Acoustic Model Adaptation for Speech Recognition

Koichi SHINODA \(^{1}\), Senior Member

SUMMARY Statistical speech recognition using continuous-density hidden Markov models (CDHMMs) has yielded many practical applications. However, in general, mismatches between the training data and input data significantly degrade recognition accuracy. Various acoustic model adaptation techniques using a few input utterances have been employed to overcome this problem. In this study, we survey these adaptation techniques, including maximum a posteriori (MAP) estimation, maximum likelihood linear regression (MLLR), and eigenvoice. We also present a schematic view called the adaptation pyramid to illustrate how these methods relate to each other.

key words: speech recognition, acoustic model adaptation, hidden Markov models

1. Introduction

In statistical speech recognition, there are usually mismatches between the conditions under which the model was trained and those of the input. Mismatches may occur because of differences between speakers, environmental noise, and differences in channels. They should be compensated in order to obtain sufficient recognition performance. Acoustic model adaptation is the process of modifying the parameters of the acoustic model used for speech recognition to fit the actual acoustic characteristics by using a few utterances from the target user.

The history of acoustic model adaptation started with speaker adaptation, i.e., adaptation to speakers. Speaker-dependent speech recognition systems that were intended to recognize utterances from one target speaker were studied until the early 1980’s. In this system, the target speaker registers his/her utterance for each word in the recognition vocabulary beforehand to create its template pattern. In recognition, each template pattern is matched to his/her input speech, and the word whose template has the smallest distance to the input speech is selected as the recognized word. In practice, the number of utterances to be registered should be as small as possible to decrease the load of users to register their voice. On the other hand, if a speaker uses a speaker-dependent system for the other speaker, its recognition accuracy seriously deteriorates, since acoustic characteristics vary much from speaker to speaker. Speech recognition systems which require only a few utterances from a user and have as high recognition performance as speaker-dependent systems were strongly demanded. This was the motivation for researchers to start development of speaker adaptation techniques.

Speaker adaptation would be easy if we could separate the two intrinsic features of speech, i.e., the phonetic feature representing the content and the speaker feature representing the characteristics of the individual speaker. However, several factors underlie speaker differences. There are intrinsic factors caused by differences in the length and the shape of the vocal tract. These factors also change as an individual ages. Other factors are para-linguistic; they depend on the speaker’s intention, physical condition, speaking style, dialect, emphasis, and emotion. The Lombard effect in noisy environments [69] belongs to this category. Although the intrinsic causes are more important, it is difficult to separate them from the other causes. Moreover, since speaker characteristics vary from phoneme to phoneme it is hardly possible to separate them from the phonetic features. For these reasons, speaker adaptation research has avoided the essential problem of what features represent speaker characteristics.

Speech recognition technologies using hidden Markov models (HMMs) have significantly advanced since the late 1980’s. In particular, speech recognition algorithms often employ continuous density HMMs (CDHMMs) using triphones as recognition units and a Gaussian mixture distribution as the output distribution. In CDHMMs, a variety of speech signals are represented as a continuous density distribution whose parameters are determined by using the expectation-maximization (EM) algorithm to make a maximum-likelihood (ML) estimation. The use of utterances from many speakers for training enables these models to represent not only phonetic features but also speaker features. Although this ability has made speaker-independent systems practical, the systems still do not perform as well as speaker-dependent systems in which the HMM parameters are estimated from a sufficient amount of utterances from one target user. Moreover, other mismatches besides speaker differences, such as those due to environmental noise and channels, also degrade recognition accuracy. This means that adaptation techniques are important to speech recognition using CDHMMs.

There have been several surveys on adaptation techniques for speech recognition. For example, Lee and Huo [49], Woodland [94] and Sagayama et al. [74] surveyed the adaptation techniques in existence around 2001. Moreover, Furui [21] comprehensively reviewed generalization
techniques for training and adaptation, and Bellegarda [3] surveyed language model adaptation techniques for large vocabulary continuous speech recognition. Our paper aims to give the reader a unified view of present-day acoustic model adaptation methods.

2. What is Acoustic Model Adaptation?

2.1 Problem

Let us assume that we have an acoustic model that has high recognition accuracy under one condition, and let us consider how we can improve its speech recognition accuracy under some other new condition by using only a few utterances worth of speech data (adaptation data).

The number of parameters in a triphone CDHMM is generally large. It consists of a few thousand states, and each state has a Gaussian mixture distribution with dozens of mixture components. Each mixture component has a mean vector with dozens of elements and its corresponding covariance matrix. Moreover, it also includes parameters for expressing transition probabilities and initial probabilities. In total, the parameters usually number more than 1,000,000. In contrast, the number of data samples that can be obtained from a few utterances is much more limited. Each utterance is typically about 1 sec. long, and one feature vector with dozens of elements can be obtained every 10 msec. That means only a few thousand data samples can be obtained from a few utterances.

In this situation, ML estimation using the EM algorithm cannot precisely estimate the model parameters. As a result, recognition accuracy would be much worse than under the original conditions. This is called the data sparseness problem.

Acoustic model adaptation aims to overcome the above problems. Let \( \theta_i \) be the parameter set of an initial model \( i \) given beforehand, and let \( \hat{\theta} \) be that of the target model to be determined. Acoustic model adaptation can be defined as a process to find a mapping function \( f \) from the space of parameters of the initial models to the space of the target model using adaptation data

\[
\hat{\theta} = f(\theta_1, \ldots, \theta_n),
\]

where \( n \) is the number of initial models provided. We hereafter call this mapping function \( f \) an adaptation model (Fig. 1). When \( f \) consists of mapping functions defined for each CDHMM parameter independently, adaptation using it is called direct adaptation. Adaptation using adaptation models with parameter sharing are called indirect adaptation [49].

An adaptation model should meet the following requirements.

1. It should improve recognition accuracy even with a small amount of adaptation data.
2. As the amount of adaptation data increases, it should make the recognition accuracy asymptotically approach the accuracy of a matched model.

Here, a matched model is the CDHMM whose parameters are estimated using a sufficient amount of data collected in the new condition. Figure 2 illustrates these requirements.

Since nothing comes from nothing, we need prior knowledge to design an adaptation model. In this sense, acoustic model adaptation can be regarded as acoustic model training with constraints placed on the model parameters and the constraints are obtained from prior knowledge. Back in the age of speaker-dependent speech recognition, the prior knowledge was taken from another speaker’s data and his/her speaker-dependent model. Once speech recognition using CDHMMs became popular, a large amount of data from various speakers and conditions became available for use as prior knowledge. We use such prior knowledge to find an adaptation model that well represents the differences between each initial model and the target model.

