Khorram *et al. EURASIP Journal on Audio, Speech, and Music Processing* 2014, **2014**:12
http://asmp.eurasipjournals.com/content/2014/1/12

EURASIP Journal on Audio, Speech, and Music Processing
a SpringerOpen Journal

RESEARCH Open Access

Context-dependent acoustic modeling based on hidden maximum entropy model for statistical parametric speech synthesis

Soheil Khorram[1*], Hossein Sameti[1], Fahimeh Bahmaninezhad[1], Simon King[2] and Thomas Drugman[3]

Abstract

Decision tree-clustered context-dependent hidden semi-Markov models (HSMMs) are typically used in statistical parametric speech synthesis to represent probability densities of acoustic features given contextual factors. This paper addresses three major limitations of this decision tree-based structure: (i) The decision tree structure lacks adequate context generalization. (ii) It is unable to express complex context dependencies. (iii) Parameters generated from this structure represent sudden transitions between adjacent states. In order to alleviate the above limitations, many former papers applied multiple decision trees with an additive assumption over those trees. Similarly, the current study uses *multiple decision trees* as well, but instead of the additive assumption, it is proposed to train the smoothest distribution by *maximizing entropy* measure. Obviously, increasing the smoothness of the distribution improves the context generalization. The proposed model, named *hidden maximum entropy model (HMEM)*, estimates a distribution that maximizes entropy subject to multiple moment-based constraints. Due to the simultaneous use of multiple decision trees and maximum entropy measure, the three aforementioned issues are considerably alleviated. Relying on HMEM, a novel speech synthesis system has been developed with maximum likelihood (ML) parameter re-estimation as well as maximum output probability parameter generation. Additionally, an effective and fast algorithm that builds multiple decision trees in parallel is devised. Two sets of experiments have been conducted to evaluate the performance of the proposed system. In the first set of experiments, HMEM with some heuristic context clusters is implemented. This system outperformed the decision tree structure in small training databases (i.e., 50, 100, and 200 sentences). In the second set of experiments, the HMEM performance with four parallel decision trees is investigated using both subjective and objective tests. All evaluation results of the second experiment confirm significant improvement of the proposed system over the conventional HSMM.

Keywords: Hidden Markov model (HMM)-based speech synthesis; Context-dependent acoustic modeling; Decision tree-based context clustering; Maximum entropy; Overlapped context clusters; Statistical parametric speech synthesis

1 Introduction

Statistical parametric speech synthesis (SPSS) has dominated speech synthesis research area over the last decade [1,2]. It is mainly due to SPSS advantages over traditional concatenative speech synthesis approaches; these advantages include the flexibility to change voice characteristics [3-5], multilingual support [6-8], coverage of acoustic space [1], small footprint [1], and robustness [4,9]. All of the above advantages stem from the fact that SPSS provides a statistical model for acoustic features instead of using original speech waveforms. However, these advantages are achieved at the expense of one major disadvantage, i.e., degradation in the quality of synthetic speech [1]. This shortcoming results from three important factors: vocoding distortion [10-13], accuracy of statistical models [14-25], and accuracy of parameter generation algorithms [26-28]. This paper is an attempt to alleviate the second factor and improve the accuracy of statistical models. Most of the researches carried out to improve the acoustic modeling performance aimed to develop systems that generate natural and high-quality speech using large training speech databases (more than 30 min) [18,21,22]. Nevertheless,

* Correspondence: khorram@ce.sharif.edu
[1]Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
Full list of author information is available at the end of the article

there exist a great number of under-resourced languages (such as Persian) for which only limited amount of data are available. To alleviate this shortcoming, we target developing a statistical approach that leads to an appropriate speech synthesis system not only with large but also with small training databases.

Every SPSS system consists of two distinct phases, namely training and synthesis [1,2]. In the training phase, first acoustic and contextual factors are extracted for the whole training database using a vocoder [12,29,30] and a natural language pre-processor. Next, the relationship between acoustic and contextual factors is modeled using a context-dependent statistical approach [14-25]. Synthesis phase starts with a parameter generation algorithm [26-28] that exploits trained context-dependent statistical models and aims to generate realistic acoustic feature trajectories for a given input text. Acoustic trajectories are then fed into the same vocoder used during the training phase in order to generate the desired synthesized speech.

