

# Context adaptive training with factorized decision trees for HMM-based statistical parametric speech synthesis

Kai Yu <sup>a</sup>, Heiga Zen <sup>b</sup>, François Mairesse <sup>a</sup> and Steve Young <sup>a</sup>

<sup>a</sup>*Machine Intelligence Lab, Cambridge University Engineering Department, Cambridge, CB2 1PZ, U.K.*

<sup>b</sup>*Toshiba Research Europe Ltd., Cambridge Research Laboratory, Cambridge, CB4 0GZ, U.K.*

---

## Abstract

To achieve natural high quality synthesized speech in HMM-based speech synthesis, the effective modelling of complex acoustic and linguistic contexts is critical. Traditional approaches use context-dependent HMMs with decision tree based parameter clustering to model the full combinatorial of contexts. However, weak contexts, such as word-level emphasis in natural speech, are difficult to capture using this approach. Also, due to combinatorial explosion, incorporating new contexts within the traditional framework may easily lead to the problem of insufficient data coverage. To effectively model weak contexts and reduce the data sparsity problem, different types of contexts should be treated independently. *Context adaptive training* provides a structured framework for this whereby standard HMMs represent normal contexts and transforms represent the additional effects of weak contexts. In contrast to speaker adaptive training in speech recognition, separate decision trees have to be built for different types of context factors. This paper describes the general framework of context adaptive training and investigates three concrete forms: MLLR, CMLLR and CAT based systems. Experiments on a word-level emphasis synthesis task show that all context adaptive training approaches can outperform the standard full-context-dependent HMM approach. However, the MLLR based system achieved the best performance.

*Key words:* HMM-based speech synthesis, context adaptive training, factorized decision tree, state clustering

*PACS:*

---

\* This research was partly funded by the UK EPSRC under grant agreement EP/F013930/1 and by the EU FP7 Programme under grant agreement 216594 (CLASSIC project: [www.classic-project.org](http://www.classic-project.org)).

## 1 Introduction

Statistical parametric speech synthesis [1] based on hidden Markov models (HMMs) [2] has grown in popularity in recent years. Based on the source-filter model assumption, phonetic and prosodic information are assumed to be conveyed in the spectral and excitation parameters. These spectral features, such as cepstral coefficients or line spectral pairs, and excitation features, such as fundamental frequency (also referred to as  $F_0$ ) and aperiodicities, can be extracted from a waveform using standard analysis techniques [3, 4]. A unified HMM framework may then be used to simultaneously model these parameters, where the spectrum and fundamental frequency are modelled as separate streams reflecting the fact that they are largely uncorrelated.<sup>1</sup> In the synthesis stage, given the context-dependent phoneme sequence generated from text analysis, a series of HMMs are concatenated and the speech features, spectrum and  $F_0$  parameters, are generated from the resulting composite HMM under consistency constraints required between static and dynamic features [5]. These speech parameters can then be converted to a waveform using a synthesis filter [6].

It is well known that the spectral and prosodic features of a particular phone in human speech are not only determined by the individual phonetic content, but also heavily affected by various background events associated with the phone. The background events which can affect the acoustic realization of a phone are referred to as its *contexts*. Compared to speech recognition, speech synthesis requires a much larger and more complex set of contexts to be represented in order to achieve high quality synthesized speech. Widely used contexts for speech synthesis include

- Identity of neighboring phones to the central phone. Normally, two phones to the left and the right of the center phone are considered as phonetic neighboring contexts.
- Position of phones, syllables, words, phrases with respect to higher level units.
- Number of phones, syllables, words, phrases with respect to higher level units.
- Syllable stress and accent status.
- Linguistic role, such as part-of-speech tag.

A number of other contexts, such as emotion, emphasis, etc. have also been used in HMM-based speech synthesis. To allow flexible modelling, even the center phone is regarded as context rather than the underlying distinct acoustic unit as in speech recognition. In a typical HMM-based speech synthesis system, there are normally around 50 different types of contexts. Compared to a typical tri-phone speech recognition system where there are only 2 types of contexts (left and right neighboring phones), this number is significantly larger. Hence, effective modelling of these complex context dependencies consequently becomes one of the most critical

---

<sup>1</sup> Other information such as an aperiodic component may also be modelled using additional streams within the HMM framework.

problems for HMM-based speech synthesis.

The traditional approach to handling complex contexts is to use a distinct HMM for each individual combination of possible contexts, referred to as a *context-dependent HMM*. The amount of available training data is normally not sufficient for robustly estimating all context-dependent HMMs since there is rarely sufficient data to cover all of the context combinations required. To address these problems, top-down decision tree based context clustering is widely used [7]. In this approach, the states of the context-dependent HMMs are grouped into “clusters” and the distribution parameters within each cluster are shared. The assignment of HMMs to clusters is performed by examining the context combination of each HMM through a binary decision tree, where one context-related yes/no question is associated with each non-leaf node. The number of clusters, namely the number of leaf nodes, determines the model complexity. The decision tree is constructed by sequentially selecting the questions which yield the largest likelihood increase of the training data. The size of the tree is controlled using a pre-determined threshold of likelihood increase or by introducing a model complexity penalty, such as the Bayesian information criterion (BIC) [8] or minimum description length (MDL) criterion [9]. With the use of context questions and state parameter sharing, the unseen contexts and data sparsity problems are effectively addressed. As the method has been successfully used in speech recognition, HMM-based speech synthesis naturally employs a similar approach to model very rich contexts. Compared to speech recognition, however, one decision tree is constructed for each stream at each state position to yield more flexibility.

Although context-dependent HMMs with decision tree-based state (stream) clustering can effectively model strong contextual effects, it is less able to model weak contexts, such as word-level emphasis in natural speech [10], because weak contexts have less influence on the likelihood of the data. When pooled with other more influential contexts, they are rarely selected during the decision tree construction. Consequently, the set of final clustered context-dependent HMMs has a poor representation of these contexts. One approach to address this problem is to split the decision tree construction into two stages. In the first stage, a decision tree is constructed using only the weak context questions. In the second stage, the remaining questions are used to further extend the decision tree [10]. Although this approach can effectively exploit weak context questions, it fragments the training data and leads to a reduction in the amount of the data that can be used in clustering the other contexts. Consequently, the quality of synthesized speech is degraded.

