CAMBRIDGE UNIVERSITY
ENGINEERING DEPARTMENT

Discriminative Cluster Adaptive Training

K. Yu and M.J.F. Gales
CUED/F-INFENG/TR486

Cambridge University Engineering Department
Trumpington Street
Cambridge. CB2 1PZ
England

E-mail: {ky219,mjfg}@eng.cam.ac.uk

Abstract

Recently, cluster adaptive training (CAT) or eigenvoices system has attracted much interest for rapid speaker and environment adaptation. Here a *multiple-cluster* model is used as the *canonical* model. A set of interpolation weights are then used to transform the canonical model to be representative of an individual speaker or acoustic environment. Maximum Likelihood (ML) for CAT has previously been well investigated. However for state-of-the-art large vocabulary continuous speech recognition systems, discriminative training is usually employed. This paper investigates discriminative training for multiple-cluster systems. In particular, minimum phone error (MPE) training is examined. For these multiple-cluster systems, modification of the smoothing function and the prior distribution associated with MPE training are required. MPE training for interpolation weights in CAT is also investigated, which leads to a complete framework for discriminative CAT. This approach can also be viewed as discriminatively training eigenvoices systems. In practice, a simplified version of discriminative adaptive training is employed for unsupervised adaptation in which transformations are first estimated using the ML criterion and then fixed. Only the model parameters are updated in later discriminative training stage. Experiments were performed on a conversational telephone speech task. Discriminative cluster adaptive training is shown to significantly outperform ML baselines and standard discriminative training systems.

Contents

List of Figures

List of Tables

1 Introduction

Adaptive training is a widely used technique to build speech recognition systems on non-homogeneous data. The aim is to train a *canonical* model which purely represents speech variability. A set of transformations is used to represent unwanted acoustic variabilities [1]. Thus in adaptive training, there are two sets of parameters to be estimated: the canonical model and the transformation. An iterative procedure is usually employed during training, in which the canonical model and transformation parameter estimation is interleaved. One commonly used approach is based on linear transforms, such as maximum likelihood linear regression (MLLR) [3] and constrained MLLR (CMLLR) [7]. An alternative approach is cluster-based adaptive training approach. The basic idea is to construct several systems and then use a weighed sum over all systems to build an adapted standard HMM system for the target domain/speaker. Cluster adaptive training (CAT) [8] or eigenvoices [15] are two such approaches. More discussion about adaptive training in multiple-cluster systems can be found in [19]. This is the approach investigated in this paper.

In cluster adaptive training, the "transformations" used to represent the non-speech variabilities are interpolation weight vectors. The canonical model is no longer a single set of HMMs but a number of model sets. In this paper, the multiple model sets are considered together as a special kind of canonical model, referred as a *multiple-cluster model*. The covariance matrices, component priors and transition matrices in the multiple-cluster model are shared. Only mean vectors are different between the sets of models [8] [16]. This allows simple estimation of the model parameters and interpolation weights. Maximum Likelihood (ML) based theory for cluster-based adaptive training has previously been published [8].

In state-of-the-art speech recognition systems, *discriminative training* is commonly employed to obtain the best performance. Discriminative training criteria take into account competing incorrect hypothesis in training rather than only increasing the likelihood of the correct hypothesis. Thus, in addition to maximising the likelihood of the correct word sequence, it attempts to minimise the likelihood of incorrect word sequence. Commonly used criteria are Maximum Mutual Information (MMI) [17], Minimum Classification Error (MCE) [2] and Minimum Phone Error (MPE) [5]. Standard auxiliary functions can not be used for these criteria. However, by using a *weak-sense auxiliary function* [22], re-estimation formulae for HMM parameters can be obtained. These formulae are the same as the extended Baum-Welch algorithm [20]. Discriminative training has been studied in a linear transform-based adaptive training framework [11] [13] [25]. In these studies, discriminative criterion is employed to estimate both model and transformation parameters. However, due to the lack of correct hypothesis in unsupervised adaptation, a simplified discriminative adaptive training is often used. After several iterations of ML based adaptive training, transformation parameters are fixed. Only the model parameters are updated using discriminative criterion. In recognition, the ML criterion is used to estimate transformation parameters [25] using the final discriminative model. Previous studies on discriminative adaptive training were focused on standard single-cluster HMM.

In this paper, discriminative training in the framework of cluster-based adaptive training is investigated. In particular, the MPE criterion for multiple-cluster HMM training is discussed in detail. For MPE training, it is necessary to use a smoothing function to obtain good model parameters [22]. Here an appropriate form for multiple-cluster model is defined, along with an appropriate form of the I-smoothing distribution. Re-estimation formulae for multiple-cluster model parameters and interpolation weights can then be derived. The selection of the prior parameters in the I-smoothing distribution is important when using a weak-sense auxiliary function. Various prior forms may result in different generalisation ability. Due to close relationship between CAT and eigenvoices, this work can also be viewed as discriminative training for eigenvoices systems. A more complex multiple-cluster model based discriminative adaptive training technique which uses a combination of CAT and CMLLR to represent complex non-speech variabilities, referred as structured transforms (ST) [27], is also discussed.

This paper is arranged as follows. Section 2 introduces ML-based CAT and the relationship with eigenvoices. Section 3 investigates various aspects of MPE training for multiple-cluster model in detail. In section 4, re-estimation formulae for discriminatively updating interpolation weights are derived. Discriminative adaptive training with structured transforms is discussed in section 5. Section 6 describes various experiments on a conversational telephone speech task. Conclusion is drawn in Section 7.

2 Cluster Adaptive Training

Cluster Adaptive Training (CAT) [6] is a multiple-cluster HMM training approach. The basic idea is to build a target-domain-specific model by using a wighted sum of *multiple* systems. As shown in figure 1, a weight vector is used to interpolate multiple-cluster systems to construct a new HMM representing an individual acoustic factor (speaker or acoustic environment).

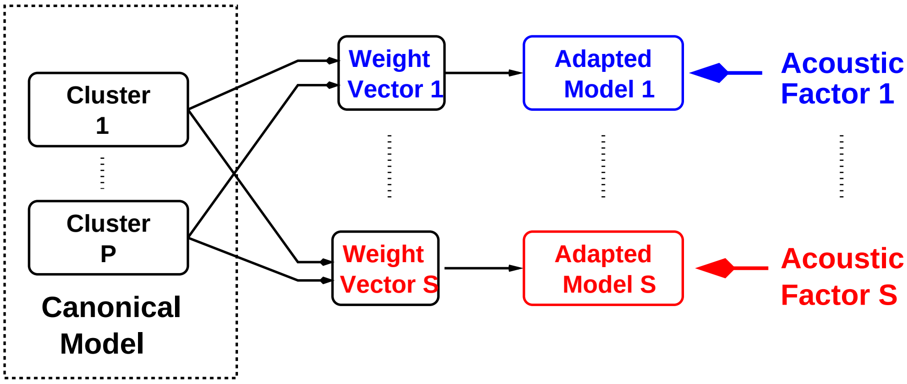

Figure 1: Illustration of cluster adaptive training

In order to simplify training, it is commonly assumed that different clusters have the same covariance matrices, transition matrices and mixture weights and only the mean values differ between the clusters. Hence only the means are adapted to a particular speaker [1]. For CAT, the adaptation is performed by interpolating the cluster means. For a particular Gaussian component m, the mean vector for speaker s, $\boldsymbol{\mu}^{(sm)}$ is given by

$$\boldsymbol{\mu}^{(sm)} = \mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)} \tag{1}$$

where $\mathbf{M}^{(m)}$ is a matrix of P cluster mean vectors,

$$\mathbf{M}^{(m)} = [\boldsymbol{\mu}_1^{(m)}, \cdots, \boldsymbol{\mu}_P^{(m)}]$$

$\boldsymbol{\mu}_p^{(m)}$ is the mean vector of component m of cluster p and the interpolation weight vector for a particular speaker s is given by:

$$\boldsymbol{\lambda}^{(s)} = [\lambda_1^{(s)}, \cdots, \lambda_P^{(s)}]^T$$

In some CAT systems, a bias cluster may be used, whose associated interpolation weight is always 1. The interpolation weight vector for a CAT system with a bias cluster is:

$$\boldsymbol{\lambda}^{(s)} = [\lambda_1^{(s)}, \cdots, \lambda_{P-1}^{(s)}, 1]^T$$

For both cases, the canonical model $\mathcal{M}$, a *multiple-cluster model*, or referred to as a *CAT model* in later sections, consists of [2]

$$\mathcal{M} = \left\{ \{\mathbf{M}^{(1)}, \cdots, \mathbf{M}^{(M)}\}, \{\boldsymbol{\Sigma}^{(1)}, \cdots, \boldsymbol{\Sigma}^{(M)}\} \right\} \tag{2}$$

[1] "Speaker" here is an example of acoustic factor, the data may be partitioned into different groups according to other acoustic factors, e.g. environment noise.

[2] In this paper, the Gaussian component weights and transition matrices are not included as their estimation, both ML-based and MPE-based, are similar to the standard estimation schemes. Refer to [12] for standard ML-based scheme and [22] for MPE-based scheme.

where M is the total number of Gaussian components and $\mathbf{\Sigma}^{(m)}$ is the covariance matrix associated with Gaussian component m.

Depending on the representation of the cluster means, there are two kinds of CAT models [8]:

- *model-based.* The clusters are represented as a distinct set of mean vectors, i.e., there is *one* distinct mean vector for each component of each cluster.
- *transform-based.* The clusters are represented by a set of cluster-specific transforms and a single standard HMM set. This representation may significantly reduces parameters to estimate.

In this work, only the *model-based* CAT model is considered. The next section describes Maximum Likelihood (ML) estimation of the CAT parameters. This is then followed by a brief discussion of how such scheme may be initialised.

2.1 ML update of model parameters and interpolation weights

Maximum Likelihood (ML) CAT training is to find the optimal multiple-cluster model and weights parameters to maximize the likelihood of observation sequence given the correct transcription. As in standard HMM training, directly optimising the criterion is hard, an expectation-maximisation (EM) algorithm [4] is again employed. In the EM framework, the auxiliary function to be optimised with respect to model parameters and weights is:

$$\begin{aligned}\mathcal{Q}(\mathcal{M},\boldsymbol{\lambda}^{(s)};\hat{\mathcal{M}},\hat{\boldsymbol{\lambda}}^{(s)}) &= -\frac{1}{2}\sum_{s,m,t}\gamma_m^{ml}(t)\Big(\mathbf{o}(t)-\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)}\Big)^T\mathbf{\Sigma}^{(m)-1}\Big(\mathbf{o}(t)-\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)}\Big)\\ &\quad -\frac{1}{2}\sum_{s,m,t}\gamma_m^{ml}(t)\log|\mathbf{\Sigma}^{(m)}| \qquad (3)\end{aligned}$$

where $\gamma_m^{ml}(t)$ is the posterior probability of Gaussian component m at time instance t given an observation sequence. It is calculated using forward-backward algorithm based on the current parameters $\hat{\mathcal{M}},\hat{\boldsymbol{\lambda}}^{(s)}$ and correct transcriptions [12]. $\boldsymbol{\lambda}^{(s)}$ is the interpolation weight vector for speaker s.

By differentiating equation (3) with respect to CAT model parameters and equating to zero, ML-based re-estimation formulae for CAT model can be derived [8]. If diagonal covariance matrices are used [3], it is only necessary to store diagonal elements of $\mathbf{L}_{ML}^{(m)}$. The mean and covariance matrix for this case can be updated by:

$$\mathbf{M}^{(m)T} = \mathbf{G}_{ML}^{(m)-1}\mathbf{K}_{ML}^{(m)} \qquad (4)$$

$$\mathbf{\Sigma}^{(m)} = \mathrm{diag}\Big(\frac{\mathbf{L}_{ML}^{(m)}-\mathbf{M}^{(m)}\mathbf{K}_{ML}^{(m)}}{\gamma_m^{ml}}\Big) \qquad (5)$$

where the sufficient statistics are

$$\gamma_m^{ml} = \sum_{s,t}\gamma_m^{ml}(t) \qquad (6)$$

$$\mathbf{G}_{ML}^{(m)} = \sum_{s,t}\gamma_m^{ml}(t)\boldsymbol{\lambda}^{(s)}\boldsymbol{\lambda}^{(s)T} \qquad (7)$$

$$\mathbf{K}_{ML}^{(m)} = \sum_{s,t}\gamma_m^{ml}(t)\boldsymbol{\lambda}^{(s)}\mathbf{o}(t)^T \qquad (8)$$

$$\mathbf{L}_{ML}^{(m)} = \sum_{s,t}\gamma_m^{ml}(t)\mathbf{o}(t)\mathbf{o}(t)^T \qquad (9)$$

[3]This assumption is also applicable to discriminative cluster adaptive training in later sections

To update the interpolation weights given a multiple-cluster model, equation (3) is differentiated with respect to $\boldsymbol{\lambda}^{(s)}$ and equated to zero. The cluster weights vector for speaker s can be estimated as

$$\boldsymbol{\lambda}^{(s)} = \mathbf{G}_{ML}^{(s)-1}\mathbf{k}_{ML}^{(s)} \tag{10}$$

where the sufficient statistics are

$$\mathbf{G}_{ML}^{(s)} = \sum_m \left(\sum_t \gamma_m^{ml}(t)\right) \mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)} \tag{11}$$

$$\mathbf{k}_{ML}^{(s)} = \sum_m \mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\left(\sum_t \gamma_m^{ml}(t)\mathbf{o}(t)\right) \tag{12}$$

As the amount of training data for a particular speaker s may vary, multiple interpolation weight vectors can be used to enhance flexibility. Each weight vector is associated with a group of Gaussian components. The estimation of the weight vector only involve Gaussian components in that group, the grouping of Gaussians can be done by using regression tree [18] [8]. An iterative re-estimation procedure is adopted to train CAT model and weights parameters [8]. This approach allows large state-of-the-art systems to be trained.