The first requirement can be met by designing good adaptation models with only a few free parameters to be estimated. However, such simple models may fail to fulfill the second requirement; they may not improve recognition accuracy once a larger amount of adaptation data becomes available. This happens because the models are too simple to represent the richness of information contained in a large amount of adaptation data. Thus, the adaptation model should have an appropriate number of free parameters (i.e., an appropriate model size).
2.2 Approach

According to the above discussion, the following problems should be solved in order to get a good adaptation model.

1. Find a model class that well represents the prior knowledge obtained from the utterances of many speakers.
2. Optimize a model size so that it is appropriate for the amount of the available adaptation data.

To simplify the following discussion, let us limit the differences to only speaker differences. Let us also limit the HMM parameters to be updated to the mean vectors for the Gaussian mixture components of the CDHMM states. Let us call the data of many speaker’s utterances used as prior knowledge as training data. Then, the information in training data can be represented by a matrix in which each column corresponds to one speaker and each row corresponds to an element in a mean vector in a CDHMM, representing a phonetic feature. We call this matrix the speaker-phone matrix (see Fig. 3). Each column vector in this matrix is a speaker-supervector whose elements are each one phonetic feature for each speaker in the training data, and each row vector is a phone-supervector, each element of which represents one speaker feature for the corresponding phonetic feature.

The first problem of finding an appropriate model class comes down to determining what kind of information should be extracted from this speaker-phone matrix.

The second problem can be solved in two different ways. The first way is to make sure the adaptation model class fits the amount of available adaptation data. This is easy if we know how much adaptation data is used in the adaptation process. However, this idea is not applicable in most cases in which the amount of adaptation data varies. The second way is to cluster the parameters of the acoustic model and share the same mapping function within each cluster. This piece-wise approach implicitly assumes embedded structures exist inside the model parameter space.

There are many ways to cluster model parameters, but all of them require the number of clusters, i.e., the degree of sharing, to be controlled according to the amount of data. The clustering process can be represented by a tree structure in which each node represents one cluster. Any clustering result can be represented by making a cut in this tree. A cut consisting of only the root node corresponds to clustering all the model parameters into one cluster. A set of all leaf nodes corresponding to case clustering is not performed. As we described above, acoustic model parameters in training data consisting of utterances from many speakers can be represented as a speaker-phone matrix. Hence, the tree structure can be illustrated as an adaptation pyramid whose base is the speaker-phone matrix (see Fig. 3). In fact, all adaptation models employ parameters clustering that can be represented by a cut inside this pyramid. An adaptation model corresponding to a cut in the upper layer in the tree is a coarse model which represents global mapping. As a cut goes down within the tree, its corresponding model becomes finer, and represents more local mapping.

The most famous method for dealing with the first problem is maximum-likelihood linear regression (MLLR) [50]. In this method, all the elements of each row vector (phone-supervector) in the speaker-phone matrix are clustered into one value so that all the speakers in the training data share the same parameter set, and an affine mapping is applied to the space spanned by those parameters. Another popular approach, eigenvoice [45], clusters column vectors (speaker-supervectors) into representative vectors and then estimates the projection of the speaker-supervector of the target speaker to the subspace spanned by the representative vectors.

Here, we should emphasize that the adaptation models are not based on the actual differences between speakers or other factors. They are selected mainly for the sake of tractability and computational cost. Hence, in many situations, the second problem above is more important than the first. For example, Loof et al. argued that shifting and affine mapping have almost the same performance when the proper model size for each is given [52].

In acoustic model adaptation, it is also important to develop good parameter estimation methods that can deal with insufficient data. A typical approach is the Bayesian one wherein prior knowledge is introduced in a straightforward way. The maximum a posteriori (MAP) estimation is the most popular of these approaches (e.g., [25], [47]); others include ML estimation and discriminative learning (these are also used in acoustic model training).

The above discussion holds true for adaptations to other differences such as environmental noise and channels. In such case, the speaker-phone matrix is replaced with a noise-phone matrix or a channel-phone matrix.

The rest of this paper is organized as follows. We discuss implementation issues in the following section and parameter estimation in Sect. 4. Sections 5 and 6 explain two popular methods of finding an appropriate model class. Section 7 describes methods for finding an appropriate model size for the available data. Finally, we discuss adaptive training, which is closely related to acoustic model adaptation in Sect. 8.
3. Implementation

3.1 Supervised and Unsupervised Adaptation

Acoustic model adaptation techniques are categorized into supervised adaptation and unsupervised adaptation. In supervised adaptation, a transcription exists for each utterance. In unsupervised adaptation, it does not.

In supervised adaptation, the user should register his or her own voice. To do so, the system shows the user predetermined words or sentences and asks him/her to utter them. The speaker registration process used in dictation software is an example of this process. The early speaker adaptation techniques required up to 20 minutes worth of speech data. Nowadays, thanks to the progress of speaker-independent recognition and speaker adaptation, most systems require only one minute worth of data.

While dictation software is intended to be used for a long time, other applications, e.g., airline ticket reservations by telephone, are intended to be used by one person for only a short time. Unsupervised adaptation techniques are needed for such short-period applications since users should not have to spend time registering their voices. Most techniques are related to supervised adaptation in that they use transcriptions obtained from speaker-independent speech recognition as the supervising signal for adaptation\(^1\). These techniques usually perform well when the recognition accuracies of speaker independent speech recognition are high enough to get reliable transcriptions.

Incorrect supervised signals generated by misrecognitions may significantly degrade adaptation performance. Some techniques for alleviating their effect calculate a confidence measure, such as a posterior probability, for each utterance and use only those utterances with confidence measures larger than a predetermined threshold for adaptation (e.g., [67]). Matsui et al. used N-best sentences output from a speech recognizer as the supervising signals [57]. More recently, Shinozaki et al. proposed a cross-validation based scheme [82].

3.2 Batch and On-Line Adaptation

Acoustic model adaptation can also be categorized as batch adaptation or on-line adaptation [101]. Batch adaptation is done after all the available utterances are collected, whereas on-line adaptation is done each time one utterance is obtained. Batch adaptation requires sufficient memory to store the statistics to be used for parameter estimation, while on-line adaptation does not require such a large memory.

Batch adaptation performs better than on-line adaptation when both methods use the same adaptation data, since it can simulate any on-line adaptation. Thanks to recent advances in computational technology, high CPU speeds and large memories, batch adaptation can use all the previous utterances each time a new utterance is obtained.

On-line adaptation is preferable for applications such as speech recognition during meetings, where the speakers often change and the change points are not given beforehand. The forgetting parameters [31] of on-line adaptation should be carefully tuned so as not to use utterances obtained before a certain point in time.

The above explanation indicates that supervised batch adaptation is fundamental and unsupervised adaptation and on-line adaptation are its applications. Hence, in what follows, we will discuss supervised batch adaptation unless otherwise noted.

