The median of the CERs of the ETM-HMMs learned only from initial labeled data was 66.7 percent. For the ETM-HMMs learned from mixed data containing $4,800$ unlabeled data, the median of CERs decreased to 54.1 percent. Of course the addition of labeled data was more effective (the median of CERs decreased 44.2 percent); nevertheless, the improvement provided by the unlabeled

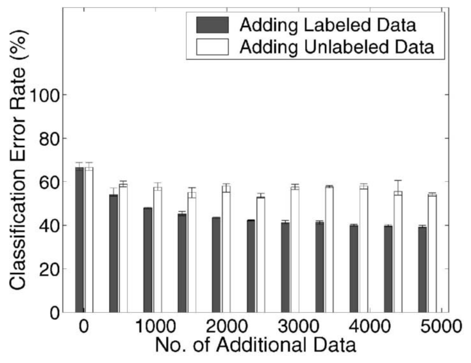

Fig. 5. The change in the classification error rates for TIMIT data when either 480 labeled or unlabeled sequences were added at a time to the initial training data set ($N_l^{\text{ini}} = 240$). The thick bars represent medians of CERs and the thin lines represent upper and lower quartiles. The number of states $U^y = 3$ for each class and the number of Gaussians $K = 500$.

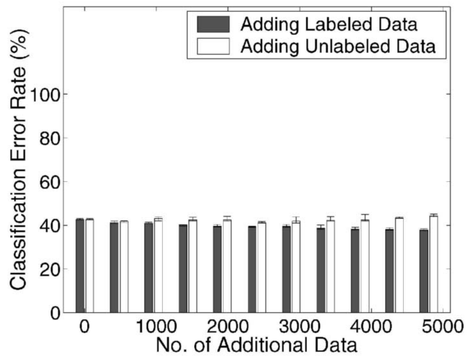

Fig. 6. The change in the classification error rates for TIMIT data when either 480 labeled or unlabeled sequences were added at a time to the initial training data set ($N_l^{\text{ini}} = 2,400$). The thick bars represent medians of CERs and the thin lines represent upper and lower quartiles. The number of states $U^y = 3$ for each class and the number of Gaussians $K = 500$.

data with the EBW algorithm may be valuable since labeled data are usually expensive and not easily available. As JSL gesture data, we presume that the improvement comes from the better estimates of Gaussian parameters.

In this experiment, the improvement provided by unlabeled data was more significant than that for the gesture data. One reason may be the difference in the dimensionalities of the two data sets. In [4], it is argued that unlabeled data are more effective when the feature dimensionality is high. The dimensionality of TIMIT data is about three times higher than that of JSL data.

Unfortunately, a significant improvement could not always be achieved by using unlabeled data. When relatively larger initial labeled data were available, the addition of unlabeled data did not reduce the errors. Fig. 6 shows the case where $N_y^{\text{ini}} = 50$. As with the previous case, $U^y = 3$ and $K = 500$. As can be seen, the addition of unlabeled data did not improve the performance; on the contrary, the addition sometimes had a detrimental effect. For example, when $4,800$ unlabeled data were added, the median of CERs was 44.2 percent, while the initial median of CERs was 42.6 percent. Here, it should be noted that the addition of labeled data did not improve the performance significantly either, although they did not degrade the performance. The main reason for the performance degradation may be that the models responsible for the different classes were close to each other as a result of the addition of unlabeled data. That is, the class boundary provided by the sufficient amount of initial labeled data may become blurred through the addition of unlabeled data.

In conclusion, the addition of unlabeled data by the EBW algorithm seems to be useful when HMMs need to be complex to achieve satisfactory performances but labeled data are too scarce to estimate their parameters accurately. In contrast, it may be not helpful when there are enough labeled data available.