1. Initialise interpolation weights $\boldsymbol{\lambda}_0^{(s)}$.
2. Use a single-cluster speaker independent (SI) model $\mathcal{M}^{SI}$ to obtain the component posterior occupancy $\gamma_m^{ml}(t)$. With $\boldsymbol{\lambda}_0^{(s)}$ and $\gamma_m^{ml}(t)$, an initial multiple-cluster model $\mathcal{M}_0$ can be estimated. Set the iteration index $k = 0$.
3. $k = k + 1$
4. Estimate $\boldsymbol{\lambda}_k^{(s)}$ given $\left\{\mathcal{M}_{k-1}, \boldsymbol{\lambda}_{k-1}^{(s)}\right\}$
5. Estimate $\mathcal{M}_k$ given $\left\{\mathcal{M}_{k-1}, \boldsymbol{\lambda}_k^{(s)}\right\}$
6. Go to 3 until the convergence criterion satisfied.

During recognition, interpolation weights are required for each test set speaker before decoding with the CAT model. From the sufficient statistics $\mathbf{G}_{ML}^{(s)}$ and $\mathbf{k}_{ML}^{(s)}$, to initialise weights for test speakers, what is required is only the multiple-cluster model and $\gamma_m^{ml}(t)$ obtained from a standard HMM, e.g. SI model. Then further weights estimation can be performed in a similar iterative fashion as in training. The adaptation procedure is illustrated in Figure 2.

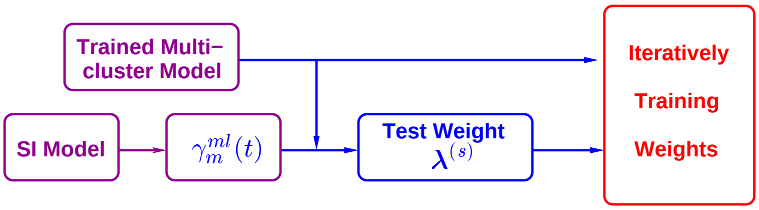

Figure 2: Adaptation of CAT system

Note that, no special initialisation approach or prior knowledge for interpolation weight vectors is required in adaptation. They can be initialised naturally. The overall test adaptation and recognition procedure is shown below. This procedure is similar to the adaptation for standard SAT training [1].

1. Use a SI model $\mathcal{M}^{SI}$ to get the component posterior occupancy for adaptation $\gamma_m^{ml}(t)$.
2. Use $\gamma_m^{ml}(t)$ and the pre-trained CAT model $\mathcal{M}$ to get the initial weights $\boldsymbol{\lambda}_0^{(s)}$. Set $k = 0$.
3. $k = k + 1$
4. Estimate $\boldsymbol{\lambda}_k^{(s)}$ given $\left\{\mathcal{M}, \boldsymbol{\lambda}_{k-1}^{(s)}\right\}$.
5. Go to 3 until the convergence criterion is satisfied.
6. Viterbi decoding with $\left\{\mathcal{M}, \boldsymbol{\lambda}_k^{(s)}\right\}$.

An alternative to the "soft" weights described above is to use a "hard" weight vector selecting only one of the clusters. If this hard-assign weight vector is used, i.e., only one element is 1, all the other are 0, the re-estimation is equivalent to update mean vector of each cluster independently, which is similar to training a set of standard cluster-dependent HMMs except for sharing covariance, transition and mixture weights.

2.2 Initialisation in CAT training

When training a model using the EM algorithm, initialisation is an important issue. It is possible to either initialise the interpolation weights first and then the multiple-cluster model, or initialise multiple-cluster model first and then the interpolation weights. In both schemes, the posterior probability of each Gaussian component at time t, $\gamma_m^{ml}(t)$, is required. This is normally obtained from an standard speaker-independent (SI) model [8].

For state-of-the-art systems, there is often a large amount of model parameters. Directly constructing multiple-cluster models is expensive. Thus in practice, interpolation weights are often initialised first. Note that, according to the sufficient statistics (6) to (9), having obtained initial weights and using a standard SI model to get $\gamma_m^{ml}(t)$, an initial CAT model can then be constructed and iterative training may continue. The procedure is illustrated in Figure 3

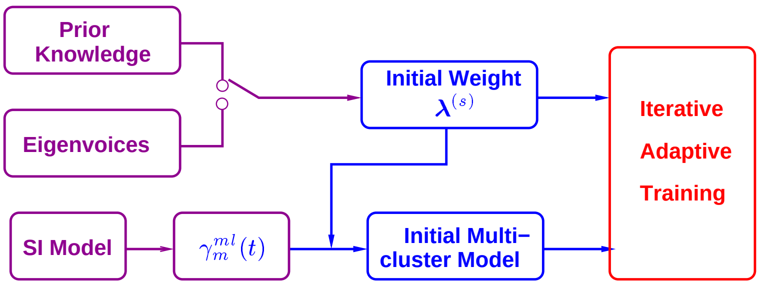

Figure 3: Initialisation of CAT system

There are two schemes for the weights initialisation.

- **Clustering based approach**. If some prior knowledge related to speaker clustering is available, the training data of each speaker can be assigned an initial weight vector based on the knowledge. Each element of the vector is a 0/1 value, according to which cluster it belongs to. For example, with gender information, an initial 2-cluster weight vector, $[0, 1]$ or $[1, 0]$ can be assigned to male and female speaker respectively. If there is no prior cluster knowledge available, standard speaker clustering approaches may be used [14].
- **Eigen-decomposition approach**. Similar to speaker clustering, a number of simple speaker-dependent models are trained first. Then the means of each model are concatenated to form a single meta-vector, i.e., one meta-vector for each speaker. Principal component

analysis (PCA) or linear discriminant analysis (LDA) is then performed on the set of meta-vectors. The orthogonal eigen-meta-vectors, or *eigenvoices*, with the largest eigenvalues are selected as a basis set. The number of the selected eigenvoices will give the number of clusters. Given the basis eigenvoices and the meta-vectors for each speaker, initial weights for each speaker can be obtained using either projection scheme or ML-based approach [8] [15]. This approach will naturally output a bias cluster, i.e. weight value for this cluster is 1.0, which is the mean of all meta-vectors. By using this approach, arbitrary number of clusters can be initialised without knowing prior knowledge.

Both CAT [6] and eigenvoices [16] approaches are cluster-based adaptive training schemes. It is interesting to contrast the two schemes. For an eigenvoices system, a set of distinct eigen mean vectors are used for the canonical model and the other model parameters are shared [15]. This is equivalent to a *model-based* CAT system. The "eigenvoices" (or basis vectors of the eigenspace) correspond to the cluster mean matrices in CAT model. The number of "eigenvoices" is equivalent to the number of clusters. In initialisation, eigenvoices systems always employ an eigen-decomposition approach based on PCA or LDA. But CAT systems may also employ prior knowledge for initialisation. For some eigenvoices systems, the initialised basis eigenvoices are not further updated but directly used in adaptation and decoding [15] while CAT systems always update the multiple-cluster model [8]. If eigenvoices are updated using Maximum Likelihood EigenSpace (MLES) approach [21], it is equivalent to updating CAT cluster mean matrices with covariance matrices unchanged [8].

To summarise, an ML-based eigenvoices system is a simplified version of model-based CAT where the model (eigenvoices) update and the interpolation weights estimation are the same. Due to this close-relationship, the discriminative training for CAT in this paper can also be naturally used in eigenvoices systems.

3 MPE Training of Multiple-cluster Model Parameters

ML criterion aims to construct a model maximising the likelihood of training data. However, the essential purpose of model training is to reduce the word error rate. People are interested in finding more appropriate criteria. Therefore, the discriminative training criteria become increasingly popular [24]. The criteria take into account competing, incorrect hypothesis in training rather than only the correct hypothesis. Thus, in principle, it not only maximizes the likelihood of observation sequences given the correct hypothesis, but also minimizes the likelihood of observation sequences given incorrect hypothesis. Consequently, the training may result in a model with more word discriminant ability. Various discriminative criteria are possible, such as maximum mutual information (MMI) [17] and minimum classification error (MCE) [2]. Minimum phone error (MPE) has been recently introduced and been found to yield good performance [5]. In this section, MPE criterion will be applied to multiple-cluster model training.

3.1 Minimum phone error (MPE) training

The MPE criterion may be expressed as below [5]

$$\mathcal{F}(\mathcal{M}) = \log\left(\frac{\sum_w p(\mathbf{O}|\mathcal{M}_w)^\kappa P(w)\mathrm{RawAccuracy}(w)}{\sum_w p(\mathbf{O}|\mathcal{M}_w)^\kappa P(w)}\right) \tag{13}$$

where $\mathrm{RawAccuracy}(w)$ is a measure of the number of phones accurately transcribed, $\mathcal{M}_w$ is the composite model for word sequence w and κ is an acoustic deweighting factor commonly used in discriminative training.

In ML training, to find the optimal model parameters, the standard EM algorithm requires an auxiliary function satisfying [4]:

$$\mathcal{Q}(\mathcal{M};\hat{\mathcal{M}}) - \mathcal{Q}(\hat{\mathcal{M}};\hat{\mathcal{M}}) \leq \mathcal{F}(\mathcal{M}) - \mathcal{F}(\hat{\mathcal{M}}) \tag{14}$$

where $\mathcal{F}$ and $\mathcal{Q}$ are criterion and auxiliary function respectively, $\hat{\mathcal{M}}$ and $\mathcal{M}$ denote the current and new model parameters respectively. This standard auxiliary function will be referred as a *strong-sense* auxiliary function [22]. It is a strict lower bound of the criterion. An increase of the auxiliary function $\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}})$ guarantees to increase the criterion $\mathcal{F}(\mathcal{M})$, unless the parameters are at a local minimum. The model parameters can be estimated by iteratively optimising the auxiliary function. In ML training, the auxiliary function can be found by applying Jensen inequality to the ML objective function [12]. Equation (3) is an example of a strong-sense auxiliary function for ML-CAT training.

In MPE training, it is hard to find a strong-sense auxiliary function due to the denominator term in the criterion. To allow the discriminative criterion to be optimised, a *weak-sense* auxiliary function is introduced [22]. A *weak-sense auxiliary function* for $\mathcal{F}(\mathcal{M})$ around the current model parameters $\hat{\mathcal{M}}$ is a smoothing function $\mathcal{Q}(\mathcal{M}; \hat{\mathcal{M}})$ such that:

$$\left.\frac{\partial \mathcal{Q}(\mathcal{M}; \hat{\mathcal{M}})}{\partial \mathcal{M}}\right|_{\mathcal{M}=\hat{\mathcal{M}}} = \left.\frac{\partial \mathcal{F}(\mathcal{M})}{\partial \mathcal{M}}\right|_{\mathcal{M}=\hat{\mathcal{M}}} \tag{15}$$

The gradients of the two functions are the same at the point $\mathcal{M} = \hat{\mathcal{M}}$. Maximizing the function $\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}})$ with respect to $\mathcal{M}$ does not guarantee an increase of $\mathcal{F}(\mathcal{M})$. However, if $\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}})$ reaches a local maximum at $\hat{\mathcal{M}}$, i.e., the gradient is 0 at that point, $\mathcal{F}(\mathcal{M})$ is guaranteed to also be at a local maximum. Figure 4 illustrates comparison between a *weak-sense* and a *strong-sense* auxiliary function.