In this paper, an alternative approach, *context adaptive training* with factorized decision trees is presented for weak context modelling. In this approach, two separate sets of parameters are used to model the weak contexts and the normal contexts (such as phone or position) respectively. Standard HMM parameters are used for normal contexts, while a set of transforms are estimated for weak contexts. Full context-dependent HMMs are then constructed using the HMM parameters for nor-

mal contexts transformed by the weak-context specific transformations. Standard adaptive training techniques [11–13] can be used to perform interleaved updates of the two sets of model parameters. However, compared to adaptive training for speech recognition, in addition to the structured HMM representation using two sets of parameters, context adaptive training also requires changing the decision tree clustering process due to the nature of contexts being adapted. *Factorized decision trees* are used to fulfill this requirement. The basic idea is to construct decision trees for weak and normal contexts individually and then combine them to construct a structured common decision tree to represent the full context information [10, 14–17]. This paper describes the general framework of context adaptive training with factorized decision trees and investigates three specific implementations of systems: maximum likelihood linear regression (MLLR), constrained MLLR and cluster adaptive training (CAT). The effectiveness of these systems is evaluated on a synthesis task which requires that a specific word in the sentence is emphasized.

The rest of the paper is structured as follows. Section 2 describes the general framework of context adaptive training with factorized decision trees. Section 3 discusses concrete forms and implementation issues. Experimental results are given in section 4, followed by conclusions in section 5.

## 2 Context adaptive training with factorized decision tree

As discussed in section 1, full context-dependent HMMs may not effectively capture weak contexts and a two-pass decision tree fragments the data remaining for the normal contexts. *Context adaptive training* is proposed here to address these problems.

*Adaptive training* has been widely used in automatic speech recognition (ASR) to build compact acoustic models on non-homogeneous data [11–13]. A set of transforms are trained to represent different acoustic conditions, and canonical context-dependent HMMs represent the pure speech variabilities. The HMMs for a particular acoustic condition are constructed by adapting the canonical HMMs using the corresponding transforms. The two sets of parameters, transforms and HMMs, are updated in an interleaved fashion. Each update is done holding the other set of parameters fixed. When adaptive training is used in ASR, the acoustic conditions are normally defined for complete data blocks, such as speaker or noise environment. Recently, this adaptive training framework has also been used to model phonetic contexts [18, 19]. To obtain robust transform estimations, a regression tree is usually constructed to allow a group of Gaussian components to share a transform [20]. It is also possible to use a decision tree as the regression tree to determine transform sharing structure. The leaf nodes of the regression tree are referred to as *regression base classes*. Assuming there is only one Gaussian in each clustered state  $r_c$ , the

adapted Gaussian parameters can be represented as

$$\hat{\Lambda}_{r_c} = \mathcal{F}_{r_t}(\Lambda_{r_c}) \quad s.t. \quad r_c \subseteq r_t \quad (1)$$

where  $\Lambda_{r_c}$  is the Gaussian parameter set of state cluster  $r_c$ ,  $\hat{\Lambda}$  denotes the adapted parameters,  $\mathcal{F}_{r_t}(\cdot)$  is the transform associated with regression base class  $r_t$ . When modelling a range of acoustic conditions using adaptive training, transforms and canonical HMMs always use the same state tying structure. Hence,  $r_c$ , as the atomic cluster for adaptation, is always a subset of any transform regression base class  $r_t$ .

In contrast to ASR, contexts in HMM-based speech synthesis are much more complex as described in section 1. There are many factors which may affect the acoustic realization of phones. The prior knowledge of those factors form the questions used in the decision tree based state clustering procedure. Due to the nature of the factors, some questions are highly correlated, for example, the phonetic broad class questions and the syllable questions. However, other sets of questions are relatively weakly correlated, such as the phonetic broad class questions and the emphasis questions. The factors corresponding to the sets of questions can be regarded as independent. Also as mentioned before, the effect of weak contexts may not be found if they are pooled together with strong contexts during decision tree based clustering. It is important that weak contexts appear in the atomic state (stream) clusters for effective modelling. Therefore, not only the model parameter representation, but also the decision tree clustering process needs to be factorized in context adaptive training.

The idea of using factorized decision trees was used for acoustic modelling in recognition and synthesis [14–16] but has not been investigated within the framework of adaptive training until recently [10, 17]. In context adaptive training, the main purpose of constructing factorized decision trees is to fully model both sets of contexts and define atomic adaptation units where both normal and weak contexts can apply their effects. To achieve this goal, rather than pooling all context questions together to construct a single decision tree, two decision trees are built, with normal context questions (e.g. phone and position questions) and weak context questions (in this paper word-level emphasis) respectively. They are then combined to form a larger common decision tree by intersecting the leaf nodes. Taking word-level emphasis as the weak contexts and phonetic/position contexts as the normal contexts, the construction of the common decision tree is illustrated in figure 1.

The leaf nodes  $r_e$  and  $r_p$  are the final state (stream) clusters for emphasis and normal decision trees respectively, each of which is a set of shared clustered states.

$$r_e = \{\theta_1, \dots, \theta_{N_e}\} \quad r_p = \{\theta_1, \dots, \theta_{N_p}\} \quad (2)$$

where  $\theta$  is a distinct state corresponding to a specific context dependent model. The leaf nodes of the combined decision tree,  $r_c$  correspond to the intersections of the

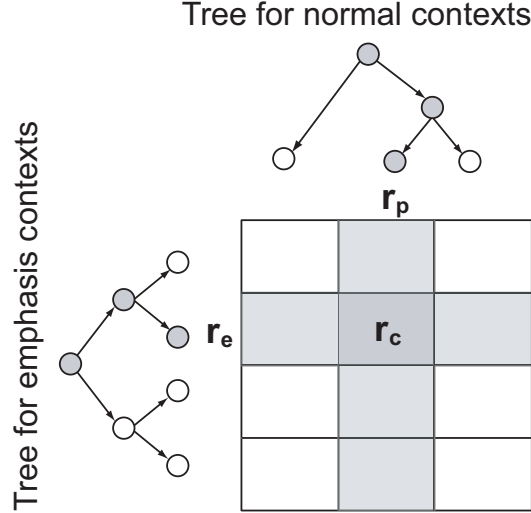


Fig. 1. Combination of normal and emphasis decision trees

leaf nodes of the emphasis decision tree  $r_e$  and the normal decision tree  $r_p$ , i.e.

$$r_c = \{\theta_1, \dots, \theta_{N_c}\} \quad s.t. \quad \theta_i \in r_e \text{ and } \theta_i \in r_p \quad i = 1, \dots, N_c \quad (3)$$

Hence,  $r_c$  are atomic units for adaptation, on which  $r_e$  and  $r_p$  will both have effect. Assuming there are  $N_e$  and  $N_p$  clustered states from the emphasis tree and normal decision tree respectively, the combined decision tree could have up to  $N_e \times N_p$  different context-dependent states. This structured representation is therefore more compact than direct full context-dependent modelling. With this factorized representation, both sets of contexts can make full use of all training data and can be extensively explored. Hence weak contexts can be effectively modelled without additionally fragmenting the training data.