6.3.4 Comparison with Naive Labeling Approach

We examined the NL approach using the same phoneme data and model structure ($U^y = 3, K = 500$). Varying C among $\{0, 0.8, 1.0\}$, we computed the changes in the CERs when unlabeled data were added. Fig. 7 shows the medians of the CERs for 10 data subsets for each amount of added data. For ease of comparison, the results obtained with the EBW algorithm are cited from Fig. 5. Clearly, the performance of the ETM-HMMs learned by the EBW algorithm was better than that with the NL approach. This result implies an advantage of our approach. In addition to the higher CERs, the results of the NL approach were unstable: for some C and N_u, the performance degraded from that of the initial ETM-HMMs. This difference in stability suggests another advantage of our approach over the NL approach.

Here, we discuss the possible reasons for the ineffectiveness of the NL approach. In the above experiment, the CER for the initial model was 66.7 percent. We may regard this CER as indicating poor performance. The pseudolabels generated by such a classifier must be unreliable and the addition of data might have adverse effects. Throughout the experiment, regardless of the amount of additional data, the poor initial parameter estimates were used for the NL approach; in contrast, for the EBW algorithm, both

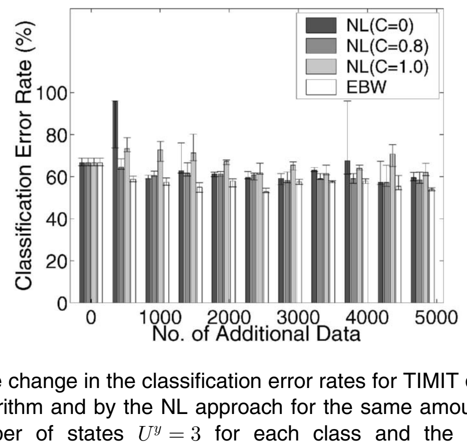

Fig. 7. The change in the classification error rates for TIMIT data by the EBW algorithm and by the NL approach for the same amount of data. The number of states $U^y = 3$ for each class and the number of Gaussians $K = 500$. Either 480 labeled or unlabeled sequences were added at a time to the initial training data set ($N_l^{\text{ini}} = 240$). The thick bars represent medians of CERs and the thin lines represent upper and lower quartiles. In the NL approach, the confidence measure threshold was changed ($C = 0, 0.8, 1.0$).

labeled and unlabeled data can be used from the beginning of the learning process. Possibly, the NL approach may work when the initial models are relatively well trained based solely on initial labeled data. When the initial models are unreliable and the purpose of using unlabeled data is to improve the classification to an acceptable level, as has been considered in this paper, our method may work better than the NL approach.

7 Conclusion

In this paper, we proposed the EBW algorithm to enable the learning of HMMs from both labeled and unlabeled sequential data. Conventionally, in HMM learning, unlabeled sequences have been used heuristically by the NL approach without the guarantee of convergence. In contrast, in the EBW algorithm, the parameter reestimation formulae have been formally derived in the framework of the EM algorithm. We also found that our method utilized unlabeled sequences more effectively than the NL approach in terms of classification performance. Two experimental results on gesture data and speech data showed that the EBW algorithm reduced the classification errors in most cases in contrast to the NL approach.

Although when the initial labeled data were scarce, our method could compensate for the insufficiency of labeled training data by using unlabeled data, when the initial labeled data were sufficient, the EBW algorithm sometimes had a detrimental effect on the classification performance. This is a limitation of our approach and the situations in which unlabeled sequences do not help should be studied further. The reason why adding unlabeled data could not monotonically improve the performance can probably be explained based on the analysis in [28].

In future work, we can apply our method for adaptation where unseen test data whose properties are different from those of training data can be regarded as unlabeled data. Furthermore, there exists a more sophisticated NL approach called cotraining [29], in which feature vectors need to be separated into two feature sets, each of which is capable of learning a classifier. When we have such a redundant data set, it is interesting to compare the NL approach, the EBW algorithm, and cotraining.