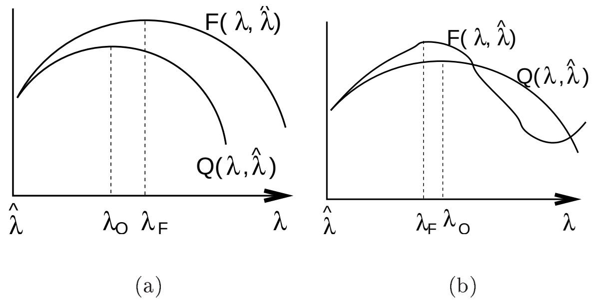

Figure 4: Use of (a) strong-sense and (b) weak-sense auxiliary functions for function optimisation

In this section, the MPE training for standard single-cluster HMM is reviewed. Definition of the single-cluster weak-sense auxiliary function is reviewed in detail for comparison to the multiple-cluster version in later sections. A weak-sense auxiliary function for standard single-cluster HMM can be expressed as [4] [22]

$$\mathcal{Q}(\mathcal{M}) = \mathcal{Q}^n(\mathcal{M}) - \mathcal{Q}^d(\mathcal{M}) \tag{16}$$

where the numerator auxiliary function, $\mathcal{Q}^n(\mathcal{M})$, has a similar form as the standard ML auxiliary function [5]:

$$\begin{aligned}\mathcal{Q}^n(\mathcal{M}) \quad = \quad & -\frac{1}{2}\sum_{m,t}\gamma_m^n(t)\left(\mathbf{o}(t) - \boldsymbol{\mu}^{(m)}\right)^T \boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{o}(t) - \boldsymbol{\mu}^{(m)}\right) \\ & -\frac{1}{2}\sum_{m,t}\gamma_m^n(t)\log|\boldsymbol{\Sigma}^{(m)}|\end{aligned} \tag{17}$$

[4]In later sections, $\mathcal{Q}(\mathcal{M})$ is used to denote $\mathcal{Q}(\mathcal{M}; \hat{\mathcal{M}})$ for brief notation.

[5]In this work, discriminative update of component weights and transition matrices are not discussed thus no weak-sense auxiliary function is defined for them.

The posterior probability for Gaussian component m at time t, $\gamma_m^n(t)$, is not calculated based on the correct transcription as the standard forward-backward algorithm does [12]. Instead, $\gamma_m^n(t)$ is calculated based on the hypothesis classified as "correct" by a phone-accuracy measure [22]. The denominator auxiliary function $\mathcal{Q}^d(\mathcal{M})$ has a similar form to the numerator auxiliary function. However, the posterior occupancy $\gamma_m^d(t)$ is based on the hypothesis classified as "incorrect" [22].

Though an auxiliary function is not bound to be convex by definition, an auxiliary function which is not convex is unlikely to yield good convergence [22]. To ensure a convex weak-sense auxiliary function and consequently improve stability in optimisation, a smoothing function, $\mathcal{Q}^{sm}(\mathcal{M})$, is added to the auxiliary function (16). This function must satisfy the following constraint

$$\frac{\partial}{\partial \mathcal{M}} \mathcal{Q}^{sm}(\mathcal{M})\bigg|_{\hat{\mathcal{M}}} = 0 \tag{18}$$

to ensure the resulting auxiliary function is still a valid weak-sense auxiliary function. One form of smoothing function is defined in [22]. Using matrix/vector notation, it can be expressed as

$$\begin{aligned} \mathcal{Q}^{sm}(\mathcal{M}) \quad = \quad & \sum_m -\frac{D_m}{2}\Bigg(\log|\boldsymbol{\Sigma}^{(m)}| + \mathrm{tr}\Big((\hat{\boldsymbol{\Sigma}}^{(m)} + \hat{\boldsymbol{\mu}}^{(m)}\hat{\boldsymbol{\mu}}^{(m)T})\boldsymbol{\Sigma}^{(m)-1}\Big) \\ & - 2\boldsymbol{\mu}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\hat{\boldsymbol{\mu}}^{(m)} + \boldsymbol{\mu}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\boldsymbol{\mu}^{(m)}\Bigg) \end{aligned} \tag{19}$$

where $\hat{\boldsymbol{\Sigma}}^{(m)}$ and $\hat{\boldsymbol{\mu}}^{(m)}$ are covariance matrix and mean vector of the current model, D_m is a component-specific smoothing constant to ensure convergence and tr() is trace of a matrix. This is guaranteed to satisfy equation (18).

For MPE training, model parameters tend to be over-trained for large vocabulary continuous speech recognition system [22]. Therefore, it is essential to perform some additional smoothing to improve the generalisation ability of the resultant model. To achieve this, a prior distribution over the model parameters, $p(\mathcal{M})$, is added to the MPE criterion (13). This is called "I-smoothing" [5]. Then the complete MPE criterion becomes

$$\mathcal{F}(\mathcal{M}) = \log\left(\frac{\sum_w p(\mathbf{O}|\mathcal{M}_w)^\kappa P(w)\mathrm{RawAccuracy}(w)}{\sum_w p(\mathbf{O}|\mathcal{M}_w)^\kappa P(w)}\right) + \log p(\mathcal{M}) \tag{20}$$

and the associated weak-sense auxiliary function becomes

$$\mathcal{Q}(\mathcal{M}) = \mathcal{Q}^n(\mathcal{M}) - \mathcal{Q}^d(\mathcal{M}) + \mathcal{Q}^{sm}(\mathcal{M}) + \log p(\mathcal{M}) \tag{21}$$

Since any function is both a weak and strong sense auxiliary function of itself, the above equation (21) is a valid weak-sense auxiliary function for equation (20). One commonly used distribution for model parameters is the Normal-Wishart distribution employed for maximum a posterior (MAP) training in [10]. The prior distribution also has a similar form as the standard auxiliary function

$$\begin{aligned} \log p(\mathcal{M}) \quad = \quad & K - \frac{\tau^I}{2}\sum_m \Bigg(\log|\boldsymbol{\Sigma}^{(m)}| + \mathrm{tr}\Big(\tilde{\boldsymbol{\Sigma}}^{(m)}\boldsymbol{\Sigma}^{(m)-1}\Big) \\ & + \Big(\boldsymbol{\mu}^{(m)} - \tilde{\boldsymbol{\mu}}^{(m)}\Big)^T \boldsymbol{\Sigma}^{(m)-1}\Big(\boldsymbol{\mu}^{(m)} - \tilde{\boldsymbol{\mu}}^{(m)}\Big)\Bigg) \end{aligned} \tag{22}$$

where τ^I is the specified parameter of the Normal-Wishart distribution which controls the impact of the prior, $\tilde{\boldsymbol{\mu}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$ are the prior parameters of the distribution, K is a normalisation term. The parameters $\tilde{\boldsymbol{\mu}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$ for I-smoothing distribution can be either based on ML statistics, which is the standard form [26], or based on MAP estimates, which is called MPE-MAP [23].

Differentiating the weak-sense auxiliary function (21) with respect to standard model parameters, close-form solution for standard MPE training can be derived [22]. The formulae have the same form as the Extended Baum-Welch algorithm [20].

3.2 Weak-sense auxiliary function for multiple-cluster model

The weak-sense auxiliary function (21) is a general form and can be used for multiple-cluster model. For convenience of derivation, the weak-sense auxiliary function is equivalently expressed in terms of statistics. From equation (3), the ML auxiliary function for CAT is given by

$$\begin{aligned}\mathcal{Q}^{ML}(\mathcal{M}) &= -\frac{1}{2}\sum_{s,m,t}\gamma_m^{ml}(t)\left(\mathbf{o}(t)-\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)}\right)^T\boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{o}(t)-\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)}\right)\\ &\quad -\frac{1}{2}\sum_{s,m,t}\gamma_m^{ml}(t)\log|\boldsymbol{\Sigma}^{(m)}| \end{aligned} \tag{23}$$

The sufficient statistics for the auxiliary function are given in equations (6) to (9). This set of sufficient statistics will be denoted as

$$\boldsymbol{\Theta}^{(ML)} = \left\{\gamma_m^{ml}, \mathbf{G}_{ML}^{(m)}, \mathbf{K}_{ML}^{(m)}, \mathbf{L}_{ML}^{(m)}\right\} \tag{24}$$

The auxiliary function $\mathcal{Q}^{ML}(\mathcal{M})$ can be equivalently rewritten as $\mathcal{G}(\boldsymbol{\Theta}^{(ML)})$ in terms of the sufficient statistics $\boldsymbol{\Theta}^{(ML)}$ and the current model parameters [6]. This expression of auxiliary function is denoted as

$$\begin{aligned}\mathcal{G}(\boldsymbol{\Theta}^{(ML)}) &= -\frac{1}{2}\sum_{m}\left\{\gamma_m^{ml}\log|\boldsymbol{\Sigma}^{(m)}| + \mathrm{tr}\left(\mathbf{L}_{ML}^{(m)}\boldsymbol{\Sigma}^{(m)-1}\right)\right.\\ &\quad \left. - 2\mathrm{tr}\left(\mathbf{K}_{ML}^{(m)}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)}\right) + \mathrm{tr}\left(\mathbf{G}_{ML}^{(m)}\mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)}\right)\right\} \end{aligned} \tag{25}$$

Similarly, the weak-sense auxiliary function, equation (21), may be expressed in term of sufficient statistics as below

$$\mathcal{Q}(\mathcal{M}) = \mathcal{G}(\boldsymbol{\Theta}) = \mathcal{G}(\boldsymbol{\Theta}^{(n)}) - \mathcal{G}(\boldsymbol{\Theta}^{(d)}) + \mathcal{G}(\boldsymbol{\Theta}^{(sm)}) + \mathcal{G}(\boldsymbol{\Theta}^{(p)}) \tag{26}$$

where $\mathcal{G}()$ is the statistics-based auxiliary function defined in equation (25) and is equivalent to corresponding auxiliary functions $\mathcal{Q}()$, $\boldsymbol{\Theta}^{(n)}$, $\boldsymbol{\Theta}^{(d)}$, $\boldsymbol{\Theta}^{(sm)}$ and $\boldsymbol{\Theta}^{(p)}$ are the sufficient statistics sets for the numerator, denominator, smoothing function and I-smoothing prior distribution respectively.

Note that the four parts have the same form of the statistics-based auxiliary function $\mathcal{G}$ with different sufficient statistics. As $\mathcal{G}$ is a linear function with respect to corresponding sufficient statistics, the whole statistics $\boldsymbol{\Theta}$ is a simple linear combination of the four individual sets of sufficient statistics. The main purpose of the section is to define appropriate forms for the four individual multiple-cluster auxiliary function and derive corresponding statistics. By finally combining the statistics, re-estimation formulae can be easily obtained.

3.2.1 Numerator and denominator functions

The numerator part has similar form to the ML multiple-cluster auxiliary function, equation (3), except for using $\gamma_m^n(t)$ to replace $\gamma_m^{ml}(t)$ as mentioned before. It can be expressed as

$$\begin{aligned}\mathcal{Q}^{n}(\mathcal{M}) &= -\frac{1}{2}\sum_{s,m,t}\gamma_m^{n}(t)\left(\mathbf{o}(t)-\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)}\right)^T\boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{o}(t)-\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)}\right)\\ &\quad -\frac{1}{2}\sum_{s,m,t}\gamma_m^{n}(t)\log|\boldsymbol{\Sigma}^{(m)}| \end{aligned} \tag{27}$$

[6] Note, the current model parameters and interpolation weights are omitted for brief notation. They are used to get component alignment in the sufficient statistics.

The sufficient statistics $\boldsymbol{\Theta}^{(n)} = \left\{\gamma_m^n, \mathbf{G}_n^{(m)}, \mathbf{K}_n^{(m)}, \mathbf{L}_n^{(m)}\right\}$ also have similar form to ML statistics $\boldsymbol{\Theta}^{(ML)}$ except for using $\gamma_m^n(t)$ instead of $\gamma_m^{ml}(t)$

$$\gamma_m^n = \sum_{s,t} \gamma_m^n(t) \tag{28}$$

$$\mathbf{G}_n^{(m)} = \sum_{s,t} \gamma_m^n(t)\boldsymbol{\lambda}^{(s)}\boldsymbol{\lambda}^{(s)T} \tag{29}$$

$$\mathbf{K}_n^{(m)} = \sum_{s,t} \gamma_m^n(t)\boldsymbol{\lambda}^{(s)}\mathbf{o}(t)^T \tag{30}$$

$$\mathbf{L}_n^{(m)} = \sum_{s,t} \gamma_m^n(t)\mathbf{o}(t)\mathbf{o}(t)^T \tag{31}$$

The denominator statistics $\boldsymbol{\Theta}^{(d)} = \left\{\gamma_m^d, \mathbf{G}_d^{(m)}, \mathbf{K}_d^{(m)}, \mathbf{L}_d^{(m)}\right\}$ are the same as numerator statistics except for using $\gamma_m^d(t)$ instead of $\gamma_m^n(t)$.