4. Parameter Estimation

The parameter estimation methods include ML estimation, MAP estimation, Bayes adaptation, and discriminative learning. We shall focus on MAP estimation and Bayes estimation, which are characteristic methods, and briefly touch on discriminative learning.

4.1 MAP Estimation

Maximum a posteriori (MAP) estimation (e.g., [13]) is used in statistical modeling and has a wide range of applications. In particular, it has been often used for acoustic model adaptation (e.g., [25]). It estimates model parameters more robustly than ML estimation when the amount of data is small, and its estimates asymptotically approach ML estimations as the amount of data increases.

Let \( f(x|\theta) \) be the probability density function (pdf) of variable \( x \). We estimate its parameter \( \theta \) by using \( T \) samples of \( x \), \( X = \{x_1, \ldots, x_T\} \). In ML estimation, the parameter is estimated as follows.

\[
\hat{\theta} = \text{argmax}_{\theta} f(X|\theta),
\]

where \( \hat{\theta} \) is the maximum likelihood estimator of \( \theta \). In MAP estimation, \( \theta \) is regarded as a random variable that follows a certain pdf. We expect that our knowledge about it increases as we observe data samples. The parameter distribution before observing the data is called a prior distribution. Let \( g(\theta) \) be the prior distribution for \( \theta \). The pdf of the parameter after observing \( X \), \( g(\theta|X) \) is called a posterior distribution, and it is written as follows (Bayes’ Theorem),

\[
g(\theta|X) = \frac{f(X|\theta)g(\theta)}{\int f(X|\theta)g(\theta)d\theta}.
\]

MAP estimation obtains the value of \( \hat{\theta} \) that maximizes the mode of the posterior distribution, that is, the value which gives the maximum of the posterior distribution:

\[
\hat{\theta} = \text{argmax}_{\theta} g(\theta|X) \\
= \text{argmax}_{\theta} f(X|\theta)g(\theta).
\]

\(^1\)Although there are unsupervised adaptation methods that are not related to supervised adaptation (i.e., [20]), their recognition accuracies are not as good as the unsupervised methods that are related to supervised adaptation.
When we have little knowledge about \( \theta \), we should select a uniform distribution over the range of possible \( \theta \) values as the prior distribution\(^1\). In this case, the MAP estimator becomes almost identical to the ML estimator.

There are no theoretically correct answers as to which class to use for prior pdfs and how to set their parameters. Users may determine them according to their own preference by making observations. However, we can analytically obtain the MAP estimator when \( f(x) \) has sufficient statistics with a fixed dimension, and such an \( f(x) \) should belong to an exponential family. Furthermore, when \( f(x) \) belongs to an exponential family and we choose the prior distribution from the same family of the kernel distribution of \( f(x) \) (the distribution whose parameters are sufficient statistics of \( f(x) \) only), which we call the conjugate family, the posterior distribution accordingly belongs to the same family. This fact makes our calculation much easier. This type of prior distribution is called a natural conjugate prior distribution, and used in many adaptation techniques.

4.2 MAP Estimation for the Normal Distribution

Let us consider the problem of estimating the parameters of a multi-dimensional normal distribution as a simple example of MAP estimation. Let the vector \( x \) be a \( k \)-dimensional random variable. We call \( x \) a feature vector. We assume that its pdf \( p(x) \) is a multi-dimensional normal distribution whose mean vector \( \mu \) and covariance \( \Sigma \) are unknown.

\[
N(x|\mu, \Sigma) = (2\pi)^{-\frac{k}{2}}|\Sigma|^{-\frac{1}{2}} \exp \left[ -\frac{1}{2}(x - \mu)'\Sigma^{-1}(x - \mu) \right].
\] (5)

The symbol \( ' \) in the above equation indicates transposition. ML estimation selects the parameter set that maximizes the likelihood function \( f(X|\theta) \), defined as

\[
f(X|\theta) = \prod_{i=1}^{T} p(x_i|\theta).
\] (6)

The ML estimator \( \hat{\theta} = (\hat{\mu}, \hat{\Sigma}) \) is calculated as follows:

\[
\hat{\mu} = \frac{1}{T} \sum_{t=1}^{T} x_t,
\] (7)

\[
\hat{\Sigma} = \frac{1}{T} \sum_{t=1}^{T} (x_t - \hat{\mu})(x_t - \hat{\mu})'.
\] (8)

The natural conjugate prior for a multidimensional normal distribution \( N(x|\mu, \Sigma) \) is the normal-Wishart distribution:

\[
g(\mu, \Sigma|\mu_0, \Sigma_0, \alpha, \tau) \propto |\Sigma|^{-\frac{k+\alpha}{2}} \exp \left[ -\frac{\tau}{2}(\mu - \mu_0)'\Sigma^{-1}(\mu - \mu_0) \right] \times \exp \left[ -\frac{1}{2}\text{tr}(\Sigma_0\Sigma^{-1}) \right].
\] (9)

where \((\mu_0, \Sigma_0, \alpha, \tau)\) are the parameters of the prior distribution, and \(\alpha > k - 1, \tau > 0\). \(\mu_0\) is a \(k\)-dimensional vector and \(\Sigma_0\) is a \(k \times k\) positive-definite matrix. The prior distribution is the product of the prior distribution for the mean vector when the prior for the covariance is \(\tau^{-1}\Sigma\) and the marginal distribution for the covariance. Note that the marginal distribution for the mean vector is a \(t\) distribution, not a normal distribution. The MAP estimator \(\hat{\theta} = (\hat{\mu}, \hat{\Sigma})\) maximizes the following posterior probability:

\[
g(\mu, \Sigma|X) = C \prod_{t=1}^{T} p(x_t|\mu, \Sigma)g(\mu, \Sigma|\mu_0, \alpha, \tau).
\] (10)

where \(C\) is a scaling factor depending on \(X\) but not on \((\mu, \Sigma)\). After a simple calculation, we obtain

\[
\hat{\mu} = \frac{\tau\mu_0 + \sum_{t=1}^{T} x_t}{\tau + T},
\] (11)

\[
\hat{\Sigma} = \frac{\Sigma_0 + \sum_{t=1}^{T} (x_t - \hat{\mu})(x_t - \hat{\mu})'}{(\alpha - k) + T}.
\] (12)

It can be easily seen from Eqs. (11), (12) that the MAP estimator in this case is the weighted average of the parameter of the prior distribution and the ML estimator. The weight for the ML parameter is small when the amount of data is small, and it increases as the amount of data increases. When no sample is available, the MAP estimator \(\hat{\theta} = (\hat{\mu}, \hat{\Sigma})\) is identical with the parameters of prior distribution, \((\mu_0, \Sigma_0)\). As the number of samples \(N\) increases, it approaches the ML estimator \(\hat{\theta} = (\hat{\mu}, \hat{\Sigma})\).