Appendix A

Scaling for the Unlabeled Posteriors

This appendix describes the scaling technique we used to prevent the computational problem that occurs during the calculation of unlabeled posteriors. In HMMs, posteriors are efficiently calculated by the forward-backward algorithm [23]. However, if the observed sequence is long, the values of the intermediate variables used in the forward-backward algorithm become too small to be handled within the precision range of computers. For labeled data, this problem can be avoided by applying a scaling technique [23]. For unlabeled data, however, it is not applicable. In the following, we show why conventional scaling does not work and present a modified scaling procedure that is applicable to unlabeled data.

As an example, we consider the calculation of transition posteriors for labeled or unlabeled data (13) or (17). Here, and in the following, for simplicity of notation, we assume a single observation and omit the index n. Let $\alpha_t^y(i) = p(\mathbf{x}_1, \dots, \mathbf{x}_t, s_t = i \mid y, \Theta)$ be the forward variable unscaled and computed from time 1 to t in the state i at time t. Let $\beta_t^y(j) = p(\mathbf{x}_{t+1}, \dots, \mathbf{x}_T \mid s_t = i, y, \Theta)$ be the backward variable unscaled and computed from time T to t in the state j at time t. Let $\tilde{\alpha}_t^y(i)$ be the forward variable scaled from time 1 to time $t-1$ in the state i at time t, and $\tilde{\beta}_t^y(j)$ be the backward variable scaled from time T to time $t+1$ in the state j at time t. Let $\mathrm{W}_t^y = 1/\sum_i \tilde{\alpha}_t^y(i)$ be the scaling coefficient at time t, $\check{\alpha}_t^y(i) = \mathrm{W}_t^y \cdot \tilde{\alpha}_t^y(i)$ be the forward variable scaled from time 1 to time t, and $\check{\beta}_t^y(j) = \mathrm{W}_t^y \cdot \tilde{\beta}_t^y(j)$ be the backward variables scaled from time T to time t. Between scaled forward or backward variables, unscaled forward or backward variables, and scaling coefficients, the following relationships holds [23]:

$$\check{\alpha}_t^y(i) = \left[\prod_{t'=1}^{t} \mathrm{W}_{t'}^y\right] \cdot \alpha_t^y(i), \tag{27}$$

$$\check{\beta}_{t+1}^y(j) = \left[\prod_{t'=t+1}^{T} \mathrm{W}_{t'}^y\right] \cdot \beta_{t+1}^y(j). \tag{28}$$

For labeled data, if this scaling procedure is applied, since the scaling coefficients in the numerator and the denominator cancel out, $\gamma_t^y(i,j)$ can be calculated by the following equation:

$$\gamma_t^y(i,j) = \frac{\check{\alpha}_t^y(i) a_{ij}^y b_j^y(\mathbf{x}_{t+1}) \check{\beta}_{t+1}^y(j)}{\sum_{i \in F^y} \check{\alpha}_T^{y}(i)}, \tag{29}$$

where $b_j^y(\mathbf{x}_{t+1}) = \sum_k c_{jk}^y \mathcal{N}(\mathbf{x}_{t+1} \mid \mu_k, \Sigma_k)$ and F^y represents the final state among the states of the class y HMM. For unlabeled data, if the scaling procedure is applied, $\lambda_t(y,i,j)$ may be calculated by the following equation:

$$\lambda_t(y,i,j) = \frac{\omega_y \check{\alpha}_t^y(i) a_{ij}^y b_j^y(\mathbf{x}_{t+1}) \check{\beta}_{t+1}^y(j)}{\left[\prod_{t'=1}^{T} \mathrm{W}_{t'}^y\right] \sum_{y'} \omega_{y'} \sum_{i \in F^{y'}} \alpha_T^{y'}(i)}. \tag{30}$$