In the most predominant statistical parametric approach, spectrum, excitation, and duration of speech are expressed concurrently in a unified framework of *context-dependent multi-space probability distribution hidden semi-Markov model (HSMM)* [14]. More specifically, a multi-space probability distribution [17] is estimated for each leaf node of decision trees [31]. These decision tree-based structures split contextual space into a number of non-overlapped clusters which form multiple groups of context-dependent HMM states, and each group shares the same output probability distribution [31]. In order to capture acoustic variations accurately, the model has to be able to express a large number of robust distributions [19,20]. Decision trees are not efficient for such expression because increasing the number of distributions by growing the tree reduces the population of each leaf and consequently reduces the robustness of the distributions. This problem stemmed from the fact that decision tree assigns each HMM state to an only one cluster (small region in contextual space), therefore, each state contributes in modeling just one distribution. In other words, the decision tree structure makes the models match training data just in non-overlapped regions which are expressed through decision tree terminal nodes [31]. In the case of limited training data, the decision tree would be small, so it cannot split contextual factor space sufficiently. In this case, the accordance between model and data is not sufficient, and therefore, the speech synthesis system generates unsatisfactory output. Accordingly, it is clear that by extending the decision tree in such a way that each state affects multiple distributions (larger portion of the contextual space), the generalization to unseen models will be improved. The main idea of this study is to extend non-overlapped regions of one decision tree to overlapped regions of multiple decision trees and hence exploit contextual factors more efficiently.

A large number of research works have already been performed to improve the quality of basic decision tree-clustered HSMM. Some of them are based on a model adaptation technique. This latter method exploits an invaluable prior knowledge attained from an average voice model [3], and adapts this general model using an adaptation algorithm such as *maximum likelihood linear regression (MLLR)* [32], *maximum a posteriori (MAP)* [33], and *cluster adaptive training (CAT)* [21]. However, working with average voice models is difficult for under-resourced languages since building such general model needs remarkable efforts to design, record, and transcribe a thorough multi-speaker speech database [3]. To alleviate the data sparsity problem in under-resourced languages, speaker and language factorization (SLF) technique can be used [34]. SLF attempts to factorize speaker-specific and language-specific characteristics in training data and then model them using different transforms. By representing the speaker attributes by one transform and language characteristics by a different transform, the speech synthesis system will be able to alter language and speaker separately. In this framework, it is possible to exploit the data from different languages to predict speaker-specific characteristics of the target speaker, and consequently, the data sparsity problem will be alleviated. Authors in [15,16] also developed a new technique by replacing maximum likelihood (ML) point estimate of HSMM with a *variational Bayesian* method. Their system was shown to outperform HSMM when the amount of training data is small. Other notable structures used to improve statistical modeling accuracy are *deep neural networks (DNNs)* [18]. The decision tree structure is not efficient enough to model complicated context dependencies such as XORs or multiplexers [18]. To model such complex contextual functions, the decision tree has to be excessively large, but DNNs are capable to model complex contextual factors by employing multiple hidden layers. Additionally, a great number of overlapped contextual factors can be fed into a DNN to approximate output acoustic features, so DNNs are able to provide efficient context generalization. Speech synthesis based on Gaussian process regression (GPR) [35] is another novel approach that has recently been proposed to overcome HMM-based speech synthesis limitations. The GPR model predicts frame-level acoustic trajectories from frame-level contextual factors. The frame-level contextual factors include the relative position of the current frame within the phone and some articulatory information. These frame-level contextual factors are employed as the explanatory variable in GPR. The frame-level modeling of GPR removes the inaccurate stationarity assumption of state output distribution in HMM-based speech synthesis. Also, GPR can directly represent the complex context dependencies without using parameter tying by decision tree clustering; therefore, it is capable of improving context generalization.