Once the combined decision tree is constructed, the state output distributions, a single Gaussian in this paper, within  $r_c$  are tied. The Gaussian parameters can then be represented using a structured form similar to adaptive training

$$\hat{\Lambda}_{r_c} = \mathcal{F}_{r_e}(\Lambda_{r_p}) \quad s.t. \quad r_c = r_p \cap r_e \quad (4)$$

where  $r_e$  are the regression base classes, which are equivalent to the leaf nodes of the emphasis decision tree. Compared to equation (1), equation (4) has a different constraint on the regression base class. Due to the factorized decision tree clustering,  $r_e$  is not a subset of  $r_p$ , and hence not the atomic state cluster. Instead, the intersections of  $r_e$  and  $r_p$ , i.e.,  $r_c$ , are used as atomic parameter sharing units.

In adaptive training, both the transform and the canonical HMM may take various forms. Linear transforms [12, 21] and cluster adaptive training [13] are the two main categories. Equation (4) shows the general form of linear transform based context adaptive training. Cluster-based context adaptive training has the general

form shown in equation (5):

$$\hat{\Lambda}_{r_c} = \mathcal{F}_{r_t}(\Lambda_{r_p}, \Lambda_{r_e}) \quad s.t. \quad r_c = r_p \cap r_e \quad r_c \subseteq r_t \quad (5)$$

Here, the canonical model is a multiple-cluster Gaussian and the transform comprises the interpolation weights between the clusters. The multiple-cluster Gaussian for state  $r_c$  consists of cluster bases from different context groups  $r_p$  and  $r_e$ . Adaptation may then be performed on an additional full context specific regression class  $r_t$  constructed from the atomic state clusters  $r_c$ .

### 3 Implementation of context adaptive training

Section 2 has described the general framework of context adaptive training with factorized decision trees. This section will discuss concrete forms of context adaptive training and some implementation issues. In all derivations, *single Gaussians with diagonal covariance matrices* will be assumed, which is the widely used setup in HMM-based speech synthesis. The generalization to Gaussian mixture models is trivial and will not be explicitly discussed. As in HMM-based speech synthesis, spectrum and  $F_0$  are normally modelled using separate streams and factorized decision trees are then built for each corresponding stream. The discussion below will focus on the single stream case for clarity.

#### 3.1 MLLR based approach

Maximum likelihood linear regression (MLLR) [21] is a widely used type of transformation in speech recognition and synthesis. When single Gaussian distributions are used, the index of each distinct Gaussian,  $m$ , is equivalent to the index of the atomic state cluster  $r_c$ . For notational clarity,  $m$  will therefore be used instead of  $r_c$  in the following derivations. In MLLR based context adaptive training, the mean and covariance of Gaussian component  $m$  are represented by

$$\begin{aligned} \hat{\boldsymbol{\mu}}_m &= \mathbf{A}_{r_e(m)} \boldsymbol{\mu}_{r_p(m)} + \mathbf{b}_{r_e(m)} = \mathbf{W}_{r_e(m)} \boldsymbol{\xi}_{r_p(m)} \\ \hat{\boldsymbol{\Sigma}}_m &= \boldsymbol{\Sigma}_{r_p(m)} \end{aligned} \quad (6)$$

where  $r_p(m)$  and  $r_e(m)$  respectively represent the leaf nodes of normal and weak context decision trees to which component  $m$  belongs.  $\mathbf{A}_{r_e}$  and  $\mathbf{b}_{r_e}$  are weak-context transform parameters and  $\boldsymbol{\mu}_{r_p}$  and  $\boldsymbol{\Sigma}_{r_p}$  are Gaussian parameters corresponding to the normal contexts.  $\boldsymbol{\xi}_{r_p} = [\boldsymbol{\mu}_{r_p}^\top \ 1]^\top$  is the extended mean vector of leaf node  $r_p$  while  $\mathbf{W}_{r_e} = [\mathbf{A}_{r_e} \ \mathbf{b}_{r_e}]$  is the extended transform associated with leaf node  $r_e$ . From equation (6), the parameters of the combined leaf node cannot be directly estimated. Instead, they are constructed using two sets of parameters with different state clustering structures. With this factorized representation, the estimation of

the transform parameters for cluster  $r_e(m)$  and the Gaussian parameters for cluster  $r_p(m)$  must be interleaved. The detailed procedure is as follows:

- (1) Construct factorized decision trees for normal contexts ( $r_p$ ) and emphasis contexts ( $r_e$ ). Let  $m = r_e(m) \cap r_p(m)$  be the atomic state cluster (also atomic Gaussian in the single Gaussian case).
- (2) Get initial parameters of the atomic Gaussians from state clustering using normal decision tree and let  $\hat{\boldsymbol{\mu}}_m = \boldsymbol{\mu}_{r_p(m)}$ .
- (3) Estimate  $\mathbf{W}_{r_e}$  given the current model parameters  $\boldsymbol{\mu}_{r_p(m)}$  and  $\boldsymbol{\Sigma}_{r_p(m)}$ . This is the standard MLLR estimate [21]. The  $d^{\text{th}}$  row of  $\mathbf{W}_{r_e}$ ,  $\mathbf{w}_{r_e,d}^\top$ , is estimated as

$$\mathbf{w}_{r_e,d} = \mathbf{G}_{r_e,d}^{-1} \mathbf{k}_{r_e,d} \quad (7)$$

where the sufficient statistics for the  $d^{\text{th}}$  row are given by

$$\mathbf{G}_{r_e,d} = \sum_t \sum_{m \in r_e} \frac{\gamma_m(t)}{\sigma_{dd}^{r_p(m)}} \boldsymbol{\xi}_{r_p(m)} \boldsymbol{\xi}_{r_p(m)}^\top \quad (8)$$

$$\mathbf{k}_{r_e,d} = \sum_t \sum_{m \in r_e} \frac{\gamma_m(t) o_{t,d}}{\sigma_{dd}^{r_p(m)}} \boldsymbol{\xi}_{r_p(m)} \quad (9)$$

where  $o_{t,d}$  is the  $d^{\text{th}}$  element of observation vector  $\mathbf{o}_t$ ,  $\sigma_{dd}^{r_p(m)}$  is the  $d^{\text{th}}$  diagonal element of  $\boldsymbol{\Sigma}_{r_p(m)}$ ,  $r_p(m)$  is the leaf node of the normal decision tree to which Gaussian component  $m$  belongs,  $\gamma_m(t)$  is the posterior for Gaussian component  $m$  at time  $t$  which is calculated using the forward-backward algorithm with the parameters from equation (6).