4.3 MAP Estimation for CDHMMs

Next, let us discuss a MAP estimation method for CDHMM parameters\(^2\). This method is called MAP adaptation, where the CDHMM itself is the adaptation model.

Let \(A = \{I, A, W, B\}\) be the parameter set of a CDHMM. Here, \(I = \{I_i\}\) is the set of initial probabilities, \(A = \{a_{ij}\}\) is the set of transition probabilities, \(W = \{w_{ik}\}\) is the set of mixture weights in the Gaussian mixture distribution, and \(B = \{b_{ik}(x)\}\) is the set of pdfs in each mixture component, where \(i, j\) are state indexes, and \(k\) is an index for each mixture component in a state.

In general, models with hidden variables, such as HMMs, do not have natural conjugate priors. Accordingly, the MAP estimator cannot be analytically calculated for these models. To overcome this problem, we assume that \(I, A, W,\) and \(B\) are independent from each other, and furthermore, their elements are independent from each other. Accordingly, the prior distribution can be defined as the joint probability of the natural conjugate prior for each parameter\(^2\). Here, the normal-Wishart distribution can be used as the prior for normal distribution, and the Dirichlet

\(^1\)We call such a prior distribution a non-informative prior distribution. An example is Jeffreys’ prior distribution.

\(^2\)But this is not the best way to obtain the MAP estimator.
distribution can be used as the prior for the initial probability, transition probability, and mixture weight. The prior for an HMM can be expressed as follows.

$$g(A) = g(T)g(g(A))g(W)g(B)$$

$$= C \prod_{i=1}^{N} \left[ a_{ij}^{\eta_{i,j} - 1} \right] \left( \sum_{i=1}^{N} a_{ij} \right)^{\eta_{i,j} - 1} \left( \prod_{k=1}^{K} w_{ik}^{\nu_{ik} - 1} \right) g(b_{ik}). \quad (13)$$

Here, $C$ is a normalization factor, and $\eta_{i,j}$, $\eta_{i,j}$, and $\nu_{ik}$ are parameters of a prior pdf for the initial probability $\pi$, transition probability $a_{ij}$, and mixture weight $w_{ik}$, respectively. $g(b_{ik})$ is the prior pdf for the normal distribution $b_{ik}(x)$ and is a normal-Wishart distribution.

Given this choice of priors, a locally optimal solution for MAP estimators can be obtained using the EM algorithm, just as in ML estimation. Let $Q(\Lambda, \Lambda)$ be the auxiliary function for ML estimation. The auxiliary function $R$ for MAP estimation is

$$R(\Lambda, \Lambda) = Q(\Lambda, \Lambda) + \log g(\Lambda). \quad (14)$$

Here, $R$ is maximized by repeating expectation and maximization; $\Lambda$ is the value of the parameter in the previous iteration. The MAP estimator in the present step can be obtained by substituting Eq. (13) into Eq. (14) and maximizing it, as follows.

$$\hat{\eta}_{i,j} = \frac{(\eta_{i,j} - 1) + \gamma_{0}}{\sum_{j=1}^{N} (\eta_{i,j} - 1) + \sum_{j=1}^{N} \gamma_{j}} \quad (15)$$

$$\hat{a}_{ij} = \frac{(\eta_{i,j} - 1) + \sum_{t=1}^{T} \hat{\xi}_{i,j,t}}{\sum_{j=1}^{N} (\eta_{i,j} - 1) + \sum_{j=1}^{N} \sum_{t=1}^{T} \hat{\xi}_{i,j,t}} \quad (16)$$

$$\hat{w}_{ik} = \frac{(\nu_{ik} - 1) + \sum_{t=1}^{T} c_{ikt}}{\sum_{k=1}^{K} (\nu_{ik} - 1) + \sum_{k=1}^{K} \sum_{t=1}^{T} c_{ik}} \quad (17)$$

$$\hat{\mu}_{ik} = \frac{\tau_{ik} \hat{\mu}_{ik} + \sum_{t=1}^{T} c_{ikt} x_{t}}{\tau_{ik} + \sum_{t=1}^{T} c_{ikt}} \quad (18)$$

$$\hat{\xi}_{ik} = \frac{1}{(\alpha_{ik} - p) + \sum_{t=1}^{T} c_{ikt}} \times \left[ \sum_{i=1}^{T} c_{ikt} (x_{t} - \hat{\mu}_{ik}) (x_{t} - \hat{\mu}_{ik})' \right] + \tau_{ik} (\hat{\mu}_{ik} - \mu_{ik}) (\hat{\mu}_{ik} - \mu_{ik})' \quad (19)$$

Here, $\gamma_{0}$ is the posterior probability of being in state $i$ at time $0$, $\hat{\xi}_{i,j,t}$ is the posterior probability of making a transition from state $i$ to state $j$ at time $t$. $\gamma_{ij}$ is the posterior probability of observing the feature vector at time $t$ in the $k$-th mixture component of state $i$. These estimated parameters are set to $\Lambda$, and this process is repeated until the auxiliary function $R$ converges.

One remaining problem is how to set the prior probabilities. In CDHMMs, the prior probabilities are set such that the MAP estimator with no observation are identical to the parameters of the initial speaker-independent CDHMMs. Note that adaptation of mean vectors is much more effective than adaptation of other parameters. Therefore, adaptation of mean vectors is often enough in most applications (e.g., [81]).

It can be easily seen from Eqs. (15)–(19) that the MAP estimator is the interpolated value between the parameter of the initial CDHMM and the ML estimator obtained from the data. When the amount of data is small, the MAP estimators are close to the parameters of the initial CDHMM. As the amount of data increases, the MAP estimator becomes asymptotically close to the ML estimator. Accordingly, MAP adaptation is robust against the data insufficiency when the amount of adaptation data is small, and its accuracy approach to that of ML estimation when the amount of adaptation data increase. It suffices the two requirements described in Sect. 2.

When the amount of data is small, however, some parameters of CDHMM have no corresponding adaptation data and remain unchanged from the initial values. Hence, the improvement of recognition accuracy is relatively smaller than other adaptation methods described later. This is because the adaptation model (CDHMM itself) in MAP adaptation has too many parameters to be adapted. For rapid adaptation, we need to prepare a smaller model as an adaptation model.

Gauvain et al. evaluated MAP adaptation by using 32-mixture monophone HMMs for the resource management task [25]. The error rate was 13.9% for speaker-independent recognition, and it decreased to 8.7% with 40 utterances. Under the same conditions, the error rate of ML estimation increased to 31.7%. The error rate of MAP estimation using 600 utterances was 3.4%, almost the same as that of ML estimation, 3.5%.