In (30), with the help of the scaling, each variable in the numerator can be calculated at each time t; while, in the denominator, the product of scaling coefficients $\prod_{t'=1}^{T} \mathrm{W}_{t'}^y$ cannot be calculated when the length of the sequence T is large. This is because W_t^y is usually large at each t and the calculation of the product often reaches infinity on computers. To avoid this problem, the following successive computation procedure is introduced. We transform the denominator of (30) as follows:

$$\begin{aligned} &\left[\prod_{t'=1}^{T} \mathrm{W}_{t'}^y\right] \sum_{y'} \omega_{y'} \sum_{i \in F^{y'}} \alpha_T^{y'}(i) \\ &= \prod_{t'=1}^{T} \mathrm{W}_{t'}^y \sum_{y'} \omega_{y'} \frac{1}{\prod_{t'=1}^{T} \mathrm{W}_{t'}^{y'}} \\ &= \sum_{y'} \omega_{y'} \prod_{t'=1}^{T} \left(\frac{\mathrm{W}_{t'}^y}{\mathrm{W}_{t'}^{y'}}\right). \end{aligned} \tag{31}$$

By so doing, in most practical cases, the value of $\mathrm{W}_t^y/\mathrm{W}_t^{y'}$ remains computable. Other posteriors for unlabeled data can be calculated by applying this transformation.

Appendix B

Convergence Criterion

This appendix describes the convergence criterion used in this paper. When the increase in likelihood is smaller than a predefined threshold, we regard the EBW algorithm to be converged. This is possible on condition that the likelihood, or log-likelihood, is computable. The log-likelihood of an ETM-HMM is given as follows:

$$\begin{aligned}\mathcal{L}(\Theta \mid \mathcal{D}) &= \sum_{y=1}^{Y}\sum_{n\in\mathcal{I}_y}\log p(X_n, y_n \mid \Theta) + \sum_{n=1}^{N_u}\log p(X_n \mid \Theta)\\ &= \sum_{y=1}^{Y}\sum_{n\in\mathcal{I}_y}\log \omega_y p(X_n \mid y, \Theta)\\ &\quad + \sum_{n=1}^{N_u}\log\sum_{y}\omega_y p(X_n \mid y, \Theta).\end{aligned} \tag{32}$$

When scaling coefficients $\mathrm{W}_{n_t}^y$ defined in Appendix A are used and since $\prod_{t=1}^{T_n}\mathrm{W}_{n_t}^y = p(X_n \mid y, \Theta)$ holds, (32) is rewritten as:

$$\begin{aligned}\mathcal{L}(\Theta \mid \mathcal{D}) &= \sum_{y=1}^{Y}\sum_{n\in\mathcal{I}_y}\log\frac{\omega_y}{\prod_{t=1}^{T_n}\mathrm{W}_{n_t}^y}\\ &\quad + \sum_{n=1}^{N_u}\log\sum_{y=1}^{Y}\frac{\omega_y}{\prod_{t=1}^{T_n}\mathrm{W}_{n_t}^y}.\end{aligned} \tag{33}$$

As has been explained in Appendix A, $\prod_{t=1}^{T_n}\mathrm{W}_{n_t}^y$ cannot be computed when T_n is large. To make the log-likelihood computable, we introduce a metascaling coefficient V whose value is defined empirically according to the data. Let N be the sum of N_l and N_u. By substituting the product of V from (33), we have:

$$\begin{aligned}\mathcal{L}'(\Theta \mid \mathcal{D}) &= \mathcal{L}(\Theta \mid \mathcal{D}) - \log\prod_{n=1}^{N}\prod_{t=1}^{T_n}\mathrm{V}\\ &= \mathcal{L}(\Theta \mid \mathcal{D}) - \sum_{y=1}^{Y}\sum_{n\in\mathcal{I}_y}\log\prod_{t=1}^{T_n}\mathrm{V} - \sum_{n=1}^{N_u}\log\prod_{t=1}^{T_n}\mathrm{V}\\ &= \sum_{y=1}^{Y}\sum_{n\in\mathcal{I}_y}\log\frac{\omega_y}{\prod_{t=1}^{T_n}(\mathrm{V}\cdot\mathrm{W}_{n_t}^y)}\\ &\quad + \sum_{n=1}^{N_u}\log\sum_{y=1}^{Y}\frac{\omega_y}{\prod_{t=1}^{T_n}(\mathrm{V}\cdot\mathrm{W}_{n_t}^y)}.\end{aligned} \tag{34}$$