- (4) Estimate  $\boldsymbol{\mu}_{r_p}$  given the emphasis transform parameters  $\mathbf{W}_{r_e}$ . This is similar to the mean update in speaker adaptive training [22]. Given the sufficient statistics

$$\begin{aligned} \mathbf{G}_{r_p} &= \sum_t \sum_{m \in r_p} \gamma_m(t) \mathbf{A}_{r_e(m)}^\top \boldsymbol{\Sigma}_m^{-1} \mathbf{A}_{r_e(m)} \\ \mathbf{k}_{r_p} &= \sum_t \sum_{m \in r_p} \gamma_m(t) \mathbf{A}_{r_e(m)}^\top \boldsymbol{\Sigma}_m^{-1} (\mathbf{o}_t - \mathbf{b}_{r_e(m)}) \end{aligned}$$

the new mean is then estimated by

$$\boldsymbol{\mu}_{r_p} = \mathbf{G}_{r_p}^{-1} \mathbf{k}_{r_p} \quad (10)$$

- (5) Given the updated mean  $\boldsymbol{\mu}_{r_p}$  and transform  $\mathbf{W}_{r_e}$ , perform context adaptation to get  $\hat{\boldsymbol{\mu}}_m$  using equation (6).
- (6) The re-estimation of  $\boldsymbol{\Sigma}_{r_p}$  is then performed using the standard covariance update formula with the adapted  $\hat{\boldsymbol{\mu}}_m$ . Here, the statistics are accumulated for each leaf node  $r_p$  rather than each individual component  $m$ .

$$\boldsymbol{\Sigma}_{r_p} = \text{diag} \left( \frac{\sum_{t,m \in r_p} \gamma_m(t) (\mathbf{o}_t - \hat{\boldsymbol{\mu}}_m) (\mathbf{o}_t - \hat{\boldsymbol{\mu}}_m)^\top}{\sum_{t,m \in r_p} \gamma_m(t)} \right) \quad (11)$$



where  $\gamma_m(t)$  is calculated using  $\hat{\boldsymbol{\mu}}_m$  constructed from the new estimate of  $\boldsymbol{\mu}_{\mathbf{r}_p}$  and  $\mathbf{W}_{\mathbf{r}_e}$ . It is worth noting that since data sufficiency is guaranteed during decision tree clustering for the normal context features, sharing covariance matrices within the leaf node  $\mathbf{r}_p$  will ensure robust estimation of the covariance matrices.

(7) Go to step 3 until convergence.

### 3.2 CMLLR based approach

Instead of MLLR, constrained MLLR (CMLLR) [12] can also be used to represent weak contexts. With CMLLR, both mean and covariance are adapted using the same linear transform as shown below

$$\begin{aligned}\hat{\boldsymbol{\mu}}_m &= \mathbf{A}'_{\mathbf{r}_e(m)} \boldsymbol{\mu}_{\mathbf{r}_p(m)} - \mathbf{b}'_{\mathbf{r}_e(m)} \\ \hat{\boldsymbol{\Sigma}}_m &= \mathbf{A}'_{\mathbf{r}_e(m)} \boldsymbol{\Sigma}_{\mathbf{r}_p(m)} \mathbf{A}'_{\mathbf{r}_e(m)\top}\end{aligned}\quad (12)$$

One advantage of using CMLLR is that CMLLR can be rewritten as a feature transform, which is represented as

$$\hat{\mathbf{o}}^{(m)} = \mathbf{A}_{\mathbf{r}_e(m)} \mathbf{o} + \mathbf{b}_{\mathbf{r}_e(m)} \quad (13)$$

where  $\mathbf{A}_{\mathbf{r}_e(m)} = \mathbf{A}'_{\mathbf{r}_e(m)-1}$  and  $\mathbf{b}_{\mathbf{r}_e(m)} = \mathbf{A}'_{\mathbf{r}_e(m)-1} \mathbf{b}'_{\mathbf{r}_e(m)}$  are the feature transforms dependent on specific Gaussian groups,  $\hat{\mathbf{o}}^{(m)}$  is the adapted observation to calculate the likelihood of  $\mathbf{o}$  with respect to Gaussian component  $m$ . The update formulae for  $\boldsymbol{\mu}_{\mathbf{r}_p}$  and  $\boldsymbol{\Sigma}_{\mathbf{r}_p}$  then take the standard form except that the adapted observations must be used and the statistics are accumulated for  $\mathbf{r}_p$  rather than the individual Gaussian components  $m$ . As in the MLLR case, the update formulae for CMLLR are similar to the standard update formulae in [12] except that the statistics are accumulated for  $\mathbf{r}_e$ . The overall procedure for CMLLR based context adaptive training is the same as for the MLLR case in section 3.1. Similarly, since the variance is shared at the  $\mathbf{r}_p$  level, there is no data sparsity issue for the covariance matrices update.

### 3.3 CAT based approach

The third implementation uses cluster adaptive training (CAT) [13]. As multiple-cluster HMMs are used here, the general form of equation (5) is used. In CAT based context adaptive training, the mean and covariance of Gaussian component  $m$  are represented by

$$\begin{aligned}\hat{\boldsymbol{\mu}}_m &= \lambda_{\mathbf{r}_t(m)}^{(p)} \boldsymbol{\mu}_{\mathbf{r}_p(m)} + \lambda_{\mathbf{r}_t(m)}^{(e)} \boldsymbol{\mu}_{\mathbf{r}_e(m)} = \mathbf{M}_m \boldsymbol{\lambda}_{\mathbf{r}_t(m)} \\ \hat{\boldsymbol{\Sigma}}_m &= \boldsymbol{\Sigma}_{\mathbf{r}_p(m)}\end{aligned}\quad (14)$$

where  $M_m = [\boldsymbol{\mu}_{r_p(m)} \ \boldsymbol{\mu}_{r_e(m)}]$  is the matrix representation of the two mean basis vectors for component  $m$ ,  $\boldsymbol{\mu}_{r_e(m)}$  is the weak-context basis,  $\boldsymbol{\lambda}_{r_t(m)} = [\lambda_{r_t(m)}^{(p)} \ \lambda_{r_t(m)}^{(e)}]^\top$  is the weight vector for component  $m$ , and  $\lambda_{r_t(m)}^{(p)}$  and  $\lambda_{r_t(m)}^{(e)}$  are weights for  $\boldsymbol{\mu}_{r_p(m)}$  and  $\boldsymbol{\mu}_{r_e(m)}$ , respectively. Similar to MLLR based context adaptive training, the estimation of the interpolation weights and Gaussian parameters must be interleaved. The difference here is mainly the initialization and parameter update formula. The detailed procedure is as follows:

- (1) Get initial parameters of  $\boldsymbol{\mu}_{r_p}$  and  $\Sigma_{r_p}$  from state clustering using the normal decision trees. Let  $\boldsymbol{\mu}_{r_e} = \mathbf{0}$  and  $\lambda_{r_t}^{(p)} = \lambda_{r_t}^{(e)} = 1$ .
- (2) Estimate  $\boldsymbol{\mu}_{r_p}$  and  $\boldsymbol{\mu}_{r_e}$  jointly given the current model parameters. Due to the intersection of  $\boldsymbol{\mu}_{r_p}$  and  $\boldsymbol{\mu}_{r_e}$ , the update formulae of cluster mean vectors for multiple-cluster HMMs in [13] must be modified [17]. Here, the mean vectors of all leaf nodes of both decision trees must be updated simultaneously.

$$\hat{\boldsymbol{\mu}} = \mathbf{G}^{-1} \mathbf{k} \quad (15)$$

where

$$\hat{\boldsymbol{\mu}} = \left[ \hat{\boldsymbol{\mu}}_{r_p=1}^\top \cdots \hat{\boldsymbol{\mu}}_{r_p=N^{(p)}}^\top \ \hat{\boldsymbol{\mu}}_{r_e=1}^\top \cdots \hat{\boldsymbol{\mu}}_{r_e=N^{(e)}}^\top \right]^\top$$

$$\mathbf{k} = \left[ \mathbf{k}_{r_p=1}^\top \cdots \mathbf{k}_{r_p=N^{(p)}}^\top \ \mathbf{k}_{r_e=1}^\top \cdots \mathbf{k}_{r_e=N^{(e)}}^\top \right]^\top$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{r_p=1} & \mathbf{0} & \mathbf{G}_{r_p=1, r_e=1} & \cdots & \mathbf{G}_{r_p=1, r_e=N^{(e)}} \\ & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{G}_{r_p=N^{(p)}} & \mathbf{G}_{r_p=N^{(p)}, r_e=1} & \cdots & \mathbf{G}_{r_p=N^{(p)}, r_e=N^{(e)}} \\ \mathbf{G}_{r_e=1, r_p=1} & \cdots & \mathbf{G}_{r_e=1, r_p=N^{(p)}} & \mathbf{G}_{r_e=1} & \mathbf{0} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{G}_{r_e=N^{(e)}, r_p=1} & \cdots & \mathbf{G}_{r_e=N^{(e)}, r_p=N^{(p)}} & \mathbf{0} & \mathbf{G}_{r_e=N^{(e)}} \end{bmatrix}$$

$$\mathbf{G}_{r_p} = \sum_t \sum_{m \in r_p} \gamma_m(t) \lambda_{r_t(m)}^{(p)} \Sigma_{r_p(m)}^{-1} \lambda_{r_t(m)}^{(p)}$$

$$\mathbf{G}_{r_e} = \sum_t \sum_{m \in r_e} \gamma_m(t) \lambda_{r_t(m)}^{(e)} \Sigma_{r_p(m)}^{-1} \lambda_{r_t(m)}^{(e)}$$

$$\mathbf{G}_{r_p, r_e} = \sum_t \sum_{m \in r_p \cap r_e} \gamma_m(t) \lambda_{r_t(m)}^{(p)} \Sigma_{r_p(m)}^{-1} \lambda_{r_t(m)}^{(e)} = \mathbf{G}_{r_e, r_p}^\top$$

$$\mathbf{k}_{r_p} = \sum_t \sum_{m \in r_p} \gamma_m(t) \lambda_{r_t(m)}^{(p)} \Sigma_{r_p(m)}^{-1} \mathbf{o}_t$$

$$\mathbf{k}_{r_e} = \sum_t \sum_{m \in r_e} \gamma_m(t) \lambda_{r_t(m)}^{(e)} \Sigma_{r_p(m)}^{-1} \mathbf{o}_t$$

Here  $r_p = n$  and  $r_e = m$  denotes  $n$ -th and  $m$ -th leaf nodes of normal and emphasis decision trees, respectively, and  $N^{(p)}$  and  $N^{(e)}$  correspond to the total numbers of leaf nodes of normal and emphasis decision trees.  $\mathbf{G}$  is the

sufficient statistics accumulated for the meta mean vector  $\hat{\boldsymbol{\mu}}$ .

- (3) Given the updated mean,  $\boldsymbol{\Sigma}_{\mathbf{r}_p}$  is re-estimated using the standard covariance update formula as in equation (11).
- (4) Given the updated mean and covariance,  $\boldsymbol{\lambda}_{\mathbf{r}_t}$  is estimated as

$$\boldsymbol{\lambda}_{\mathbf{r}_t} = \mathbf{G}_{\mathbf{r}_t}^{-1} \mathbf{k}_{\mathbf{r}_t} \quad (16)$$

$$\mathbf{G}_{\mathbf{r}_t} = \sum_t \sum_{m \in \mathbf{r}_t} \gamma_m(t) \mathbf{M}_m^\top \boldsymbol{\Sigma}_{\mathbf{r}_p(m)}^{-1} \mathbf{M}_m \quad (17)$$

$$\mathbf{k}_{\mathbf{r}_t} = \sum_{m \in \mathbf{r}_t} \mathbf{M}_m^\top \boldsymbol{\Sigma}_{\mathbf{r}_p(m)}^{-1} \sum_t \gamma_m(t) \mathbf{o}_t \quad (18)$$

where  $\gamma_m(t)$  is calculated using  $\hat{\boldsymbol{\mu}}_m$  constructed from the new estimate of  $\boldsymbol{\mu}_{\mathbf{r}_p}$ ,  $\boldsymbol{\mu}_{\mathbf{r}_e}$ , and  $\boldsymbol{\Sigma}_{\mathbf{r}_p}$ .  $\mathbf{G}_{\mathbf{r}_t}$  and  $\mathbf{k}_{\mathbf{r}_t}$  are sufficient statistics accumulated for all Gaussians within the regression baseclass  $\mathbf{r}_t$ .

- (5) Go to step 2 until convergence.

As the cluster mean vectors, covariance matrices, and interpolation weights are shared at  $\mathbf{r}_p$ ,  $\mathbf{r}_e$ , and  $\mathbf{r}_t$  levels, respectively, there is no data sparsity issue for updating these parameters.

### 3.4 State clustering in context adaptive training

The previous sections have discussed the training procedure of MLLR, CMLLR and CAT based context adaptive training. Factorization of the decision trees also impacts on the clustering process itself since the structured Gaussian parameter representation complicates the computation of the data likelihood.