Quasi-Bayes adaptation [31], [32] is an application of MAP estimation to on-line adaptation. In this method, the posterior probability is approximated with a normal distribution in the sequential Bayes estimation scheme. The parameter estimation is carried out using the following auxiliary function.

$$R(\Lambda, \Lambda) = Q(\Lambda, \Lambda) + \rho \log g(\Lambda) \quad (20)$$

Here, the model parameter $\Lambda$ is estimated from all past samples. $\rho$ is the forgetting factor, which should be optimized for each application. While it is not necessary to memorize sufficient statistics for the past samples, the estimated parameters may not converge to the ML estimator obtained in batch training.

MAP adaptations for discrete HMMs and semi-continuous HMMs have also been studied [30].

4.4 Bayes Estimation

Bayes estimation is not a kind of MAP estimation although it is related to it (e.g., [86]). Let $p(x|\theta)$ be the output probability of a random vector $x$, and let the parameter $\theta$ have a distribution. Then, the output probability of $x$ after observing data $X$ is

$$p(x|X) = \int p(x|\theta) g(\theta|X) d\theta \quad (21)$$
In MAP estimation, we maximize the posterior probability \( g(\theta|x) \) and use it as the parameter (point estimation). In Bayes estimation, in contrast, we calculate the posterior distribution, the left-hand side of Eq. (21).

As an example, let us estimate the mean \( \mu \) of a normal distribution when the variance \( \sigma^2 \) is known. Let \( \sigma_0^2 \) be the variance of the prior distribution of \( \mu \), \( \hat{\mu} \) be the MAP estimator of \( \mu \). Then, the posterior probability of \( x \) is a normal distribution \( N(x; \hat{\mu}, \sigma_0^2 + \sigma^2) \). That is, the estimated value is the same as the MAP estimator, and the variance is the sum of the variance of the mean and the variance of the given distribution. The estimation of variances is more robust than MAP estimation when the amount of data is small. As previously mentioned, however, variance adaptation gives much smaller improvement than mean adaptation. Hence, there is not much difference in performance between Bayes adaptation and MAP adaptation when both mean and variance are adapted.

4.5 Discriminative Learning

Discriminative learning methods are effective in estimating the parameters of CDHMMs. They are especially effective when the available data is insufficient, and they should also be good for estimating the parameters of adaptation models. In fact, many adaptation methods using discriminative learning have been proposed. For example, the parameters of the adaptation model used in MLLR have been estimated by calculating the maximum mutual information (MI) [89], minimum classification error (MCE) [95], and minimum phone error (MPE) [92].

5. Adaptation Models Using Parameter Mean

Here, we shall explain adaptation models using parameter averages as prior knowledge. This kind of model may be used when relatively little data is available, that is, when the estimated value for the average of a parameter over many speakers is reliable, but that of its covariance are not.

Models belonging to this category are built by sharing some of the parameters of CDHMMs or assigning constraints to them in order to have fewer free parameters, and these free parameters are estimated from a small amount of data. Each model belongs to this category is represented by a mapping from the averaged parameters of the initial CDHMM to those of the target CDHMM. Adaptation methods using adaptation models in this category are often called transformation-based adaptation.

We shall focus on three adaptation models: shift, linear mapping, and non-linear mapping.

5.1 Shift

A shift is the difference between mean vectors before adaptation and after adaptation. The method of sharing one shift for the mean vectors of all mixture components in a CDHMM is called signal bias removal (SBR) [72]. It corresponds to parallel displacement in the parameter space. It has been used to adapt to multiplicative noise when cepstral coefficients are used as features. It is a special case of the MLLR method explained later, where \( A = I \) (I is an identity matrix). It becomes identical to ML estimation of mean vectors when a shift is provided for each mean vector.

Between SBR and ML estimation lies spectral interpolation [77] and vector field smoothing (VFS) [64]. Spectral interpolation [77] estimates the shifts for the parameters without corresponding data samples in the adaptation data by interpolating the shifts in the neighborhood in the parameter space. Its estimates asymptotically become close to ML estimators. VFS [64] applies smoothing to shifts close to each other after interpolation. The smoothing is effective when the amount of adaptation data is small. The recognition accuracy of VFS cannot reach that of ML estimation when the amount of adaptation data is sufficient. To avoid this, the degree of smoothing has to be controlled according to the amount of available data. Tonomura et al. used MAP estimation to estimate shifts in VFS [91].

Stochastic matching (SM) [75] estimates not only the shift but also its variance to improve robustness against noise. Chien et al. [7] used MAP estimation for stochastic matching (SM) [75].

5.2 Linear Mapping

Maximum likelihood linear regression (MLLR) [50] uses a linear mapping between the acoustic feature spaces of different speakers as the adaptation model. It is one of the most popular model adaptation methods since it is easy to use and performs well in most cases.

In MLLR, the mean vectors of the Gaussian distributions in the HMMs, \( \mu = (\mu_1, \ldots, \mu_n)' \) where \( n \) is the dimension of a feature vector, are updated according to the following transformation:

\[
\hat{\mu} = A\mu + b, \tag{22}
\]

Here, \( A \) is an \( n \times n \) matrix, and \( b \) is a \( n \)-dimensional vector. Equation (22) is an affine transformation for \( \mu \), but it can be rewritten into a linear mapping as follows.

\[
\hat{\mu} = W\xi, \tag{23}
\]

where \( \xi = (1, \mu_1, \ldots, \mu_n)' \). \( W \) is an \( n \times (n + 1) \) matrix whose first column is identical to \( b \).

\( W \) is calculated by using the EM algorithm to make the ML estimation. Given the sequence of feature vectors \( X = [x_1, \ldots, x_T] \), the auxiliary function becomes

\[
Q(W, \hat{W}) = K - \frac{1}{2} \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m(t) [K_m + \log |\Sigma_m| + (x_t - W\xi)'\Sigma_m^{-1}(x_t - W\xi)], \tag{24}
\]

where \( \gamma_m(t) \) is the posterior probability of being in mixture component \( m \) at time \( t \), \( K \) is a term independent from the
output probabilities, and $K_m$ is a normalization factor for mixture component $m$. Accordingly, the ML estimator $\hat{W}$ of $W$ can be obtained from Eq. (24), as follows:

$$\sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_m(t) \Sigma_m^{-1} x_m = \sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_m(t) \Sigma_m^{-1} \hat{\mu}_m^{\prime}.$$ (25)

This equation can be solved in a straightforward way when the covariance matrix for each mixture component is diagonal. Let the left-hand side of Eq. (25) be $Z$:

$$Z = \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m(t) \Sigma_m^{-1} x_m^{\prime}.$$ (26)

Furthermore, define a matrix $G(i)$ whose $(j,q)$-th element, $g_{jq}$, is

$$g_{jq} = \sum_{m=1}^{M} \gamma_m^{(m)} d_{ij}^{(m)},$$ (27)

where $v_{ij}$ is the $(i,j)$-th element of matrix $V$, $d_{ij}$ is the $(i,j)$-th element of matrix $D$, $V$ and $D$ are

$$V^{(m)} = \sum_{t=1}^{T} \gamma_m(t) \Sigma_m^{-1},$$ (28)

$$D^{(m)} = \xi_m \xi_m^{\prime}.$$ (29)

Using these equations, $\hat{W}$ is obtained as follows.

$$\hat{\mu}_m^{\prime} = G(i)^{-1} z_i^{\prime}.$$ (30)

where $\hat{\mu}_m$ is the $i$-th column vector of $\hat{W}_m$, and $z_i$ is the $i$-th column vector of $Z$.