Thus, if we choose V appropriately so that $\prod_{t=1}^{T}(\mathrm{V}\cdot\mathrm{W}_{n_t}^y)$ is computable, we can obtain $\mathcal{L}'(\Theta \mid \mathcal{D})$. Although $\mathcal{L}'(\Theta \mid \mathcal{D})$ is no longer the log-likelihood itself, since the substituted value, $\log\prod_{n=1}^{N}\prod_{t=1}^{T_n}\mathrm{V}$, is a constant, we can use the change in $\mathcal{L}'(\Theta \mid \mathcal{D})$ to determine the convergence of the EBW algorithm.

Acknowledgments

The authors would like to thank Erik McDermott and Atsushi Nakamura for their helpful comments on speech data.

References

[1] S.J. Raudys and A.K. Jain, "Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 3, pp. 252-264, Mar. 1991.

[2] S. Ganesalingam and G.J. McLachlan, "Some Efficiency Results for the Estimation of the Mixing Proportion in a Mixture of Two Normal Distributions," *Biometrics,* vol. 37, pp. 22-33, Mar. 1981.

[3] G.J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition.* John Wiley and Sons, 1992.

[4] B.M. Shahshahani and D.A. Landgrebe, "The Effect of Unlabeled Samples in Reducing the Small Sample Size Problem and Mitigating the Hughes Phenomenon," *IEEE Trans. Geoscience and Remote Sensing,* vol. 32, no. 5, pp. 1087-1095, Sept. 1994.

[5] D.J. Miller and H.S. Uyar, "A Mixture of Experts Classifier with Learning Based on Both Labelled and Unlabelled Data," *Advances in Neural Information Processing Systems,* M.C. Mozer, M.I. Jordan, T. Petsche, eds., vol. 9, pp. 571-577, 1997.

[6] K. Nigam, A.K. McCallum, S. Thrun, and T. Mitchell, "Text Classification from Labeled and Unlabeled Documents Using EM," *Machine Learning,* vol. 39, nos. 2/3, pp. 103-134, May 2000.

[7] T. Kemp and A. Waibel, "Unsupervised Training of a Speech Recognizer: Recent Experiments," *Proc. Eurospeech,* vol. 6, pp. 2725-2728, 1999.

[8] L. Lamel, J.L. Gauvain, and G. Adda, "Lightly Supervised and Unsupervised Acoustic Model Training," *Computer Speech and Language,* vol. 16, no. 1, pp. 115-129, Jan. 2002.

[9] L.R. Bahl, F. Jelinek, and R. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 5, pp. 179-190, Mar. 1983.

[10] T.E. Starner and A. Pentland, "Visual Recognition of American Sign Language Using Hidden Markov Models," *Proc. Int'l Workshop Automatic Face and Gesture Recognition,* pp. 189-194, 1995.

[11] A. Krogh, M. Brown, I.S. Mian, K. Sjölander, and D. Haussler, "Hidden Markov Models in Computational Biology Applications to Protein Modeling," *J. Molecular Biology,* vol. 235, no. 5, pp. 1501-1531, Feb. 1994.

[12] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun, "A Practical Part-of-Speech Tagger," *Proc. Third Conf. Applied Natural Language Processing,* pp. 133-140, 1992.

[13] D.M. Bikel, R. Schwartz, and R.M. Weischedel, "An Algorithm that Learns What's in a Name," *Machine Learning,* vol. 34, nos. 1-3, pp. 211-231, Feb. 1999.

[14] B. Merialdo, "Tagging English Text with a Probabilistic Model," *Computational Linguistics,* vol. 20, no. 2 pp. 155-171, June 1994.