The basic idea of decision tree based state clustering is to use a binary decision tree, in which a question is attached to each non-leaf node, to assign the state distribution of every possible full context HMM model to a state cluster [7]. As discussed before, the tree is built using a top-down sequential optimization procedure. All states are pooled in the root node and then the node is split into two by finding a context question which partitions the states in the parent node so as to give the maximum likelihood increase. When using a single Gaussian as the state output distribution and assuming that tying states does not change the frame/state alignment, then considering that the Gaussian parameters  $\boldsymbol{\mu}(\boldsymbol{\theta})$  and  $\boldsymbol{\Sigma}(\boldsymbol{\theta})$  are ML estimates, the log likelihood of a set of states  $\boldsymbol{\theta}$  can be represented as

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_t \sum_{\theta \in \boldsymbol{\theta}} \gamma_\theta(\mathbf{o}_t) \log \mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{\Sigma}(\boldsymbol{\theta})) \quad (19)$$

$$= -\frac{\gamma(\boldsymbol{\theta})}{2} \left( \log |\boldsymbol{\Sigma}(\boldsymbol{\theta})| + D \log(2\pi) + D \right) \quad (20)$$

where  $D$  is the data dimension,  $\gamma(\boldsymbol{\theta})$  and  $\boldsymbol{\Sigma}(\boldsymbol{\theta})$  are the total occupancy and the

covariance matrix of the pooled state respectively:

$$\gamma(\boldsymbol{\theta}) = \sum_{\theta \in \boldsymbol{\theta}} \left( \sum_t \gamma_{\theta}(\mathbf{o}_t) \right) \quad (21)$$

$$\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \sum_{\theta \in \boldsymbol{\theta}} \left( \sum_t \gamma_{\theta}(\mathbf{o}_t) \right) \left( \boldsymbol{\mu}_{\theta}^{\top} \boldsymbol{\mu}_{\theta} + \boldsymbol{\Sigma}_{\theta} \right). \quad (22)$$

Equations (20) and (22) rely on the fact that  $\boldsymbol{\mu}_{\theta}$  and  $\boldsymbol{\Sigma}_{\theta}$  are standard ML estimates.

When using a structured context adaptive training representation, there are two sets of parameters to be clustered: transform and Gaussian parameters, resulting in two or more decision trees. There are three ways to build these trees

- **Independent construction** assumes that the factorized decision trees are independent of each other and are therefore built separately. This is an approximation which is simple and efficient to implement. It results in a factorization that is purely dependent on the different sets of context questions used during the decision tree construction.
- **Dependent construction** builds the factorized decision trees one-by-one. Each tree is built assuming that the remaining parameter sets and the sharing structure are fixed. An iterative process must be used in this case with all parameters being re-estimated after every split.
- **Simultaneous construction** builds all factorized decision trees in one go. At each split, all trees are optimized inter-dependently until the stopping criterion is met.

The choice of tree building strategy depends on the trade-off between computational cost and model accuracy. For context adaptive training, equations (20) and (22) do not hold naturally for either transform or Gaussian parameters due to the structured parameter representation. As the two sets of parameters are dependent on each other, dependent or simultaneous construction will involve re-estimation of both sets of parameters at each split. This will result in very high computational cost, especially for linear transform based approaches. Hence, in this paper, independent construction is employed. In the case of cluster-based context adaptive training, dependent and simultaneous constructions of cluster-dependent decision trees have been derived [17, 23]. However, in this paper, independent construction is employed for consistency.

In HMM-based speech synthesis, the decision tree based state clustering is usually performed twice to get better state/stream clustering structure. The general procedure is as below:

- (1) Train mono-phone HMMs and construct untied full context dependent HMMs
- (2) Perform one Expectation Maximization (EM) re-estimation of the untied full context dependent HMMs

- (3) Perform state/stream clustering given the state alignment and the parameters of the untied model in step 2 [7]
- (4) Perform several iterations of EM re-estimation of the clustered HMMs
- (5) Untie the clustered HMMs and perform one further EM re-estimation to get updated parameters of the untied full context dependent HMMs
- (6) Perform state/stream clustering given the state alignment and the parameters of the untied model in step 5 [7]
- (7) Perform several iterations of EM re-estimation of the clustered HMMs

Due to the structured modelling of context adaptive training, not only the EM re-estimation of step 4 and step 7 involves two sets of model parameters, the construction of the second untied full context HMMs in step 5 also needs to take into account both sets of parameters. In this paper, a common procedure is used for step 5. The clustered HMMs are first untied and during re-estimation context transforms are applied as the HMM model re-estimation in step 4. The statistics are then accumulated at *untied* state/stream level and used to reestimate the single Gaussian parameters for each untied state/stream <sup>2</sup>.

## 4 Experiments

### 4.1 Experimental conditions

The context adaptive training techniques were evaluated in a *natural word level emphasis* synthesis task [10]. The training data is a subset of the male English voice with a Scottish accent (*awb*) in the CMU ARCTIC speech database [25]. One judge annotated the 597 utterances of the set A of the dataset, by labelling the word(s) that were perceived as the focus of the utterance based on the natural emphasis of the speaker.<sup>3</sup> It is worth noting that there was no intention of collecting speech with emphasis during the construction of the ARCTIC speech database, hence, it does not contain strong stylistic variation. The emphasis labels were given to the naturally emphasized words (e.g., content words) as well as involuntary fluctuations of the speaker. The judge labelled 2.32 emphasized words per utterance on average (26.3% of the words). In order to assess the reliability of the annotation, a second judge annotated a subset of 50 utterances of the 597 sentences. This yielded an agreement of 1.04 words per utterance on average, and a disagreement of 1.52 words on average.<sup>4</sup> This suggests that the natural emphasis information obtained from a human judge is highly subjective. However, most of the disagreements were

<sup>2</sup> In the CAT case, as the target single Gaussian only has one mean vector, two-model re-estimation is needed to change the HMM structure [24].

<sup>3</sup> Available at <http://mi.eng.cam.ac.uk/~farm2/emphasis>.

<sup>4</sup> Cohen's Kappa cannot be used here because the phrases are not distinct elements.

due to a difference of granularity when labelling emphasis, as there was an overlap for 72% of the utterances. This shows that there exists consistent *rough agreement* over natural emphasis. Though emphasis is likely to be harder to capture when it is not explicitly generated in natural speech, techniques that can extract the emphasis component from natural data can significantly reduce the cost of stylistic modelling.