Leggetter et al. evaluated MLLR using speech recognition with triphone HMMs in the resource management task [50]. When the error rate of speaker-independent recognition was 4.3%, MLLR decreased it to 3.3% using three utterances as adaptation data, although it had larger error rates with one or two words. However, its recognition accuracy was not much better with more data, and it did not reach the error rate of speaker-dependent recognition (1.8%). This is mainly because only one transformation matrix can not represent the diverse information contained in a large amount of data. To alleviate this problem, the mixture components of HMMs can be divided into several groups, and one transformation matrix can be shared among the components in the same group. Phone labels or distances between components are often used for grouping components. Note that data-insufficiency problem becomes serious when there are too many groups. Moreover, it becomes critical when there is very little data even if only one transformation matrix is utilized. In such case, it may be necessary to use a matrix with fewer parameters, e.g., a diagonal matrix.

Besides the above methods of transforming mean vectors, there are two major methods of transforming covariance matrices: constrained MLLR [14] and unconstrained MLLR [22]. Constrained MLLR transforms features in feature space. The covariance matrix is transformed as follows:

$$\hat{\Sigma} = \Lambda \Sigma \Lambda^{\prime}.$$ (31)

$A$ cannot be analytically calculated since it is inside the Jacobian of the variable transformation. Instead, it can be numerically calculated by using Newton’s method, etc. LU decomposition can also be used in some situations [63]. Unconstrained MLLR, on the other hand, assumes that the covariance matrix represents speaker characteristics that are different from the mean. It results in a different transformation matrix from that of the mean vector and estimates its parameters independently. Although it increases the number of parameters, unconstrained MLLR is useful in noisy environments where the variances of parameters are usually large. As in the case of MAP adaptation, adaptation of variances does not bring much improvement to MLLR. For example, Gales [23] evaluated MLLR on the Wall Street Journal (WSJ) task. Adaptation of only mean vectors reduced the error rate by 13%, but further adaptation of the mean and covariance amounted to only a 2% reduction.

Several studies have applied MLLR to on-line adaptation [8], [40]. For example, Chien et al. applied Quasi-Bayes estimation to affine mapping parameters [8].

MAP is a method to estimate parameters and MLLR provides a class of adaptation models. Using MAP estimation within the MLLR framework is thus expected to yield a larger improvement than using them independently. Digalakis et al. [15] used the mean vectors obtained by MLLR as the mean vectors of the prior distribution for MAP adaptation. Different from MLLR alone, this method performs as well as ML estimation even when the amount of data is large. On the WSJ task, it achieved higher recognition accuracies than MAP alone or MLLR alone, regardless of the amount of adaptation data. Chesta et al. [6] and Chou [9] separately proposed maximum a posteriori linear regression (MAPLR); the algorithm refines the MLLR algorithm in the same way as MAP estimation does ML estimation. That is, the problem whereby the MLLR estimate becomes unstable when the amount of data is extremely small can be solved by using an elliptically symmetric matrix variant prior, which is a natural conjugate prior for a linear mapping.

5.3 Non-linear Mapping

MLLR adaptation assumes that the mapping of mean vectors between two speakers is linear, but of course, this is not true; it should be a non-linear mapping. Several studies have attempted to extend the mapping-based adaptation framework to the more precise non-linear mapping framework [85]. Although their methods precisely estimate the mapping, they need a lot of adaptation data. Hence, they are rarely used in real applications. Padmanabhan et al. combined linear mapping and non-linear mapping to achieve robustness against the data insufficiency [68]. Mak et al. applied kernel methods to estimate an affine mapping in a higher dimension space than that of features [55]. In this method, regularization techniques were used to achieve robustness against the data insufficiency.
6. Adaptation Method Using Speaker-Phone Matrix

Here, we discuss adaptation methods that use the speaker-phone matrix obtained from training data consisting of utterances from many speakers as prior knowledge. These methods have benefited from the many large speech corpuses that have become available.

One naive method in this category is to select the closest speaker to the target speaker in the model parameter space, and to use his/her speaker-dependent model for speech recognition. Even if the adaptation data is only one phone \( p \), this method can select one speaker under the assumption that two speakers similar in phone \( p \) should be similar in the other phones. This method is not practical for two reasons. First, the variety of speakers is so large that even 100 speakers is insufficient. Second, even if the speech databases consist of speech from many speakers, the amount of data for each speaker is not usually enough to make accurate speaker-dependent models.

Recently, more sophisticated methods, which use the speaker-phone matrix more effectively, have been developed. They improve recognition accuracy by using a few utterances. We will explain some of them.

6.1 Predictive Adaptation, EMAP

Predictive adaptation calculates the model parameters of acoustic units (phones, syllables, etc) with no adaptation data from the parameters with adaptation data by using multiple linear regression (MLR) [10], [19]. Let \( \mathbf{v} \) be a vector with dimension \( m \) made by concatenating all the unobserved parameters in adaptation data, and \( \mathbf{u} \) be a vector with dimension \( n \) made by concatenating all the observed parameters in adaptation data. Then, the values of the unobserved parameters are estimated by using the following multiple linear regression:

\[
\mathbf{v} = \mathbf{A}\mathbf{u} + \mathbf{b},
\]

where \( \mathbf{A} \) is an \( m \times n \) matrix, \( \mathbf{b} \) is an \( m \)-dimensional vector. \( \mathbf{A} \) and \( \mathbf{b} \) are estimated by using utterances of many speakers recorded beforehand. This adaptation is especially effective when the vocabulary of adaptation utterances is known in advance.

EMAP (Extended MAP) [46], [84] uses the joint probability of more than one model parameter as the prior probability for MAP estimation. In speech recognition, the joint probability of each pair of mean vectors of mixture components is estimated from many speakers’ utterances [102]. A method of selecting pairs for adaptation was also proposed [76].

Predictive adaptation and EMAP require too many parameters to be estimated when it is directly applied to triphone-based CDHMMs. Therefore, these methods are usually applied to simple recognition models such as monophone models.