[15] D. Elworthy, "Does Baum-Welch Reestimation Help Taggers?" *Proc. Fourth Conf. Applied Natural Language Processing,* pp. 53-58, 1994.

[16] K. Seymore, A. McCallum, and R. Rosenfeld, "Learning Hidden Markov Model Structure for Information Extraction," *Proc. AAAI Workshop Machine Learning for Information Extraction,* pp. 37-42, 1999.

[17] J.R. Bellegarda and D. Nahamoo, "Tied Mixture Continuous Parameter Modeling for Speech Recognition," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. 38, no. 12, pp. 2033-2045, Dec. 1990.

[18] X.D. Huang, "Phoneme Classification Using Semicontinuous Hidden Markov Models," *IEEE Trans. Signal Processing,* vol. 40, no. 5, pp. 1062-1067, May 1992.

[19] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc. B,* vol. 39, no. 1, pp. 1-38, 1977.

[20] M. Inoue and N. Ueda, "HMMs for Both Labeled and Unlabeled Time Series Data," *Proc. 11th IEEE Workshop Neural Networks for Signal Processing,* pp. 93-102, 2001.

[21] G. Aversano, A. Esposito, A. Esposito, and M. Marinaro, "A New Text-Independent Method for Phoneme Segmentation," *Proc. 44th IEEE Midwest Symp. Circuits and Systems,* vol. 2, pp. 516-519, 2001.

[22] L.E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Function of Markov Chains," *The Annals of Math. Statistics,* vol. 41, no. 1, pp. 164-171, 1970.

[23] X.D. Huang, Y. Ariki, and M.A. Jack, *Hidden Markov Models for Speech Recognition.* Edinburgh: Edinburgh Univ. Press, 1990.
[24] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification,* second ed. John Wiley and Sons, 2001.
[25] J.S. Garofolo, L.F. Lamel, W.M. Fisher, J.G. Fiscus, D.S. Pallett, and N.L. Dahlgren, *DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CD-ROM.* Gaithersburg, Md.: Nat'l Inst. of Standards and Technology, 1993.
[26] K.F. Lee and H.W. Hon, "Speaker-Independent Phone Recognition Using Hidden Markov Models," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. 37, no. 11, pp. 1641-1648, Nov. 1989.
[27] E. McDermott and S. Katagiri, "String-Level MCE for Continuous Phoneme Recognition," *Proc. Eurospeech,* vol. 1, pp. 123-126, 1997.
[28] A. Corduneanu and T. Jaakkola, "Continuation Methods for Mixing Heterogeneous Sources," *Uncertainty in Artificial Intelligence: Proc. 18th Conf.,* pp. 111-118, 2002.
[29] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training," *Proc. Workshop Computational Learning Theory,* pp. 92-100, 1998.

Masashi Inoue received the BA degree in natural science from the International Christian University (ICU), Japan, in 1999 and the MS degree in information science from the Nara Institute of Science and Technology (NAIST), Japan, in 2001. He is currently a doctoral candidate with the Graduate School of Information Science at NAIST. His research interests include pattern recognition and information extraction in human communication. He is a student member of the Organization for Human Brain Mapping (OHBM) and the IEEE.

Naonori Ueda received the BS, MS, and PhD degrees in communication engineering from Osaka University, Osaka, Japan, in 1982, 1984, and 1992, respectively. In 1984, he joined the Electrical Communication Laboratories, NTT, Kanagawa, Japan. In 1991, he joined the NTT Communication Science Laboratories, Kyoto, Japan, as a senior research scientist. He is an executive manager of the Intelligent Communication Laboratory. From 1993 to 1994, he was a visiting scholar at Purdue University, West Lafayette, Indiana. His current research interests are statistical pattern recognition and statistical analysis of complex data. He is a member of the Institute of Electronics, Information, and Communication Engineers (IEICE), the Information Processing Society of Japan (IPSJ), the Japanese Neural Networks Society (JNNS), and the IEEE.