Altogether four systems were built, three context adaptive training systems as described in section 3 and a standard full context-dependent HMM system which uses both normal and emphasis contexts in state clustering.<sup>5</sup> All systems were built using a modified version of the HMM-based speech synthesis toolkit (HTS) [26]. The HMM with globally tied distribution (HMM-GTD) technique for  $\log F_0$  modelling [27] was used as it yielded better speech quality. Six emphasis contexts were used to form questions for emphasis decision tree construction.<sup>6</sup> They (including one more question about inexistence of emphasis) include

- (1) Whether the previous, the current and the next words are emphasized
- (2) Whether the previous and the next words are emphasized
- (3) Whether the previous and the current words are emphasized
- (4) Whether the next word is emphasized
- (5) Whether the previous word is emphasized
- (6) Whether the current word is emphasized
- (7) Whether none of the previous, the current and the next words is emphasized

The static feature set comprised 25 mel-cepstral coefficients [28] including the zero-th coefficients,  $\log F_0$  and aperiodic energy components in five frequency bands (0 to 1, 1 to 2, 2 to 4, 4 to 6 and 6 to 8 kHz). All features were extracted using STRAIGHT [3]. A five state, left-to-right HMM structure with no skip transitions was used. During HMM training, the stream weight for the aperiodic component was set to zero. Hence, the forward-backward alignment depends only on the spectral and  $F_0$  features. Statistics for updating parameters of aperiodic components were however collected and their parameters were updated in the standard way. The MDL criterion was used to stop growing the decision trees. The MLLR and the CMLLR systems used block diagonal transforms (3 blocks) for the spectrum and full transforms for  $\log F_0$  and aperiodic component features. The CAT system used fixed global weights ( $\forall_m \lambda_{r_t(m)} = [1.0 \ 1.0]^T$ ) to interpolate mean vectors of normal and emphasis contexts. In this paper, context adaptive training techniques were only applied to the spectrum,  $\log F_0$  and aperiodic components, while duration was still modelled using standard full context-dependent HMMs. The speech

<sup>5</sup> The purpose of this experiment is to compare different approaches for complex context modelling. Though emphasis word adaptation can be an alternative approach for emphasis synthesis as shown in [10], it is not a generally applicable approach for modelling complex contexts, such as phone positions. Hence, it is not considered in this paper.

<sup>6</sup> Each emphasis related question consists of one emphasis context feature and one normal context feature. This will lead to powerful transforms as the number of transforms is large.

parameter generation algorithm considering global variance [29] was used during synthesis.

## 4.2 Experimental results

A subjective listening test was performed to measure the ability to convey word-level emphasis. For each system, 10 utterances in the tourist information domain were first generated without any emphasized words. The same utterances were generated again but with *one* word emphasized, forming 10 contrasting pairs (e.g., ‘Char Sue is an *expensive* Chinese restaurant’). The words to emphasize were selected randomly from the content words in the sentence which carry semantic information. The contrasting waveform pairs from the four systems were then provided to listeners. Hence, altogether each listeners listened to 40 contrasting utterance pairs. When perceiving a difference of emphasis, the listener was asked to select the word that carried the emphasis, otherwise to indicate that there is no perceivable emphasis. Altogether 14 listeners, 7 native and 7 non-native, participated in the test. The performance of emphasized word detection is shown in table 1.

| System | # Det | Rec (%) | Pre (%) | <i>F</i> -measure |
|--------|-------|---------|---------|-------------------|
| GMM    | 2.0   | 20.0    | 53.8    | 0.29              |
| MLLR   | 4.7   | 47.1    | 68.0    | 0.56              |
| CMLLR  | 3.2   | 32.1    | 68.2    | 0.44              |
| CAT    | 2.9   | 28.6    | 69.0    | 0.40              |

Table 1

Average number of correctly detected emphasized words and Recall, Precision and *F*-measure of emphasis detection.

The row labelled GMM in table 1 is the standard full context-dependent system. It can be observed that all context adaptive training systems obtained better emphasis detection performance than the standard full context-dependent HMMs. A pair-wise two-tailed Student’s t-test was performed to evaluate the statistical difference of the average number of correctly detected emphasized words<sup>7</sup>. It was found that the improvements of the context adaptive training systems compared to the standard system were all significant at the 95% confidence level. Amongst the different forms of context adaptive training, the MLLR system achieved the best performance while the CMLLR and the CAT systems were similar. As there is only one emphasized word per utterance, the recall rate of emphasis detection

<sup>7</sup> If the word selected by the listener is the one actually being emphasized by the synthesis system, it is regarded as a *correctly detected* emphasized word. Selection of either the wrong emphasized word selection or selecting the no emphasis option is regarded as a detection error

is proportional to the number of detected emphasized words. As for precision, all context adaptive training approaches outperformed standard full context modelling and there is no significant difference between the different types of transformation. Due to the difference in recall rate, the  $F$ -measure for the MLLR-based approach performs the best overall.

Model complexity may be one reason for the performance difference between context adaptive training systems. Emphasis is mainly carried by  $\log F_0$  features and the number of  $\log F_0$  states in the factorized decision trees are shown in table 2.<sup>8</sup>

| System | $\#r_p$ | $\#r_e$ | $\#r_c = r_p \cap r_e$ | #para ( $\times 10^3$ ) |
|--------|---------|---------|------------------------|-------------------------|
| GMM    | 2286    |         | —                      | 13.7                    |
| MLLR   | 2445    | 2988    | 17739                  | 48.7                    |
| CMLLR  | 1959    | 2458    | 13361                  | 40.1                    |
| CAT    | 2402    | 2894    | 16307                  | 23.1                    |

Table 2

Number of  $\log F_0$  states in the factorized decision trees.

In table 2,  $r_p$  and  $r_e$  are the numbers of clustered states in the normal and emphasis decision trees of the  $\log F_0$  stream, respectively.  $r_c$  is the set of leaf nodes of the intersection of the two trees, which is the number of atomic units for adaptation. The last column gives the total number of free parameters, consisting of both HMM and transform parameters. For comparison, the number of parameters of the standard GMM system is also shown where there is only one set of parameters (context-dependent HMMs). Due to the structured modelling which makes uses of additional parameters, all context adaptively trained systems have more parameters. It is worth noting that due to the use of regression base classes during the estimation of MLLR and CMLLR, the actual number of emphasis transforms is smaller than the number of emphasis decision tree leaf nodes ( $\#r_e$ ). It can also be observed that the CAT system has far fewer parameters than the MLLR system. This is mainly because global weights are used and there is only one basis vector associated with each emphasis state  $r_e$ . This significant reduction of parameters may then limit its power to transform normal contexts to full contexts. In contrast, there is no large difference in the number of parameters between the CMLLR system and the MLLR system. The performance degradation may therefore come from other aspect.

It can be observed that the numbers of the clustered states of the CMLLR system are much smaller than the MLLR and the CAT system. This implies there might be some issue related to the clustering process. As described in section 3.4, a common state/stream clustering procedure was used for all systems in this paper. At the second stage, the untied full context model was estimated given the adapted

<sup>8</sup> The ordering of complexity for spectrum and aperiodic component features are similar to those shown for  $F_0$  in table 2.



model parameters. However, in the CMLLR case, since it is implemented as a feature transform, the parameter estimation also used the transformed observations in equation (13). When multiple regression base classes are used, the determinants of different CMLLR transforms may then affect the likelihood calculation using equation (20) during state clustering. It was found that after the second state clustering stage, the MLLR system received an increased number of clustered states, while the CMLLR system received a notable decrease. This shows that the effect of feature transform determinants cannot be ignored and is likely to be the main reason for the performance difference between the CMLLR and the MLLR system. Alternative state clustering procedure for feature-based context transforms will be investigated in future work.