6.2 Speaker Clustering

Speaker clustering clusters speaker-supervectors and prepares an HMM for each resulting cluster. In the recognition phase, a few utterances from a speaker are used to identify the cluster to which he or she belongs, and the corresponding HMM is used to recognize his/her voice.

The measure of the distance between speakers is the key issue in speaker clustering. Popular measures are the Bhattacharyya distance between output probabilities [42] and the probability of generating one speaker’s data from another speaker’s model after clustering [66]. Yoshizawa et al. used sufficient statistics to measure the distance [26], [96]. Hazen et al. used a soft selection method in which a speaker model is represented by a weighted sum of more than one speaker cluster [28].

Kosaka et al. proposed a tree clustering method using the Bhattacharyya distance for speaker clustering [42] and obtained a 25% error reduction by using three words in Japanese syllable recognition. However, the error rate did not improve when the adaptation data was increased to five words. This result indicates that only a few utterances are needed for selecting a speaker cluster and that more data does not contribute to improving recognition performance.

Although gender-dependent models, in which a cluster is made for each gender, are effective, clustering inside the same gender results in little improvement. This is mainly because the data used for making a cluster model becomes smaller as the number of clusters increases. That is, there is a trade-off between the detailed representation of speaker characteristics and the amount of data to make a precise cluster model.

Speaker clustering decreases the size of recognition models without incurring a large degradation in speech recognition accuracy, and thus, it decreases the computational cost for recognition.

6.3 Eigenvoice

Kuhn et al. recently proposed the eigenvoice [43], [45] method for speaker adaptation. This name eigenvoice is in analogy to the eigenface [41] method, which employs principal component analysis for face image recognition. Eigenvoice uses principal component analysis to project a speaker-supervector to a subspace of much smaller dimension.

In the training phase, training data from a large number of speakers are prepared and a speaker-dependent model is built for each speaker. Then, for each speaker, a speaker-supervector is constructed by concatenating all the mean vectors of his/her speaker-dependent HMM. Next, principal component analysis is done on the set of speaker-supervectors, and the principal components (eigen vectors) are extracted. Each set of eigen vectors is called an eigenvoice, and it forms a subspace of much smaller dimension than that of the speaker-supervectors.
A linear mapping of a new speaker’s supervector to the subspace is estimated by using ML estimation on a small amount of his/her speech data. Let $M$ be the dimension of a speaker-supervector and $J$ be the number of its eigen vectors ($J < M$), which are expressed as:

$$e(j) = (e_1(j), \ldots, e_M(j))',$$  \hspace{1cm}  j = 1, \ldots, J.  \tag{33}$$

The speaker-supervector of a new speaker is approximated by the weighted sum of the eigen vectors as follows.

$$\mu = (\mu_1, \ldots, \mu_M)' = \sum_{j=1}^{J} w(j)e(j),  \tag{34}$$

where the weight for each eigen vector $w(j), j = 1, \ldots, J$ is ML estimated with the EM algorithm. This estimation procedure is called maximum likelihood eigen-decomposition (MLED). Let $X = \{x_1, \ldots, x_T\}$ be an input feature vector sequence. The auxiliary function to be maximized is

$$Q(W, \tilde{W}) = K - \frac{1}{2} \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m^0[\Sigma_m + \log |\Sigma_m|]$$

$$+ (x_t - \mu)' \Sigma_m^{-1} (x_t - \mu).  \tag{35}$$

The weight $w(j)$ is obtained after a simple calculation, as follows:

$$\sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m(t)[e_m(j)'] \Sigma_m^{-1} x_t$$

$$= \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m(t) \sum_{j=1}^{J} w(j)[e_m(j)'] \Sigma_m^{-1} e_m(j),$$

$$j = 1, \ldots, J.  \tag{36}$$

In triphone-based CDHMMs, the speaker-supervector is too large in dimension to be used directly. Hence, we have to carefully select the parameters to which principal component analysis is applied.

Kuhn et al. evaluated this method in an alphabet recognition task[45]. There were 26 monophones, 6 states in each monophone, the output distribution in each state was a single Gaussian distribution, and the feature vector had 18 dimensions. The speaker vector had $26 \times 6 \times 18 = 2808$ dimensions. Five eigen vectors were used for MLED. The error reduction from using four utterances was 25%. In contrast, MAP adaptation and MLLR adaptation with this small amount of data degraded recognition accuracy. These results indicate that the eigenvoice method is effective when the amount of adaptation data is extremely small. However, as the adaptation data increases, the improvement becomes smaller than that of MAP adaptation and MLLR adaptation. This method has the same merits and demerits as speaker clustering.

As previously mentioned, the dimension of a speaker-supervector is usually very large but the amount of data for each speaker used for training is usually relatively small. Many techniques have been proposed to deal with the data insufficiency problem. The original eigenvoice paper proposed to use mean vectors estimated by eigenvoice adaptation as the priors for MAP adaptation, and this proved to be effective[43]. Other approaches have used probabilistic PCA (PPCA) in eigenvoice adaptation [34], [37]–[39]. Mak et al. applied non-linear PCA using kernel methods [53], [54]. Tanji et al. explored the way to efficiently cluster the speaker-phone matrix [88].

There is an alternative approach that combines MLLR and eigenvoice [5] wherein a transformation matrix for each speaker in the training data is used to form a speaker-supervector and eigenvoice adaptation is applied to the set of speaker-supervectors.

Some other studies the the other multivariate analysis method than PCA to obtain the subspace. For example, Duchateau et al. employed non-negative matrix factorization (NMF) [16], Hahm et al. used probabilistic latent semantic analysis (PLSA) [27].

7. Structural Approach

All of the adaptation methods described until now have a fixed structure for sharing parameters in acoustic space; the effective number of free parameters is constant regardless of the amount of adaptation data. Therefore, they might not improve recognition performance when the amount of adaptation data is not within the expected range.

One solution might be to switch adaptation methods according to the amount of available data. However, it is difficult to estimate the switching point because the vocabularies of the adaptation data and the speaker characteristics vary. What is needed is a method that can handle varying amounts of adaptation data.

Several studies have tried to make a tree structure out of the parameters and make the degree of sharing the parameters vary with the amount of available data [33], [78], [79]. For example, Shinoda et al. [78] proposed the automatic model complexity control (AMCC) method. In this method, a tree of Gaussian distributions is constructed using the Kullback-Leibler pseudo distance as the distance between distributions. The root node represents the whole acoustic space, and each of its leaf nodes corresponds to a Gaussian distribution in an HMM (Fig. 4). A shift is assigned to each node in the tree structure, and it is estimated as a shared parameter among its descendant leaf nodes. In the adaptation phase, nodes with more data samples than a predetermined threshold are selected and their shifts are used to update the HMM parameters. Their method uses a coarse model when the amount of data is small, but switches to a finer model when the amount of data increases. In this way, it automatically controls the model size according to the amount of data. Note that a similar approach has been taken with MLLR [50].