3.2.2 Smoothing function

As discussed in section 3.1, the smoothing function $\mathcal{Q}^{sm}(\mathcal{M})$ must yield the current *multiple-cluster* model parameters as the ML estimate. As the adapted mean vector of each Gaussian component is a speaker-specific weighted sum of multiple-cluster means. $\mathcal{Q}^{sm}(\mathcal{M})$ must be defined at speaker level to satisfy the constraint (18). One suitable smoothing function is given by

$$\begin{aligned}\mathcal{Q}^{sm}(\mathcal{M}) = & -\sum_{m,s} \frac{D_m \nu_m^{(s)}}{2}\Big(\log|\boldsymbol{\Sigma}^{(m)}| + \mathrm{tr}\Big((\hat{\boldsymbol{\Sigma}}^{(m)} + \hat{\boldsymbol{\mu}}^{(sm)}\hat{\boldsymbol{\mu}}^{(sm)T})\boldsymbol{\Sigma}^{(m)-1}\Big) \\ & -2\boldsymbol{\lambda}^{(s)T}\mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\hat{\boldsymbol{\mu}}^{(sm)} + \boldsymbol{\lambda}^{(s)T}\mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)}\Big)\end{aligned} \tag{32}$$

where $\hat{\boldsymbol{\mu}}^{(sm)} = \hat{\mathbf{M}}^{(m)}\hat{\boldsymbol{\lambda}}^{(s)}$, $\hat{\boldsymbol{\Sigma}}^{(m)}$ are the current model parameters [7]. The constant D_m is a positive smoothing constant for component m to ensure convergence. The smoothing function can be shown as a valid smoothing function for both multiple-cluster model parameters and interpolation weights for all values of $\nu_m^{(s)}$ (see appendix 8.1). However it is sensible to use this value to reflect the proportions of data for that particular components of a speaker, so in this work

$$\nu_m^{(s)} = \frac{\sum_t \gamma_m^n(t)}{\sum_{s,t} \gamma_m^n(t)} \tag{33}$$

where the summation in the numerator only involves data associated with speaker s.

Given the above definition, $\mathcal{Q}^{sm}(\mathcal{M})$ can also be expressed as $\mathcal{G}(\boldsymbol{\Theta}^{(sm)})$, where $\boldsymbol{\Theta}^{(sm)}$ is the set of sufficient statistics for smoothing function

$$\boldsymbol{\Theta}^{(sm)} = \left\{D_m, D_m\mathbf{G}_D^{(m)}, D_m\mathbf{K}_D^{(m)}, D_m\mathbf{L}_D^{(m)}\right\} \tag{34}$$

where

$$\mathbf{G}_D^{(m)} = \sum_s \nu_m^{(s)}\boldsymbol{\lambda}^{(s)}\boldsymbol{\lambda}^{(s)T} \tag{35}$$

$$\mathbf{K}_D^{(m)} = \mathbf{G}_D^{(m)}\hat{\mathbf{M}}^{(m)T} \tag{36}$$

$$\mathbf{L}_D^{(m)} = \hat{\boldsymbol{\Sigma}}^{(m)} + \hat{\mathbf{M}}^{(m)}\mathbf{G}_D^{(m)}\hat{\mathbf{M}}^{(m)T} \tag{37}$$

For multiple-cluster MPE training, selection of D_m is different from the single-cluster MPE training. This will be discussed in section 3.4.

[7] Note when updating model parameters, $\hat{\boldsymbol{\lambda}}^{(s)} = \boldsymbol{\lambda}^{(s)}$

3.2.3 I-smoothing distribution

A speaker-level Normal-Wishart distribution is defined as the I-smoothing distribution for a multiple-cluster model parameters. This has a similar form to the standard I-smoothing distribution for single-cluster model. With appropriate Normal-Wishart parameters, the general form of log distribution for multiple-cluster model parameters may be written as

$$\begin{aligned}\log p(\mathcal{M}) \quad &= \quad K - \frac{\tau^I}{2}\sum_{s,m}\tilde{\nu}_m^{(s)}\Bigg(\log|\boldsymbol{\Sigma}^{(m)}| + \mathrm{tr}(\tilde{\boldsymbol{\Sigma}}^{(m)}\boldsymbol{\Sigma}^{(m)-1}) \\ &\quad + \left(\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)} - \tilde{\boldsymbol{\mu}}^{(sm)}\right)^T\boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)} - \tilde{\boldsymbol{\mu}}^{(sm)}\right)\Bigg) \end{aligned} \tag{38}$$

where τ^I is the parameter of the Normal-Wishart distribution. K is the appropriate normalisation term. $\tilde{\boldsymbol{\mu}}^{(sm)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$ are prior parameters for mean and covariance of the m^{th} component for speaker s in the distribution. $\tilde{\nu}_m^{(s)}$ is a slightly modified version of equation (33). Since in I-smoothing distribution, ML estimates are often used as the prior parameters, the standard ML posterior occupancy, $\gamma_m^{ml}(t)$, are used to define $\tilde{\nu}_m^{(s)}$ instead of the numerator occupancy $\gamma_m^n(t)$. The I-smoothing distribution can also be expressed as statistics based auxiliary function $\mathcal{G}(\boldsymbol{\Theta}^{(p)})$, where the sufficient statistics are

$$\boldsymbol{\Theta}^{(p)} = \left\{\tau^I, \tau^I\tilde{\mathbf{G}}^{(m)}, \tau^I\tilde{\mathbf{K}}^{(m)}, \tau^I\tilde{\mathbf{L}}^{(m)}\right\} \tag{39}$$

where

$$\tilde{\mathbf{G}}^{(m)} = \sum_s \tilde{\nu}_m^{(s)}\boldsymbol{\lambda}^{(s)}\boldsymbol{\lambda}^{(s)T} \tag{40}$$

$$\tilde{\mathbf{K}}^{(m)} = \sum_s \tilde{\nu}_m^{(s)}\boldsymbol{\lambda}^{(s)}\tilde{\boldsymbol{\mu}}^{(sm)T} \tag{41}$$

$$\tilde{\mathbf{L}}^{(m)} = \left(\sum_s \tilde{\nu}_m^{(s)}\tilde{\boldsymbol{\mu}}^{(sm)}\tilde{\boldsymbol{\mu}}^{(sm)T}\right) + \tilde{\boldsymbol{\Sigma}}^{(m)} \tag{42}$$

The selection of prior parameter $\tilde{\boldsymbol{\mu}}^{(sm)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$ is essential for I-smoothing. In multiple-cluster MPE training, there are more options than standard single-cluster MPE training. The selection will be discussed in section 3.3.

3.2.4 Re-estimation formulae of multiple-cluster model parameters

Having specified the form of the smoothing function and I-smoothing distribution, differentiating the weak-sense auxiliary function (26) with respect to multiple-cluster model parameters and equating it to zero yields a simple close-form estimate for the multiple-cluster model parameters.

$$\mathbf{M}^{(m)T} = \mathbf{G}^{(m)-1}\mathbf{K}^{(m)} \tag{43}$$

$$\boldsymbol{\Sigma}^{(m)} = \mathrm{diag}\left(\frac{\mathbf{L}^{(m)} - \mathbf{M}^{(m)}\mathbf{K}^{(m)}}{\gamma_m}\right) \tag{44}$$

where the sufficient statistics $\boldsymbol{\Theta}$ is a linear combination of the individual statistics sets $\boldsymbol{\Theta}^{(n)}$, $\boldsymbol{\Theta}^{(d)}$, $\boldsymbol{\Theta}^{(sm)}$ and $\boldsymbol{\Theta}^{(p)}$ as below

$$\gamma_m = \gamma_m^n - \gamma_m^d + D_m + \tau^I \tag{45}$$

$$\mathbf{G}^{(m)} = \mathbf{G}_n^{(m)} - \mathbf{G}_d^{(m)} + D_m\mathbf{G}_D^{(m)} + \tau^I\tilde{\mathbf{G}}^{(m)} \tag{46}$$

$$\mathbf{K}^{(m)} = \mathbf{K}_n^{(m)} - \mathbf{K}_d^{(m)} + D_m\mathbf{K}_D^{(m)} + \tau^I\tilde{\mathbf{K}}^{(m)} \tag{47}$$

$$\mathbf{L}^{(m)} = \mathbf{L}_n^{(m)} - \mathbf{L}_d^{(m)} + D_m\mathbf{L}_D^{(m)} + \tau^I\tilde{\mathbf{L}}^{(m)} \tag{48}$$

3.3 Form of the prior parameters in I-smoothing distribution

One key issue in defining the I-smoothing distribution (38) is to choose an appropriate form of the prior parameters $\tilde{\boldsymbol{\mu}}^{(sm)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$. The most basic prior forms include *multiple-cluster* and *single-cluster*. When using a multiple-cluster prior, a global multiple-cluster matrix is used and the prior mean vector is speaker-specific. Alternatively, a *single-cluster* prior may also be selected, then the prior mean vector is speaker-independent. In both cases, prior covariance matrix is speaker-independent. From another point of view, the selected prior, either multiple-cluster or single-cluster, can be parameters of an existing model set (*static*) or model estimates in terms of sufficient statistics on-the-fly (*dynamic*). Also, the prior may be generated using different criteria, such as ML, MMI or MPE. Besides standard HMM, adaptively trained model parameters can also be used as the prior.

There are many feasible combinations of the above properties, which leads to quite a large number of possible priors to choose. Only general multiple-cluster and single-cluster prior are discussed below. A study of multiple-cluster MAP prior can be found in appendix 8.2. Note that, for the different forms of prior, only the corresponding sufficient statistics $\boldsymbol{\Theta}^{(p)} = \left\{\tau^I, \tau^I\tilde{\mathbf{G}}^{(m)}, \tau^I\tilde{\mathbf{K}}^{(m)}, \tau^I\tilde{\mathbf{L}}^{(m)}\right\}$ needs to be modified, the other sets of sufficient statistics remain unchanged in the re-estimation formulae.

3.3.1 Multiple-cluster prior

Here the prior mean vector is considered as speaker-specific, which is an interpolation among all clusters. Thus the following substitution is applied to equation (38),

$$\tilde{\boldsymbol{\mu}}^{(sm)} = \tilde{\mathbf{M}}^{(m)}\boldsymbol{\lambda}^{(s)} \tag{49}$$

where $\tilde{\mathbf{M}}^{(m)}$ is the prior cluster mean matrix, $\boldsymbol{\lambda}^{(s)}$ is the interpolation weights for speaker s. Consequently, the sufficient statistics (40) to (42) become

$$\tilde{\mathbf{G}}^{(m)} = \sum_s \tilde{\nu}_m^{(s)}\boldsymbol{\lambda}^{(s)}\boldsymbol{\lambda}^{(s)T} \tag{50}$$

$$\tilde{\mathbf{K}}^{(m)} = \left(\sum_s \tilde{\nu}_m^{(s)}\boldsymbol{\lambda}^{(s)}\boldsymbol{\lambda}^{(s)T}\right)\tilde{\mathbf{M}}^{(m)T} = \tilde{\mathbf{G}}^{(m)}\tilde{\mathbf{M}}^{(m)T} \tag{51}$$

$$\tilde{\mathbf{L}}^{(m)} = \tilde{\boldsymbol{\Sigma}}^{(m)} + \tilde{\mathbf{M}}^{(m)}\tilde{\mathbf{G}}^{(m)}\tilde{\mathbf{M}}^{(m)} \tag{52}$$

An existing CAT model may be used as the prior parameters $\tilde{\mathbf{M}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$. Alternatively, ML estimates for $\tilde{\mathbf{M}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$ can be employed as dynamic priors. By substituting ML estimates for $\tilde{\mathbf{M}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$, equations (50) to (52) become

$$\tilde{\mathbf{G}}^{(m)} = \frac{1}{\gamma^{ml}}\mathbf{G}_{ML}^{(m)} = \frac{1}{\gamma^{ml}}\sum_{s,t}\gamma_m^{ml}(t)\boldsymbol{\lambda}^{(s)}\boldsymbol{\lambda}^{(s)T} \tag{53}$$

$$\tilde{\mathbf{K}}^{(m)} = \frac{1}{\gamma^{ml}}\mathbf{K}_{ML}^{(m)} = \frac{1}{\gamma^{ml}}\sum_{s,t}\gamma_m^{ml}(t)\boldsymbol{\lambda}^{(s)}\mathbf{o}(t)^T \tag{54}$$

$$\tilde{\mathbf{L}}^{(m)} = \frac{1}{\gamma^{ml}}\mathbf{L}_{ML}^{(m)} = \frac{1}{\gamma^{ml}}\sum_{s,t}\gamma_m^{ml}(t)\mathbf{o}(t)\mathbf{o}(t)^T \tag{55}$$

Comparing the above statistics with ML statistics (7) to (9), they are all normalised by γ_m^{ml} to yield "unit" counts, as the "occupancy" of I-smoothing part is represented by τ^I. This is a natural extension from standard MPE training, in which ML estimates are also used as priors of I-smoothing distribution [22]. Besides the ML estimates, multi-cluster MAP estimates may also be used for dynamic multiple-cluster priors (appendix 8.2).

Observing the whole MPE sufficient statistics (45) to (48), when $\tau^I \to \infty$, sufficient statistics of I-smoothing distribution $\boldsymbol{\Theta}^{(p)}$ will dominate $\boldsymbol{\Theta}$. In this case, if multiple-cluster prior is used, it can be shown that the MPE estimates of model parameters will degrade to the multiple-cluster prior. Particularly, if multiple-cluster ML dynamic prior is used, the MPE estimates will degrade to the ML CAT estimate (4) and (5).