## 5 Conclusions

This paper has described a context adaptive training framework for modeling complex contexts in HMM-based speech synthesis. Two sets of parameters are constructed to represent the different context groups (normal and weak emphasis contexts in this paper) and are estimated inter-dependently. In contrast to adaptive training for speech recognition, decision tree clustering must be modified for context adaptive training in order to avoid data sparsity issues and to ensure that weak contexts are not overwhelmed by the strong contexts. In this paper, this problem has been solved using a novel factorized decision tree approach whereby separate decision trees are built for the normal and the weak contexts. Context adaptation is then performed for the intersections of the two trees. This approach allows for fine-grained control of any dimension of interest, which is important for expressive TTS in order to make spoken language interfaces better able to convey the linguistic and social cues required in normal human-human conversation. Three forms of context adaptive training systems, MLLR, CMLLR and CAT, have been investigated. Experiments using a word-level emphasis synthesis task have shown that context adaptive training significantly outperforms standard full context HMMs with the MLLR system showing the best performance overall. The results presented were, however, based on the use of a simple independent factorized state clustering scheme and it is possible that the approximations involved may compromise performance. Future work will therefore investigate the impact of using dependent and simultaneous state clustering schemes. Also, whether to associate a particular context factor to the base HMMs or to the transforms is an open problem for future research.

## References

- [1] H. Zen, K. Tokuda, A. Black, Statistical parametric speech synthesis, *Speech Communication* 51 (11) (2009) 1039–1064.
- [2] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, T. Kitamura, Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis, in: *Proc. Eurospeech*, 1999, pp. 2347–2350.
- [3] H. Kawahara, I. Masuda-Katsuse, A. Cheveigné, Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based  $f_0$  extraction: possible role of a repetitive structure in sounds, *Speech Communication* 27 (1999) 187–207.
- [4] H. Zen, T. Toda, M. Nakamura, K. Tokuda, Details of Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005, *IEICE Trans. Inf. & Syst. E-90D* (1) (2007) 325–333.
- [5] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, T. Kitamura, Speech parameter generation algorithms for HMM-based speech synthesis, in: *Proc. ICASSP*, 2000, pp. 1315–1318.
- [6] S. Imai, Cepstral analysis synthesis on the mel frequency scale, in: *Proc. ICASSP*, 1983, pp. 93–96.
- [7] S. Young, J. Odell, P. Woodland, Tree-based state tying for high accuracy acoustic modelling, in: *ARPA Workshop on Human Language Technology*, 1994, pp. 307–312.
- [8] W. Chou, W. Reichl, Decision tree state tying based on penalized Bayesian information criterion, in: *Proc. ICASSP*, 1999, pp. 345–348.
- [9] K. Shinoda, T. Watanabe, Acoustic modeling based on the MDL principle for speech recognition, in: *Proc. EUROSPEECH*, 1997, pp. 99–102.
- [10] K. Yu, F. Mairesse, S. Young, Word-level emphasis modelling in HMM-based speech synthesis, in: *Proc. ICASSP*, 2010, pp. 4238–4241.
- [11] T. Anastasakos, J. Mcdonough, R. Schwartz, J. Makhoul, A compact model for speaker adaptive training, in: *Proc. ICSLP*, 1996, pp. 1137–1140.
- [12] M. Gales, Maximum likelihood linear transformations for HMM-based speech recognition, *Computer Speech & Language* 12 (2) (1998) 75–98.
- [13] M. Gales, Cluster adaptive training of hidden Markov models, *IEEE Trans. Speech and Audio Processing* 8 (4) (2000) 417–428.
- [14] N. Iwahashi, Y. Sagisaka, Statistical modelling of speech segment duration by constrained tree regression, *Trans. of IEICE E83-D* (2000) 1550–1559.
- [15] Y. Nankaku, K. Nakamura, H. Zen, K. Tokuda, Acoustic modeling with contextual additive structure for HMM-based speech recognition, in: *Proc. ICASSP*, 2008, pp. 4469–4472.

- [16] K. Saino, A clustering technique for factor analyzed voice models, Master thesis, Nagoya Institute of Technology, (in Japanese) (2008).
- [17] H. Zen, N. Braunschweiler, Context-dependent additive log  $F_0$  model for HMM-based speech synthesis, in: Proc. of Interspeech, 2009, pp. 2091–2094.
- [18] D. Povey, L. Burget, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. Goel, M. Karafiat, A. Rastrow, R. Rose, P. Schwarz, S. Thomas, Subspace Gaussian mixture models for speech recognition, in: Proc. ICASSP, 2010, pp. 4330–4333.
- [19] M. Gales, K. Yu, Canonical state models for automatic speech recognition, in: Proc. Interspeech, 2010, pp. 58–61.
- [20] M. Gales, The generation and use of regression class trees for MLLR adaptation, Tech. Rep. CUED/F-INFENG/TR263, Cambridge University Engineering Department (1996).
- [21] C. Leggetter, P. Woodland, Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models, Computer Speech & Language 9 (1995) 171–185.
- [22] T. Anastasakos, J. Mcdonough, R. Schwartz, J. Makhoul, A compact model for speaker adaptive training, in: Proc. ICSLP, 1996, pp. 1137–1140.
- [23] H. Zen, Speaker and language adaptive training for HMM-based polyglot speech synthesis, in: Proc. Interspeech, 2010, pp. 410–413.
- [24] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X.-Y. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, P. Woodland, The HTK Book (for HTK version 3.4), Cambridge University Engineering Department, 2009.
- [25] J. Kominek, A. Black, CMU ARCTIC databases for speech synthesis, Tech. Rep. CMU-LTI-03-177, Carnegie Mellon University (2003).
- [26] K. Tokuda, K. Oura, K. Hashimoto, H. Zen, J. Yamagishi, T. Toda, T. Nose, S. Sako, A. Black, The HMM-based speech synthesis system, <http://hts.sp.nitech.ac.jp/>.
- [27] K. Yu, T. Toda, M. Gasic, S. Keizer, F. Mairesse, B. Thomson, S. Young, Probabilistic modelling of  $f_0$  in unvoiced regions in HMM based speech synthesis, in: Proc. ICASSP, 2009, pp. 3773–3776.
- [28] T. Fukada, K. Tokuda, T. Kobayashi, S. Imai, An adaptive algorithm for mel-cepstral analysis of speech, in: Proc. ICASSP, 1992, pp. 137–140.
- [29] T. Toda, K. Tokuda, A speech parameter generation algorithm considering global variance for HMM-based speech synthesis, IEICE Trans. Inf. Syst. E90-D (5) (2007) 816–824.