One problem with this approach is how to obtain the optimal degree of parameter sharing for the given data. Many methods including AMCC use the number of adaptation data samples corresponding to each parameter for
8. Adaptive Training

Adaptation updates model parameters to fit the speaker’s acoustic features. Normalization, on the contrary, modifies the feature space to fit a prepared model. Sometimes, these two approaches can be combined (e.g., [75]). We shall discuss normalization in this section, in particular, speaker adaptive training, which is normalization for speaker differences.

8.1 Feature Compensation

Feature compensation methods, for example, cepstrum mean normalization (CMN) [2] and cepstrum normalization (VTLN) [17], do not use speaker characteristics, environmental noise, and channels as input features.

In CMN, the long-time average of the cepstrum coefficients is subtracted from the cepstrum coefficients. Influences from surrounding noise or channel variations, whose rates of change are much slower than phonetic features in speech, are removed from the features. CMN is a standard method in practical applications.

The formant frequencies in the power spectrum vary from speaker to speaker, since vocal tract lengths vary. VTLN estimates the vocal tract length of each speaker from his/her spectrum of speech data and transforms the spectrum to that of a canonical speaker. It is difficult to estimate the vocal tract length precisely, so some methods have used ML estimation (ML-VTLN) [48], [100]. These methods prepare several different-length vocal tract models and choose the model maximizing the likelihood for the speaker’s utterances. McDonough et al. approximated the warping function in VTLN by using all-pass transforms [58]. VTLN is a special case of speaker adaptive training using MLLR (explained in the next subsection) where the transformation matrix has free parameters only in its diagonal elements and their neighborhood [18], [70].

8.2 Speaker Adaptive Training

Speaker adaptive training (SAT) and related techniques are intended to provide a good initial model for model adaptation [1], [71]. If we assume that speaker adaptation is always carried out, an initial model of a canonical speaker who has the average nature of all speakers is preferable to a speaker-independent model representing the difference between phonemes and the difference between speakers.

The canonical speaker model is estimated as follows. First, an initial model is prepared and the mapping between its parameters and the parameters of the speaker-dependent model for each training speaker is estimated. Next, this mapping is used to transform speech data of each training speaker. The canonical speaker model is trained with the transformed data of many training speakers and set as the initial model for the next step. This process is repeated several times. The recognition phase estimates the mapping from the canonical model to the target speaker by using the speaker’s utterances and the mapping to adapt the model. The mapping should be carefully selected so that it can be precisely estimated even when there is only a small amount of data for each training speaker. Affine mapping as in MLLR is often used.

Anastasakos et al. [1] evaluated SAT using triphone HMMs in the WSJ task. The error rate of speaker-
independent recognition was 6.5%, and speaker adaptation using 40 utterances reduced it to 5.3%. SAT further reduced it to 4.8%. The amount of improvement was significant, albeit not large. The main contribution of SAT is to reduce the variances between speakers from the variances of the initial model, but the variances do not influence recognition performance much. This may be why SAT does not bring about a large improvement. Yu et al. reported that MAP estimation and variational Bayes estimation of SAT parameters improve recognition accuracy [98].

Cluster adaptive training (CAT) uses several models made by speaker clustering for SAT, and it simultaneously estimates the model parameters and the weight among the models [24]. When only the weight coefficients for speakers are estimated in adaptation, the number of free parameters is very small and thus has a similar tendency with eigenvoice. Yu et al. introduced discriminative learning to CAT [97]. Arindam et al. combined CAT with MLLR [56]. Tang et al. discussed the relationship between CAT and eigenvoice [87].

9. Adaptation to Noisy Speech

Since speech recognition accuracy significantly deteriorates in noisy environments, many studies have sought ways to lessen the effect of noise. The methods can be roughly classified into three categories: feature compensation, model adaptation, and missing feature theory. Many techniques have been applied to model adaptation in noisy environments. For example, Zhang et al. applied tree-structure-based adaptation [103], and Nguyen et al. applied MLLR and eigenvoice [62]. Adaptation to noisy speech is different from speaker adaptation in that noisy speech has not only convolutive factors, but also additive factors in the spectral domain. The vector Taylor series based approach [59], Jacobian adaptation [73], and their extensions (e.g., [4], [51]) have been extensively studied.

10. Conclusion

We surveyed acoustic model adaptation techniques for speech recognition using CDHMMs. Figure 5 illustrates the relationship between those from the view point of the prior knowledge and the amount of adaptation data required.

In the future, as more speech data recorded in different noisy environments and channels becomes available, we expect that the adaptation techniques using the speakerphone matrices will become especially prominent. Such techniques will include ones that can efficiently exploit transcriptions that vary largely among speakers [26], [96], those based on multivariate analysis such as eigenvoice, subspace-based methods to separate phonetic features and speaker features [61], and unified approaches for speaker and phonetic features [35], [36].

Moreover, many practical applications would benefit from seamless adaptation to changes the amount of data, such as in the structural approaches. To make this possible, a means of the statistical model selection has to be devised. Unfortunately, the HMM is a non-regular model with hidden variables, and hence, information criteria for regular models, such as Akaike’s information criteria (AIC) or the MDL criterion, cannot be directly applied. The measure of complexity for non-regular models should be investigated using statistical learning theory. Furthermore, on-line unsupervised adaptation should be developed for more applications (e.g., [82], [93]).

It is important to develop vocabulary selection methods so that less data needs to be used to produce an adequate result [11], [60].

Lastly, the analysis of speaker variety remains as an important challenge. A large amount of data from many speakers is now available, so we believe it is time to tackle this problem (e.g., [29], [44]).

References


Koichi Shinoda received the B.S. degree in 1987, M.S. degree in 1989, both in physics from the University of Tokyo, and Dr. Eng. degree in computer science from Tokyo Institute of Technology in 2001. In 1989, he joined NEC Corporation Japan, and was involved in research on automatic speech recognition. From 1997 to 1998 he was a visiting scholar with Bell Labs, Lucent Technologies, Murray Hill, NJ. From Jun. 2001 to Sep. 2001, he was a principal researcher with Multimedia Research Laboratories, NEC Corporation. From Oct. 2001 to Mar. 2003, he was an associate professor with the University of Tokyo. He is currently an associate professor with Tokyo Institute of Technology. His research interests include speech recognition, statistical pattern recognition, and human interface. Dr. Shinoda received the Awaya Prize from the Acoustic Society of Japan in 1997 and the Excellent Paper Award from the Institute of Electronics, Information, and Communication Engineers (IEICE) in 1998. He is an associate editor of Computer Speech and Language. He is a member of IEEE, ACM, ASJ, IPSJ and JSAI.