3.3.2 Single-cluster prior

When using a single-cluster prior, it is possible to use a standard single-cluster HMM as the prior. The following substitution is applied to equation (38)

$$\tilde{\boldsymbol{\mu}}^{(sm)} = \tilde{\boldsymbol{\mu}}^{(m)} \tag{56}$$

The sufficient statistics for I-smoothing (40) to (42) become

$$\begin{aligned} \tilde{\mathbf{G}}^{(m)} &= \sum_s \tilde{\nu}_m^{(s)} \boldsymbol{\lambda}^{(s)} \boldsymbol{\lambda}^{(s)T} &(57)\\ \tilde{\mathbf{K}}^{(m)} &= \Big(\sum_s \tilde{\nu}_m^{(s)} \boldsymbol{\lambda}^{(s)}\Big) \tilde{\boldsymbol{\mu}}^{(m)T} &(58)\\ \tilde{\mathbf{L}}^{(m)} &= \tilde{\boldsymbol{\mu}}^{(m)} \tilde{\boldsymbol{\mu}}^{(m)T} + \tilde{\boldsymbol{\Sigma}}^{(m)} &(59) \end{aligned}$$

Similarly to the multiple-cluster priors, either *static* or *dynamic* single-cluster priors can be employed. When using static priors, parameters from some existing standard HMM are selected. The topology of the prior HMM should be the same as the model to update, otherwise, there is no unique component mapping and it is hard to choose prior parameters for a particular component. Even with the same topology, risk still exists. In theory, it is not guaranteed that the order of components for a particular state in the prior HMM is the same as in the current model to update. However, if the prior HMM has the same seed model with the current model to update, e.g., the prior MPE-SI model and the current ML-CAT model are both generated from the same ML-SI model, it is assumed that the component order issue would not affect the training much.

When using dynamic priors, as the statistics are accumulated at component level on-the-fly, the component order is not an issue. Available options can be ML or MPE statistics. ML statistics can be accumulated as in standard HMM update. However, to accumulate single-cluster MPE statistics, "current single-cluster mean vector" is needed, which does not actually exist because the current model is a multiple-cluster model. One solution for this is to first read in a MPE-SI model as the "current standard model" for the first iteration and the single-cluster MPE statistics can be obtained. Then a new standard HMM estimated from the MPE statistics can be output as a "current single-cluster model" for the second iteration and so on. Therefore, for each iteration, two model sets are output, one is the multiple-cluster model to update, the other is a "current standard model" for accumulating standard MPE statistics in the next iteration.

Note that, different from the multiple-cluster prior, when $\tau^M \to \infty$, the MPE estimates can not degrade to the exact single-cluster prior due to the interpolation weights $\boldsymbol{\lambda}^{(s)}$.

3.4 Selection of the smoothing constant

The constant D_m is a critical value in discriminative training to make the weak-sense auxiliary function convex and give rapid convergence. As suggested in [22], D_m is determined by

$$D_m = \max\left(2\tilde{D}_m, E\gamma_m^d\right) \tag{60}$$

where $\tilde{D}_m$ is the smallest value required to ensure the updated covariance matrix (44) is positive-definite, E is a user-specified constant. γ_m^d is the total denominator posterior occupancy for component m. Thus,to find an appropriate $\tilde{D}_m$, the equation $\boldsymbol{\Sigma}^{(m)} = \mathbf{0}$ must be solved, For a single-cluster system, the equation for each dimension is a quadratic equation and can be easily solved [22]. However, for a multiple-cluster model, it can be shown that $\tilde{D}_m$ must satisfy the following constraint

$$(\mathbf{k}_1 + \tilde{D}_m \mathbf{k}_2)^T (\mathbf{G}_1 + \tilde{D}_m \mathbf{G}_2)^{-1} (\mathbf{k}_1 + \tilde{D}_m \mathbf{k}_2) = \mathbf{l}_1 + \tilde{D}_m \mathbf{l}_2 \tag{61}$$

Where $\mathbf{k}_1$ and $\mathbf{k}_2$ are $P \times 1$ vectors, $\mathbf{G}_1$ and $\mathbf{G}_2$ are $P \times P$ matrices, $\mathbf{l}_1$ and $\mathbf{l}_2$ are scalars, P is the number of clusters, they are all constants given appropriate accumulated statistics.

The inversion of a matrix can be represented as its adjoint matrix divided by the determinant. It can be shown that the order of the above polynomial equation is $P+1$. Therefore, unfortunately, this equation has no close-form solutions for $P > 3$.

For some special cases, such as $P = 2$ or $P = 3$, i.e. cubic or quartic polynomial equation, close form solutions of the above function can be derived and then the largest real root can be found directly. For larger number of clusters, numerical approaches may be used to find the largest real root. Or as a rough approximation, $E\gamma_m^d$ may be used directly as D_m together with appropriate variance floors to ensure positive definite of covariance matrices [8].

4 MPE Training of Interpolation Weights

In adaptive training framework, there are two distinct stages: model parameters estimation and transformation estimation. Linear transform based adaptive training has been investigated under some discriminative criteria [11] [13] [25]. Here cluster-based discriminative adaptive training is investigated. Discriminative update for the multiple-cluster model parameters has been discussed in section 3. In this section, MPE criterion is used to estimate the interpolation weights in CAT.

4.1 Re-estimation formulae of interpolation weights

In section 3.2, a weak-sense auxiliary function has been defined for a multiple-cluster model. It can again be used for interpolation weights update. Though the equation form is the same, the parameters to update here is $\boldsymbol{\lambda}^{(s)}$ rather than model parameters. Since the estimation of interpolation weights involves all Gaussian component, the smoothing function $\mathcal{Q}^{sm}(\boldsymbol{\lambda}^{(s)})$ may be defined at component level rather than speaker level (only relevant parts listed). Hence

$$\mathcal{Q}^{sm}(\boldsymbol{\lambda}^{(s)}) = -\sum_m \frac{D_m \nu_m}{2} \left(-2\boldsymbol{\lambda}^{(s)T}\mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\hat{\boldsymbol{\mu}}^{(sm)} + \boldsymbol{\lambda}^{(s)T}\mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)} \right) \quad (62)$$

where $\hat{\boldsymbol{\mu}}^{(sm)} = \hat{\mathbf{M}}^{(m)}\hat{\boldsymbol{\lambda}}^{(s)}$ are current parameters [9]. The constant D_m is the positive smoothing constant for component m. ν_m here is defined to reflect the proportion of each component when estimating interpolation weights:

$$\nu_m = \frac{\sum_t \gamma_m^n(t)}{\sum_{m,t} \gamma_m^n(t)} = \frac{\gamma_m^n}{\sum_m \gamma_m^n} \quad (63)$$

Similarly, the I-smoothing distribution become (only relevant parts listed)

$$\log p(\boldsymbol{\lambda}^{(s)}) = -\frac{\tau^I}{2} \sum_m \tilde{\nu}_m \left(\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)} - \tilde{\boldsymbol{\mu}}^{(sm)} \right)^T \boldsymbol{\Sigma}^{(m)-1} \left(\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)} - \tilde{\boldsymbol{\mu}}^{(sm)} \right) \quad (64)$$

where $\tilde{\boldsymbol{\mu}}^{(sm)}$ are prior parameters and can have various forms as discussed in section 3.3, $\tilde{\nu}_m$ has the same form as the above ν_m except for using ML posterior occupancy γ_m^{ml} instead of γ_m^n.

Re-estimation formulae for interpolation weights can be derived by differentiating the whole auxiliary function. The interpolation weight vector can then be re-estimated as:

$$\boldsymbol{\lambda}^{(s)} = \mathbf{G}^{(s)-1}\mathbf{k}^{(s)} \quad (65)$$

[8]The approximate selection has been shown in experiments to give only marginal difference from the strict selection. In this work, strict selection of D_m is used for 2-cluster systems. For more clusters, approximate selection is used.

[9]Note in weights update, $\hat{\mathbf{M}}^{(m)} = \mathbf{M}^{(m)}$

where the sufficient statistics are

$$\gamma_m^n = \sum_t \gamma_m^n(t), \qquad \gamma_m^d = \sum_t \gamma_m^d(t) \tag{66}$$

$$\mathbf{G}^{(s)} = \sum_m \left(\gamma_m^n - \gamma_m^d + D_m\nu_m + \tau^I\tilde{\nu}_m\right)\mathbf{M}^{(m)T}\mathbf{\Sigma}^{(m)-1}\mathbf{M}^{(m)} \tag{67}$$

$$\mathbf{k}^{(s)} = \sum_m \mathbf{M}^{(m)T}\mathbf{\Sigma}^{(m)-1}\left(\sum_t \gamma_m^n(t)\mathbf{o}(t) - \sum_t \gamma_m^d(t)\mathbf{o}(t) + D_m\mathbf{k}_D^{(m)} + \tau^I\tilde{\mathbf{k}}^{(m)}\right) \tag{68}$$

where $D_m\mathbf{k}_D^{(m)}$ are statistics from the smoothing function

$$\mathbf{k}_D^{(m)} = \nu_m\left(\mathbf{M}^{(m)}\hat{\boldsymbol{\lambda}}^{(s)}\right) \tag{69}$$

$\tau^I\tilde{\mathbf{k}}^{(m)}$ are statistics from the I-smoothing distribution $\log p(\boldsymbol{\lambda}^{(s)})$

$$\tilde{\mathbf{k}}^{(m)} = \tilde{\nu}_m\tilde{\boldsymbol{\mu}}^{(sm)} \tag{70}$$

Similar to multiple-cluster model update, there are several options for the prior parameters in I-smoothing distribution. If a multiple-cluster prior is used, $\tilde{\boldsymbol{\mu}}^{(sm)}$ in (70) is replaced by $\mathbf{M}^{(m)}\tilde{\boldsymbol{\lambda}}^{(s)}$. Particularly, when using ML estimates as the prior parameters, equation (70) become

$$\tilde{\mathbf{k}}^{(m)} = \frac{1}{\sum_{m,t}\gamma_m^{ml}(t)}\sum_t \gamma_m^{ml}(t)\mathbf{o}(t) \tag{71}$$

In this case, if $\tau^I \to \infty$, the MPE estimate will degrade to ML estimate (10).

If standard single-cluster prior is selected as in section 3.3.2, $\tilde{\boldsymbol{\mu}}^{(sm)}$ is replaced as $\tilde{\boldsymbol{\mu}}^{(m)}$ in equation (70). $\tilde{\boldsymbol{\mu}}^{(m)}$ also has various forms as in section 3.3.2.

For the smoothing constant D_m, to guarantee validity of the weak-sense auxiliary function and optimisation convergence, it can be simply set to $E\gamma_m^d$, where E is commonly selected as 1 or 2 as described in [22].

4.2 Simplified discriminative adaptive training

So far, the complete update formulae for discriminative cluster adaptive training based on MPE criterion have been obtained. Recently, various techniques of discriminative adaptive training are of research interest. However, complete discriminative adaptive training has not achieved satisfactory results in unsupervised adaptation. In practice, many implementations of discriminative adaptive training schemes employ a simplified training scheme, which yields about the same performance as discriminatively training all the parameters, but is simpler and more consistent when dealing with unsupervised adaptation task [25]. The basic training procedures are as follows.

1. ML-based adaptive training is performed on both model parameters and transformation parameters with normal iterative training fashion.
2. ML trained transformation parameters are fixed and only model parameters are discriminatively trained in later iterations.

In adaptation, given the discriminatively trained model, ML criterion is again used to estimate transformation parameters for test domains. In common with the simplified implementation, this work will only consider discriminatively training a multiple-cluster model given ML estimates of all interpolation weights in the experiments.

5 Structured Transforms

Standard adaptive training often uses a single transform to represent non-speech variabilities. However, for found data, there are commonly multiple acoustic factors affecting the speech signal. More recently schemes use multiple forms of transformations, denoted here as *structured transforms* (ST), to represent complex non-speech variabilities in an adaptive training framework [27] [9]. In this paper, one kind of ST is investigated, which combines CAT interpolation weights [8] and CMLLR transforms [7].

In the ST of CAT+CMLLR, the canonical model is a *multiple-cluster model*. Since CMLLR can be viewed as a feature transform, the ML auxiliary function for this ST-based adaptive training is:

$$\mathcal{Q}(\mathcal{M}) = -\frac{1}{2}\sum_{m,t,s}\gamma_m(t)\left(\log|\mathbf{\Sigma}^{(m)}| - \log(|\mathbf{A}^{(s)}|^2) + (\mathbf{o}^{(s)}(t) - \boldsymbol{\mu}^{(sm)})^T\mathbf{\Sigma}^{(m)-1}(\mathbf{o}^{(s)}(t) - \boldsymbol{\mu}^{(sm)})\right) \quad (72)$$

CMLLR feature transform gives

$$\mathbf{o}^{(s)}(t) = \mathbf{A}^{(s)}\mathbf{o}(t) + \mathbf{b}^{(\mathbf{s})} \quad (73)$$

CAT interpolation of the means is $\boldsymbol{\mu}^{(sm)} = \mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)}$ as equation (1).

As the auxiliary function suggests, estimation of the multiple-cluster canonical model is a simple extension to the model-based CAT estimation approach [8]. In addition to considering the speaker-specific interpolation weights, all features are transformed using the associated CMLLR transform. Re-estimation formulae are the same as (4) and (5) except for using $\mathbf{o}^{(s)}(t)$ instead of $\mathbf{o}(t)$ in sufficient statistics. Transformations estimation is also a simple iterative process, where given the interpolation weights, the adapted mean, $\boldsymbol{\mu}^{(sm)}$ is used instead of $\boldsymbol{\mu}^{(m)}$ to estimate the CMLLR transform as described in [7]. Then the interpolation weights, $\boldsymbol{\lambda}^{(s)}$ are estimated using the transformed features $\mathbf{o}^{(s)}(t)$ in equation (10).

Based on the ST of CAT+CMLLR, discriminatively training multiple-cluster model parameters is also a straight-forward extension of model update in discriminative CAT. By using the transformed feature vectors $\mathbf{o}^{(s)}(t)$ instead of standard feature vectors $\mathbf{o}(t)$, formulae in section 3.2.4 can be modified for the adaptive training with ST. Since a simplified MPE training scheme is employed here, discriminative re-estimation of transformation parameters are not considered. A detailed discussion can be found in [27].

6 Experiments

6.1 Systems description

The performance of discriminative cluster adaptive training was evaluated on a state-of-the-art large vocabulary speech recognition system, conversational telephone speech (Switchboard). The training dataset consists of 3 corpora (`che`, `Swbd`, `sw-cell`), 5446 speakers (2747 female, 2699 male), giving a total of about 295 hours of data. This is referred to as the `h5train03` training dataset. The performance was evaluated on two held out test sets. One test dataset is a subset, half of the `dev01` test data, consisting of 59 speakers (30 female, 29 male), `Swbd` corpora, about 3 hours, called `dev01sub` test data. Another test dataset is the `eval03` dataset consisting of 144 speakers (77 female, 67 male), two different corpora (`Swbd` and `Fisher`)about 6 hours.

All systems used a 12-dimensional PLP front-end with log energy and its first, second and third derivatives. Then a HLDA transform was applied to reduce the feature dimension to 39. VTLN was also used. The use of VTLN decreased the possible gains that could be obtained using adaptive training, but gave a more realistic baseline. A tri-gram language model was used in full decoding for all experiments.

Two baseline systems were built. A 16 components-per-state system was used for fast run in development. A 28 component-per-state system was built for generating results of state-of-the-art systems.

Besides speaker-independent (SI) and gender-dependent (GD) systems, several adaptive MPE systems were constructed using the simplified MPE adaptive training, in which only the canonical model parameters were updated while the ML-trained transforms (or interpolation weights) were fixed [25]. The MPE adaptive systems include MPE-SAT, MPE-CAT and MPE-ST which employed CAT weights and CMLLR transforms as the structured transforms (ST). The MPE-SAT used CMLLR [7]. As CMLLR can be implemented as feature transforms, training of MPE-SAT is the same as MPE-SI training except for applying CMLLR to feature vectors. During training and test adaptation, a global interpolation weight was estimated and separate speech and silence transforms were used for CMLLR. To fully compare the adaptation performance, CMLLR adaptation was also applied on top of MPE baseline systems and MPE-CAT systems. In testing adaptation, for all systems, supervisions were generated using the MPE-SI model.

All ML training employed 4 iterations. MPE training for 16 component systems employed 4 MPE iterations for fast development. For 28 component state-of-the-art systems, 8 MPE iterations were used. An exception is GD-MAP, which has further smoothing in MPE training for GD model [23]. The GD-MPE-MAP system originated from 8-iteration trained MPE-SI model and employed 2 further MPE training iterations with only mean and component weights updated. Therefore, it was actually about 10 MPE training iterations, which should be a little more powerful than the comparable 8-iteration systems.

6.2 Results and analysis

6.2.1 Baseline performance

Single-cluster baseline SI(GI) systems' performance is shown in table 1.

SI/GI System	`dev01sub`		`eval03`	
	ML	MPE	ML	MPE
16 Comp.	33.4	30.4	32.6	29.2
28 Comp.	32.2	28.9	31.5	28.3

Table 1: Baseline SI/GI systems performance

From the results, the 28 component system works significantly better than the 16 component system in both ML and MPE training. Performance of the 28 component system is suitable for evaluating new techniques in state-of-the-art system. The SI baseline results also show that MPE training significantly outperforms ML training by 3 to 3.4 percent absolute on both systems. This is why investigation on MPE is of great interest.

Comp.	ML System	`dev01sub`	`eval03`
16 Comp.	GD	32.7	32.2
	CAT(GDInit)	32.6	31.9
28 Comp.	GD	32.2	31.2
	CAT(GDInit)	31.9	31.2

Table 2: ML baseline performance: GD and CAT

Results of ML-CAT systems initialised with gender information are given in table 2. This is for comparison with MPE-CAT systems in later experiments. The results also show that ML-CAT systems significantly outperform ML-SI baselines and are as good as or better than the ML-GD systems. Then it is also interesting to compare MPE-CAT systems to MPE-SI and MPE-GD systems in later experiments.

6.2.2 Investigation on different forms of I-smoothing prior in MPE-CAT

I-smoothing has been shown to be essential for MPE training [22]. As previously indicated, the selection of the prior parameters is of great interest for MPE-CAT. Here several multiple-cluster and single-cluster priors were explored. Multiple-cluster MAP prior (appendix 8.2 [10]) was also investigated. In this investigation, the development system setup was used, i.e., 16 components per state with 4 MPE training iterations.

Table 3 shows the performance of MPE-CAT using static and dynamic single-cluster priors. Table 4 shows the results of MPE-CAT using dynamic multiple-cluster priors.

Single-cluster Prior	Criterion	`dev01sub`	`eval03`
dynamic	ML	29.7	29.1
	MPE	29.3	28.4
static	MPE	29.3	28.5

Table 3: WER of MPE-CAT systems with different single-cluster priors

From table 3, the form of the I-smoothing priors have significant effect on the generalisation ability, the difference is up to 0.7 percent absolute on `eval03`. The experiment shows that the MPE based priors can significantly outperform ML based prior. Dynamic and static MPE priors have slight difference. It is also worth noting that by using MPE prior, the performance of MPE-CAT systems are much better than the MPE-SI baselines. For `eval03`, they are even approaching the performance of 28 component MPE-SI system.

Criterion	τ^M	`dev01sub`	`eval03`
ML	—	29.7	28.9
MAP	0	29.7	28.9
	25	29.7	28.8
	50	29.6	28.8
	100000	29.4	28.6
	∞	29.3	28.5

Table 4: WER of MPE-CAT systems with different dynamic multiple-cluster priors

Table 4 shows the performances of dynamic multiple-cluster priors. Compared to table 3, dynamic multiple-cluster ML prior is slightly better than the dynamic single-cluster ML prior on `eval03`. As indicated in appendix 8.2, the MAP prior is a trade-off between the multiple-cluster dynamic ML prior and the standard static MPE-SI prior. τ^M is the coefficient to balance the two priors. From the results, there is also a trend of improving performance from ML-based prior to MPE-based prior.

The priors can be viewed as the *center* values of an I-smoothing distribution. They act as the back-off values in model parameters estimation. A "better" prior will result in a "better" performance [11]. Also, dynamic priors may be expected to be slightly better than static priors as they are calculated on-the-fly with new component alignment using current static model set. Multiple-cluster priors should be slightly better than the equivalent standard priors as they incorporate some adaptation effect. Since there is only marginal difference between the dynamic and static priors, the standard single-cluster MPE-SI model was used as the I-smoothing prior for all the following MPE-CAT and MPE-ST systems.

[10] For priors with MAP adaptation type, standard MPE-SI model was used as the additional prior of Normal-Wishart distribution for MAP estimates.

[11] "better" is defined in terms of word error rate.

6.2.3 Investigation on initialisation approaches of CAT

This is a fast development study on initialisation approaches, 16 component-per-state systems with 4 MPE training iterations were employed. Two kinds of prior knowledge based initialisation approaches and several eigenvoices initialised systems were investigated. One kind of prior knowledge is gender information for each speaker as discussed in 2.2. The corresponding system is a *2-cluster* CAT system. Besides gender information, corpus information is also available for `h5etrain` as indicated in system description. Thus it is possible to construct a *3-cluster* CAT system initialised by assigning the initial weights: $[1, 0, 0]$ for speakers in `che` corpus, $[0, 1, 0]$ for `Swbd` and $[0, 0, 1]$ for `sw-cell`. For eigen-decomposition initialisation approach, both non-bias and bias systems were built. As discussed in [8], the eigen-decomposition approach naturally generates a bias cluster. The bias cluster may be kept in later ML-CAT training, i.e. weight value for the bias cluster is always set to be 1 and only values associated with the other clusters are updated. During model update, parameters are re-estimated as normal. Alternatively, after eigen-decomposition initialisation, model parameters and weight vectors are both re-estimated as normal in later ML-CAT training without keeping a weight 1 for the bias cluster. This is referred to as a non-bias system. *2-cluster*, *3-cluster* and *4-cluster* eigenvoices initialised systems with bias cluster were built. A *3-cluster* non-bias system was also built for comparison. Performance is shown in table 5.

Initialization	Bias	#Cluster	`dev01sub`		`eval03`	
			ML	MPE	ML	MPE
gender info.	no	2	32.6	29.3	31.9	28.5
corpus info.		3	32.3	29.2	31.7	28.3
eigenvoices		3	32.3	29.0	31.5	28.2
eigenvoices	yes	2	32.8	29.3	32.0	28.5
eigenvoices		3	32.3	29.0	31.6	28.3
eigenvoices		4	32.3	29.0	31.5	28.3

Table 5: Comparison between different CAT initialisation approaches

From the results, *3-cluster* CAT systems perform better than *2-cluster* CAT systems, which is reasonable since more clusters should basically be more descriptive for the non-homogeneity of the speech data. There is no distinct difference between the bias and non-bias eigenvoices initialised *3-cluster* systems. The table also shows that *4-cluster* system did not perform better than *3-cluster* systems. This may imply that, having captured enough non-homogeneity, more clusters may not help. Thus for a certain set of training data, there exists an optimal number of clusters which can efficiently describe the homogeneity. Comparing the two *3-cluster* non-bias systems, eigen-decomposition approach slightly outperforms prior knowledge based approach with the same number of cluster. To investigate this in detail, the break-down MPE wer number on `eval03` of the two systems is given in table 6 [12].

Initialization	Swbd	Fisher	Overall
corpus info.	32.6	23.6	28.3
eigenvoices	32.4	23.7	28.2

Table 6: Breakdown MPE WER on `eval03` of *3-cluster* non-bias systems

The superiority of the eigen-decomposition approach may contribute to the "orthogonal" property of eigenvoices which makes a better description of training data. Table 6 shows that for test data with the same corpus as training (`Swbd`), the eigen-decomposition approach yields better

[12] `dev01sub` has the same corpus with training data, thus no need to show breakdown numbers.

result. However, for `Fisher` data which is not included in training data, eigen-decomposition approach does not perform as well as corpus initialisation approach. This may imply that for test data different from training data, prior knowledge based approach may have better generalisation ability.

In the following experiments, gender-information is used to initialise CAT and ST systems to obtain comparable results to gender-dependent (GD) systems.

6.2.4 State-of-the-art MPE-CAT system performance

In this experiment, a state-of-the-art gender-initialised 2-cluster MPE-CAT system was built for comparison with the gender-dependent (GD) and GD-MAP MPE systems [23]. All systems were 28 component systems with 8 MPE training iterations except for GD-MPE-MAP. Note that for GD and GD-MPE-MAP systems, the test set gender information was assumed to be known, while for CAT systems, the information was not required as discussed in section 2.1. Table 7 shows both ML and MPE results.

System	`dev01sub`		`eval03`	
	ML	MPE	ML	MPE
GI	32.2	28.9	31.5	28.3
GD	32.2	29.2	31.2	28.5
GD(MPE-MAP)	—	28.5	—	28.0
CAT(GD-Init)	31.9	28.8	31.2	27.8

Table 7: Comparison between CAT system and baselines

From the results, MPE-GD was overtrained and got even worse than the MPE-GI system. With the MAP prior, GD-MPE-MAP gains over both MPE-GD and MPE-SI. This is consistent with the gains that were obtained on the Broadcast News task [23]. The gender-initialised CAT system performs well even without knowing any gender information on test data. Though it does not get as much gain as GD-MPE-MAP in `dev01sub`, it consistently obtains gain over GI systems both in ML and MPE cases, and gets more gains in the larger test dataset `eval03`, which is more representative. The good performance may contribute to the soft interpolation weights, which improved the generalisation ability during adaptation. In table 7, MPE-CAT systems significantly outperform ML-CAT systems by 3 to 3.4 percent absolute, which is consistent with the gain of MPE baselines. After MPE-training, CAT system keep outperforming GI system. Especially for `eval03` dataset, it is much better than the baselines. This shows that MPE-training can be effectively used for multiple-cluster systems.

6.2.5 State-of-the-art MPE-ST systems performance

Finally, several state-of-the-art adaptive MPE systems were built. All systems are 28 component systems with 8 MPE training iterations except for GD-MPE-MAP. Table 8 shows the performance.

System	Training Adaptation	Test Adaptation	`dev01sub`	`eval03`
GI/SI	—	CMLLR	27.1	26.1
GD(MPE-MAP)	gender info	CMLLR	27.1	25.8
SAT	CMLLR	CMLLR	26.8	25.9
CAT(GDInit)	CAT	ST	27.1	25.7
ST(GDInit)	ST	ST	26.6	25.5

Table 8: Comparison between MPE-trained systems with adaptation

Here ST represents CAT+CMLLR. In CAT system, test adaptation of ST means applying CMLLR adaptation on top of CAT adapted model. From table 8, adaptively trained systems always outperform standard systems. GD-MPE-MAP still outperforms MPE-GI baseline after adaptation though the large gain on `dev01sub` disappears after adaptation. As the test gender information was assumed to be known, there is no gender classification error propagated to the adaptation stage for GD-MPE-MAP. MPE-CAT system slightly outperforms GD-MPE-MAP on both test datasets after ST adaptation. One interesting comparison is among SAT, CAT with ST adaptation and ST system. SAT and CAT are transform-based and cluster-based adaptive systems respectively. After adaptation, the two systems show similar performance. There is no consistent trend that one is better than the other. Both systems outperform baseline MPE-SI system and are similar to GD-MPE-MAP system after adaptation. MPE-ST system always gives the lowest error rates on both `dev01sub` and `eval03` and significantly outperforms baseline MPE-SI system. Compared to MPE-SAT, it is also shown to be more powerful, especially on `eval03`. This shows that MPE adaptive training with ST can effectively outperform standard MPE adaptive training systems.

7 Conclusion

This paper investigates various aspects of applying discriminative training criteria to cluster adaptive training (CAT), in which a multi-cluster hmm model is used as canonical model and a set of interpolation weights are used to represent non-speech variabilities. Theory based on Minimum Phone Error (MPE) criterion is developed. By redefining appropriate weak-sense auxiliary function for multiple-cluster model, formulae for discriminatively updating multi-cluster model parameters and interpolation weights are all obtained. Among the redefinitions, selection of I-smoothing distribution prior form is an important issues and discussed in detail. The MPE training for CAT can also be viewed as a discriminative training method for eigenvoices systems. Provided a weak-sense auxiliary function is available, other discriminative training criteria such as MMI, can also be applied to cluster adaptive training using similar approach. Furthermore, CAT interpolation weights and CMLLR transforms are combined in an adaptive training framework to model complex non-speech variabilities, which is called structured transforms (ST). The discriminative CAT can then be extended to discriminative adaptive training with ST. A simplified discriminative training scheme is used in this work, in which only model parameters are discriminatively trained given ML estimates of all transform parameters. Experiments show that MPE criterion can be successfully used for CAT. The gain obtained from MPE-CAT compared to ML-CAT is consistent with the gain of traditional MPE systems compared to ML systems. For a 2-cluster gender information initialised MPE-CAT system with global weights, it significantly outperform baseline MPE-GI and MPE-GD systems and yields similar performance as GD-MPE-MAP system. The MPE-CAT systems also significantly outperform ML-CAT systems and get consistent gain as standard MPE training. In more complex adaptation experiments, MPE-ST system also significantly outperforms all traditional adaptive training techniques.

References

[1] T. Anastasakos, J. Mcdonough, R. Schwartz, and J. Makhoul. A compact model for speaker adaptive training. In *Proc. ICSLP*, pages 1137–1140, 1996.

[2] B.H.Juang, W.Chou, and C.H.Lee. Minimum classification error rate methods for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 5:266–277, 1997.

[3] C.J.Leggetter and P.C.Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. *Computer Speech and Language*, 9:171–186, 1995.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[5] D.Povey and P.C.Woodland. Minimum phone error and I-smoothing for improved discriminative training. In *Proc. ICASSP*, 2002. Orlando.

[6] M. J. F. Gales. Cluster adaptive training for speech recognition. In *Proc. ICSLP*, pages 1783–1786, 1998.

[7] M. J. F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12:75–98, 1998.

[8] M. J. F. Gales. Cluster adaptive training of hidden markov models. *IEEE Transactions on Speech and Audio Processing*, 8:417–428, 2000.

[9] M. J. F. Gales. Acoustic factorization. In *Proc. ASRU*, 2001.

[10] J. L. Gauvain and C. H. Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Trans. on SAP*, 2:291–298, 1994.

[11] A. Gunawardana and W. Byrne. Discriminative Speaker Adaptation with Conditional Maximum Likelihood Linear Regression. In *Proc. EuroSpeech*, 2001. Scandinavia.

[12] X. Huang, A. Acero, and H. Hon. *Spoken Language Processing*. Prentice Hall PTR, 2001.

[13] T. Schaaf J. McDonough and A. Waibel. On Maximum Mutual Information Speaker- Adapted Training . In *Proc. ICASSP*, Florida, USA, May, 2002.

[14] T. Kosaka and S. Sagayama. Tree structured speaker clustering for fast speaker adaptation. In *Proc. ICASSP*, volume 1, pages 245–248, 1994.

[15] R. Kuhn, J. C. Junqua, P. Nguyen, and N. Niedzielski. Rapid speaker adaptation in eigenvoice space. *IEEE Trans. on SAP*, 8(6):695–707, 2000.

[16] R. Kuhn, P. Nguyen, J. C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, and M. Contolini. Eigenvoices for speaker adaptation. In *Proc. ICSLP*, pages 1771–1774, 1998.

[17] L.R.Bahl, P.F.Brown, P.V.De Souza, and R.L. Mercer. Maximum mutual information estimation of hidden markov model parameters for speech recognition. In *Proc. ICASSP*, volume 1, pages 49–52, 1986.

[18] M.J.F.Gales. Transformation smoothing for speaker and environmental adaptation. In *Proc. EuroSpeech*, 1997.

[19] M.J.F.Gales. Multiple-cluster adaptive training schemes. In *Proc. ICASSP*, 2001.

[20] Y. Normandin. *Hidden markov models, maximum mutual information estimation, and the speech recognition problem*. PhD thesis, McGill University, 1991.

[21] P. Nuguyen, C. Wellekens, and J. C. Junqua. Maximum likelihood eigenspace and MLLR for speech recognition in noisy environments. In *Proc. Eurospeech*, pages 2519–2522, 1999.

[22] D. Povey. *Discriminative Training for Large Vocabulary Speech Recognition*. PhD thesis, Cambridge University, 2003.

[23] D. Povey, M. J. F. Gales, D. Y. Kim, and P. C. Woodland. MMI-MAP and MPE-MAP for acoustic model adaptation. In *Proc. EuroSpeech*, 2003.

[24] R. Schlter and W. Macherey. Comparison of discriminative training criteria. In *Proc. ICASSP*, 1998.

[25] L. Wang and P. C. Woodland. Discriminative Adaptive Training Using The MPE Criterion. In *Proc. ASRU*, 2003.

[26] P. C. Woodland and D. Povey. Large scale discriminative training of hidden markov models for speech recognition. *Computer Speech and Language*, 16:25–48, 2002.

[27] K. Yu and M. J. F. Gales. Adaptive Training Using Structured Transforms. In *Proc. ICASSP*, 2004.

8 Appendix

8.1 Validity of multiple-cluster smoothing function

The smoothing auxiliary function in discriminative CAT is defined as a sum of speaker-level functions as described in section 3.2.2:

$$\begin{aligned}\mathcal{Q}^{sm}(\mathcal{M}) &= \sum_s \mathcal{Q}^{sm}{}_s(\mathcal{M}) = -\sum_{m,s} \frac{D_m \nu_m^{(s)}}{2}\Big(\log|\boldsymbol{\Sigma}^{(m)}| + \mathrm{tr}\Big((\hat{\boldsymbol{\Sigma}}^{(m)} + \hat{\boldsymbol{\mu}}^{(sm)}\hat{\boldsymbol{\mu}}^{(sm)T})\boldsymbol{\Sigma}^{(m)-1}\Big) \\ &\quad -2\boldsymbol{\lambda}^{(s)T}\mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\hat{\boldsymbol{\mu}}^{(sm)} + \boldsymbol{\lambda}^{(s)T}\mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)}\Big) \qquad (74)\end{aligned}$$

where $\hat{\boldsymbol{\mu}}^{(sm)} = \hat{\mathbf{M}}^{(m)}\hat{\boldsymbol{\lambda}}^{(s)}$, $\hat{\mathbf{M}}^{(m)}$, $\hat{\boldsymbol{\Sigma}}^{(m)}$ and $\hat{\boldsymbol{\lambda}}^{(m)}$ are the current parameters and "tr()" is the trace of a matrix. The constant D_m is the positive smoothing constant for component m to ensure convergence.

To simplify the derivation, we use the following notations:

$$\begin{aligned}A &= \sum_m -\frac{D_m \nu_m^{(s)}}{2}\log|\boldsymbol{\Sigma}^{(m)}| \\ B &= \sum_m -\frac{D_m \nu_m^{(s)}}{2}\mathrm{tr}\Big((\hat{\boldsymbol{\Sigma}}^{(m)} + \hat{\mathbf{M}}^{(m)}\hat{\boldsymbol{\lambda}}^{(s)}\hat{\boldsymbol{\lambda}}^{(s)T}\hat{\mathbf{M}}^{(m)T})\boldsymbol{\Sigma}^{(m)-1}\Big) = \mathrm{tr}(B_1\boldsymbol{\Sigma}^{(m)-1}) \\ C &= \sum_m -\frac{D_m \nu_m^{(s)}}{2}\boldsymbol{\lambda}^{(s)T}\mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\hat{\mathbf{M}}^{(m)}\hat{\boldsymbol{\lambda}}^{(s)} = \boldsymbol{\lambda}^{(s)T}\mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\mathbf{c} \\ D &= \sum_m -\frac{D_m \nu_m^{(s)}}{2}\boldsymbol{\lambda}^{(s)T}\mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)}\end{aligned}$$

Therefore, the speaker-specific smoothing auxiliary function can be represented as:

$$\mathcal{Q}^{sm}{}_s(\mathcal{M}) = A + B - 2C + D$$

Before derivation, we stated the following results of matrix derivatives:

$$\frac{\partial \ln|\mathbf{W}|}{\partial \mathbf{W}} = (\mathbf{W}^T)^{-1} \qquad (75)$$

$$\frac{\partial(\mathbf{a}^T\mathbf{W}\mathbf{b})}{\partial \mathbf{W}} = \mathbf{b}\mathbf{a}^T \qquad (76)$$

$$\frac{\partial(\mathbf{a}^T\mathbf{W}^T\mathbf{A}\mathbf{W}\mathbf{b})}{\partial \mathbf{W}} = \mathbf{A}^T\mathbf{W}\mathbf{a}\mathbf{b}^T + \mathbf{A}\mathbf{W}\mathbf{b}\mathbf{a}^T \qquad (77)$$

$$\frac{\partial(\mathbf{a}^T\mathbf{W}\mathbf{a})}{\partial \mathbf{a}} = (\mathbf{W} + \mathbf{W}^T)\mathbf{a} \qquad (78)$$

$$\frac{\partial(\mathbf{a}^T\mathbf{b})}{\partial \mathbf{a}} = \mathbf{b} \qquad (79)$$

$$\frac{\partial(\mathrm{tr}(\mathbf{A}\mathbf{W}))}{\partial \mathbf{W}} = \mathbf{A}^T \qquad (80)$$

where $\mathbf{W}$ and $\mathbf{A}$ are matrices [13] and $\mathbf{b}$ and $\mathbf{a}$ are vectors.

Therefore, we can easily get the following differentiation results of C and D with respect to $\mathbf{M}^{(m)}$ (note that differentiation results of A and B are 0, $\boldsymbol{\Sigma}^{(m)-1}$ is symmetric,in this case $\hat{\boldsymbol{\lambda}} = \boldsymbol{\lambda}$):

$$\begin{aligned}\frac{\partial C}{\partial \mathbf{M}^{(m)}} &= -\frac{D_m \nu_m^{(s)}}{2}\boldsymbol{\Sigma}^{(m)-1}\hat{\mathbf{M}}^{(m)}\boldsymbol{\lambda}^{(s)}\boldsymbol{\lambda}^{(s)T} \\ \frac{\partial D}{\partial \mathbf{M}^{(m)}} &= -D_m \nu_m^{(s)}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)}\boldsymbol{\lambda}^{(s)T}\end{aligned}$$

[13] square matrix is required only for the equation (75)

Therefore, the whole differentiation result with respect to $\mathbf{M}^{(m)}$ at the current model parameters is:

$$\frac{\partial \mathcal{Q}^{sm}{}_s(\mathcal{M})}{\partial \mathbf{M}^{(m)}}\Big|_{\mathcal{M}=\hat{\mathcal{M}}} = D_m\nu_m^{(s)}\mathbf{\Sigma}^{(m)-1}\big(\hat{\mathbf{M}}^{(m)} - \mathbf{M}^{(m)}\big)\boldsymbol{\lambda}^{(s)}\boldsymbol{\lambda}^{(s)T}\Big|_{\mathcal{M}=\hat{\mathcal{M}}} = 0$$

Similarly we can also get the differentiation results of A, C and D with respect to $\mathbf{\Sigma}^{(m)-1}$ according to the above formula:

$$\begin{aligned}
\frac{\partial A}{\partial \mathbf{\Sigma}^{(m)-1}} &= \frac{D_m\nu_m^{(s)}}{2}\mathbf{\Sigma}^{(m)-1} \\
\frac{\partial B}{\partial \mathbf{\Sigma}^{(m)-1}} &= -\frac{D_m\nu_m^{(s)}}{2}(\hat{\mathbf{\Sigma}}^{(m)} + \hat{\mathbf{M}}^{(m)}\boldsymbol{\lambda}^{(s)}\boldsymbol{\lambda}^{(s)T}\hat{\mathbf{M}}^{(m)T}) \\
\frac{\partial C}{\partial \mathbf{\Sigma}^{(m)-1}} &= -\frac{D_m\nu_m^{(s)}}{2}\hat{\mathbf{M}}^{(m)}\boldsymbol{\lambda}^{(s)}\boldsymbol{\lambda}^{(s)T}\mathbf{M}^{(m)T} \\
\frac{\partial D}{\partial \mathbf{\Sigma}^{(m)-1}} &= -\frac{D_m\nu_m^{(s)}}{2}\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)}\boldsymbol{\lambda}^{(s)T}\mathbf{M}^{(m)T}
\end{aligned}$$

Therefore, the whole differentiation result with respect to $\mathbf{\Sigma}^{(m)-1}$ at current parameters is:

$$\begin{aligned}
\frac{\partial \mathcal{Q}^{sm}{}_s(\mathcal{M})}{\partial \mathbf{\Sigma}^{(m)-1}}\Big|_{\mathcal{M}=\hat{\mathcal{M}}} &= \frac{D_m\nu_m^{(s)}}{2}\big(\mathbf{\Sigma}^{(m)-1} - \hat{\mathbf{\Sigma}}^{(m)} - \hat{\mathbf{M}}^{(m)}\boldsymbol{\lambda}^{(s)}\boldsymbol{\lambda}^{(s)T}\hat{\mathbf{M}}^{(m)T} \\
&\quad + 2\hat{\mathbf{M}}^{(m)}\boldsymbol{\lambda}^{(s)}\boldsymbol{\lambda}^{(s)T}\mathbf{M}^{(m)T} - \mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)}\boldsymbol{\lambda}^{(s)T}\mathbf{M}^{(m)T}\big)\Big|_{\mathcal{M}=\hat{\mathcal{M}}} \\
&= 0
\end{aligned}$$

Since $\mathbf{\Sigma}^{(m)-1}$ is a function of $\mathbf{\Sigma}^{(m)}$, from the above, we also have

$$\frac{\partial \mathcal{Q}^{sm}{}_s(\mathcal{M})}{\partial \mathbf{\Sigma}^{(m)}}\Big|_{\mathcal{M}=\hat{\mathcal{M}}} = 0$$

For cluster weights $\boldsymbol{\lambda}^{(s)}$, the same smoothing function is used. The $\boldsymbol{\lambda}^{(s)}$ is only relevant to C and D. Note that when update weights, $\hat{\mathbf{M}}^{(m)} = \mathbf{M}^{(m)}$, $\hat{Cov}^{(m)-1} = \mathbf{\Sigma}^{(m)-1}$.

$$\begin{aligned}
\frac{\partial C}{\partial \boldsymbol{\lambda}^{(s)}} &= \sum_m -\frac{D_m\nu_m^{(s)}}{2}\mathbf{M}^{(m)}\mathbf{\Sigma}^{(m)-1}\mathbf{M}^{(m)}\hat{\boldsymbol{\lambda}}^{(s)} \\
\frac{\partial D}{\partial \boldsymbol{\lambda}^{(s)}} &= 2\sum_m -\frac{D_m\nu_m^{(s)}}{2}\mathbf{M}^{(m)}\mathbf{\Sigma}^{(m)-1}\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)}
\end{aligned}$$

Therefore,

$$\frac{\partial \mathcal{Q}^{sm}{}_s(\mathcal{M})}{\partial \boldsymbol{\lambda}^{(s)}}\Big|_{\boldsymbol{\lambda}^{(s)}=\hat{\boldsymbol{\lambda}}^{(s)}} = \sum_m D_m\nu_m^{(s)}\big(\mathbf{M}^{(m)}\mathbf{\Sigma}^{(m)-1}\mathbf{M}^{(m)}\hat{\boldsymbol{\lambda}}^{(s)} - \mathbf{M}^{(m)}\mathbf{\Sigma}^{(m)-1}\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(s)}\big)\Big|_{\boldsymbol{\lambda}^{(s)}=\hat{\boldsymbol{\lambda}}^{(s)}} = 0$$

Therefore, the smoothing function is valid, i.e., is a ML estimation at current model parameters and interpolation weights.

8.2 Multiple-cluster MAP prior

For systems with a large number of parameters, Maximum a Posteriori (MAP) statistics have been previously used to improve standard MPE training [23], in which a static model is used to balance the ML-based statistics and parameters from an existing HMM set. Similarly, multiple-cluster MAP statistics may also be used instead of ML statistics as the multiple-cluster prior $\tilde{\mathbf{M}}^{(m)}$ and $\tilde{\mathbf{\Sigma}}^{(m)}$ in equation (49) in section 3.3.

To obtain the multiple-cluster MAP estimates $\tilde{\mathbf{M}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$, a new Normal-Wishart distribution for the MAP estimates is needed. One possible form is similar to the distribution (38) which employs standard single priors

$$\begin{aligned} \log p(\tilde{\mathcal{M}}) &= K - \frac{\tau^M}{2}\sum_{s,m}\tilde{\nu}_m^{(s)}\Big(\log|\tilde{\boldsymbol{\Sigma}}^{(m)}| + \mathrm{tr}\Big(\tilde{\boldsymbol{\Sigma}}_{MAP}^{(m)}\tilde{\boldsymbol{\Sigma}}^{(m)-1}\Big) \\ &\quad + \Big(\tilde{\mathbf{M}}^{(m)}\tilde{\boldsymbol{\lambda}}^{(s)} - \tilde{\boldsymbol{\mu}}_{MAP}^{(m)}\Big)^T\tilde{\boldsymbol{\Sigma}}^{(m)-1}\Big(\tilde{\mathbf{M}}^{(m)}\tilde{\boldsymbol{\lambda}}^{(s)} - \tilde{\boldsymbol{\mu}}_{MAP}^{(m)}\Big)\Big) \end{aligned} \tag{81}$$

where $\tilde{\boldsymbol{\mu}}_{MAP}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}_{MAP}^{(m)}$ are standard *single-cluster* prior parameters of the MAP distribution, which also have several options as discussed in section 3.3.2. $\tilde{\mathbf{M}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$ are the MAP estimates to be obtained, i.e., the multiple-cluster MAP prior.

As standard MAP update, to obtain the MAP estimates, a new auxiliary function is defined by adding an additional prior distribution term to the ML auxiliary function

$$\begin{aligned} \mathcal{Q}(\tilde{\mathcal{M}}) &= -\frac{1}{2}\sum_{s,m,t}\gamma_m^{ml}(t)\Big(\mathbf{o}(t) - \tilde{\mathbf{M}}^{(m)}\tilde{\boldsymbol{\lambda}}^{(s)}\Big)^T\tilde{\boldsymbol{\Sigma}}^{(m)-1}\Big(\mathbf{o}(t) - \tilde{\mathbf{M}}^{(m)}\tilde{\boldsymbol{\lambda}}^{(s)}\Big) \\ &\quad -\frac{1}{2}\sum_{s,m,t}\gamma_m^{ml}(t)\log|\tilde{\boldsymbol{\Sigma}}^{(m)}| + \log p(\tilde{\mathcal{M}}) \end{aligned} \tag{82}$$

where $\gamma_m^{ml}(t)$ is the ML posterior occupancy for component m. Note that the distribution $\log p(\tilde{\mathcal{M}})$ is defined for $\tilde{\mathbf{M}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$ and is different from the I-smoothing distribution (38). Note that, to get MAP estimate for model parameters, $\tilde{\boldsymbol{\lambda}}^{(s)} = \boldsymbol{\lambda}^{(s)}$.

By differentiating the above auxiliary function with respect to $\tilde{\mathbf{M}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$, the sufficient statistics for MAP estimates $\boldsymbol{\Theta}_{MAP} = \Big\{\gamma_m^{map}, \mathbf{G}_{MAP}^{(m)}, \mathbf{K}_{MAP}^{(m)}, \mathbf{L}_{MAP}^{(m)}\Big\}$ can be obtained as follows

$$\gamma_m^{map} = \gamma_m^{ml} + \tau^M \tag{83}$$

$$\mathbf{G}_{MAP}^{(m)} = \mathbf{G}_{ML}^{(m)} + \tau^M\sum_s\tilde{\nu}_m^{(s)}\tilde{\boldsymbol{\lambda}}^{(s)}\boldsymbol{\lambda}^{(s)T} \tag{84}$$

$$\mathbf{K}_{MAP}^{(m)} = \mathbf{K}_{ML}^{(m)} + \tau^M\Big(\sum_s\tilde{\nu}_m^{(s)}\tilde{\boldsymbol{\lambda}}^{(s)}\Big)\tilde{\boldsymbol{\mu}}_{MAP}^{(m)T} \tag{85}$$

$$\mathbf{L}_{MAP}^{(m)} = \mathbf{L}_{ML}^{(m)} + \tau^M\Big(\tilde{\boldsymbol{\mu}}_{MAP}^{(m)}\tilde{\boldsymbol{\mu}}_{MAP}^{(m)T} + \tilde{\boldsymbol{\Sigma}}_{MAP}^{(m)}\Big) \tag{86}$$

where τ^M is the specified parameter for the new Normal-Wishart distribution, γ^{ml}, $\mathbf{G}_{ML}^{(m)}$, $\mathbf{K}_{ML}^{(m)}$ and $\mathbf{L}_{ML}^{(m)}$ are ML statistics $\boldsymbol{\Theta}^{(ML)}$ as in section 2. Substituting the multiple-cluster MAP prior for multiple-cluster ML prior, equations (53) to (55) become

$$\tilde{\mathbf{G}}^{(m)} = \frac{1}{\gamma_m^{map}}\mathbf{G}_{MAP}^{(m)} \tag{87}$$

$$\tilde{\mathbf{K}}^{(m)} = \frac{1}{\gamma_m^{map}}\mathbf{K}_{MAP}^{(m)} \tag{88}$$

$$\tilde{\mathbf{L}}^{(m)} = \frac{1}{\gamma_m^{map}}\mathbf{L}_{MAP}^{(m)} \tag{89}$$

The whole sufficient statistics for I-smoothing distribution is still

$$\boldsymbol{\Theta}^{(p)} = \Big\{\tau^I, \tau^I\tilde{\mathbf{G}}^{(m)}, \tau^I\tilde{\mathbf{K}}^{(m)}, \tau^I\tilde{\mathbf{L}}^{(m)}\Big\}$$

There are two tunable parameters controlling prior occupancy: τ^I for the whole I-smoothing distribution and τ^M for distribution of MAP estimates. Observing the formulae, it can be found that if $\tau^M \to 0$, the MAP statistics degrades to multiple-cluster ML statistics as in section 3.3.1. If $\tau^M \to \infty$, the MAP statistics degrades to a standard single-cluster prior as in section 3.3.2. Therefore the multiple-cluster MAP statistics is actually a trade-off of the previous two forms of priors and τ^M is the coefficient to control the balance.

For discriminative update of interpolation weights, MAP estimate can also be selected as the prior parameters for I-smoothing distribution. The MAP distribution and auxiliary function to obtain the MAP estimate are the same as equation (81) and (82). However, the parameters to estimate is $\tilde{\boldsymbol{\lambda}}^{(s)}$ rather than $\tilde{\mathbf{M}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$. When using MAP estimate as the I-smoothing prior, it can be shown equation (70) become

$$\tilde{\mathbf{k}}^{(m)} = \frac{1}{\sum_{m,t} \gamma_m^{ml}(t) + \tau^M} \left(\sum_t \gamma_m^{ml}(t)\mathbf{o}(t) + \tau^M \tilde{\nu}_m \tilde{\boldsymbol{\mu}}^{(m)} \right) \tag{90}$$

Similar to I-smoothing for multiple-cluster model parameters, equation (90) is a trade-off between the multiple-cluster prior and the single-cluster prior. When $\tau^M \to 0$, the MAP prior (90) degrades to the ML prior (71). When $\tau^M \to \infty$, the MAP prior (90) degrades to the single-cluster form.