Acoustic modeling with *contextual additive* structure has also been proposed to represent dependencies between contextual factors and acoustic features more precisely [19,20,23,32,36-40]. In this structure, acoustic trajectories are considered to be a sum of independent acoustic components which have different context dependencies (different decision trees have to be trained for those components). Since the mean vectors and covariance matrices of the distribution are equal to the sum of mean vectors and covariance matrices of additive components, the model would be able to exploit contextual factors more efficiently. Furthermore, in this structure, each training data sample contributes to modeling multiple mean vectors and covariance matrices. Many papers applied the additive structure just for F0 modeling [37-40]. Authors in [37] proposed an additive structure with multiple decision trees for mean vectors and a single tree for variance terms. In this paper, for different additive components, different sets of contextual factors were used and multiple trees were built simultaneously. In [40], multiple additive decision trees are also employed, but they train this structure using minimum generation error (MGE) criterion. Sakai [38] defines an additive model with three distinct layers, namely intonational phrase, word-level, and pitch-accent layers. All of these components were trained simultaneously using a regularized least square error criterion. Qian et al. [39] propose to use multiple additive regression trees with a gradient-based tree-boosting algorithm. Decision trees are trained in successive stages to minimize the error squares. Takaki et al. [19,20] applied additive structure for spectral modeling and reported that the computational complexity of this structure is extremely high for full context labels as used in speech synthesis. To alleviate this issue, they proposed two approaches: covariance parameter tying and a likelihood calculation algorithm using matrix inversion lemma [19]. Despite all the advantages, this additive structure may not match training data accurately because once training is done, the first and second moments of the training data and model may not be exactly the same in some regions.

Another important problem of conventional decision tree-clustered acoustic modeling is difficulty in capturing the effect of weak contextual factors such as word-level emphasis [23,36]. It is mainly because weak contexts have less influence on the likelihood measure [23]. One clear approach to address this issue is to construct the decision tree in two successive steps [36]. In the first step, all selections are done among weak contextual factors, and in the second step, the remaining questions are adopted [36]. This procedure can effectively exploit weak contextual factors, but it leads to a reduction in the amount of training data available for normal contextual factors. Context adaptive training with factorized decision trees [23] is another approach that can exploit weak context questions efficiently. In this system, a canonical model is trained using normal contextual factors and then a set of transforms is built by weak contextual factors. In fact, canonical models and transforms, respectively, represent the effects of normal and weak contextual factors [23]. However, this structure also improves context generalization of conventional HMM-based synthesis by exploiting adaptation techniques.

This paper introduces a *maximum entropy model* (*MEM*)-based speech synthesis. MEM [41] has been demonstrated to be positively effective in numerous applications of speech and natural language processing such as speech recognition [42], prosody labeling [43], and part-of-speech tagging [44]. Accordingly, the overall idea of this research is to improve HSMM context generalization by taking advantage of a distribution which not only matches training data in many overlapped contextual regions but also is optimum in the sense of an entropy criterion. This system has the potential to model the dependencies between contextual factors and acoustic features such that each training sample contributes to train multiple sets of model parameters. As a result, context-dependent acoustic modeling based on MEM could lead to a promising synthesis system even for limited training data.

The rest of the paper is organized as follows. Section 2 presents HSMM-based speech synthesis. The hidden maximum entropy model (HMEM) structure and the proposed HMEM-based speech synthesis system are explained in Section 3. Section 4 is dedicated to experimental results. Finally, Section 5 concludes this paper.

2 HSMM-based speech synthesis

This section aims to explain the predominant statistical modeling approach applied in speech synthesis, i.e., *context-dependent multi-space probability distribution left-to-right without skip transitions HSMM* [3,14] (simply called HSMM in the remainder of this paper). The discussion presented in this section provides a preliminary framework which will be used as a basis to introduce the proposed HMEM technique in Section 3. The most significant drawback of HSMM, namely inadequate context generalization, is also pointed out.

2.1 HSMM structure

HSMM is a hidden Markov model (HMM) having explicit state duration distribution instead of self-state transition probabilities. Figure 1 illustrates the standard HSMM. As it can be observed, HSMM initially partitions acoustic parameter (observation) trajectories into a fixed number of time slices (so-called states) in order to moderate the undesirable influence of non-stationarity. Note that state durations are latent variables and have to be trained in an

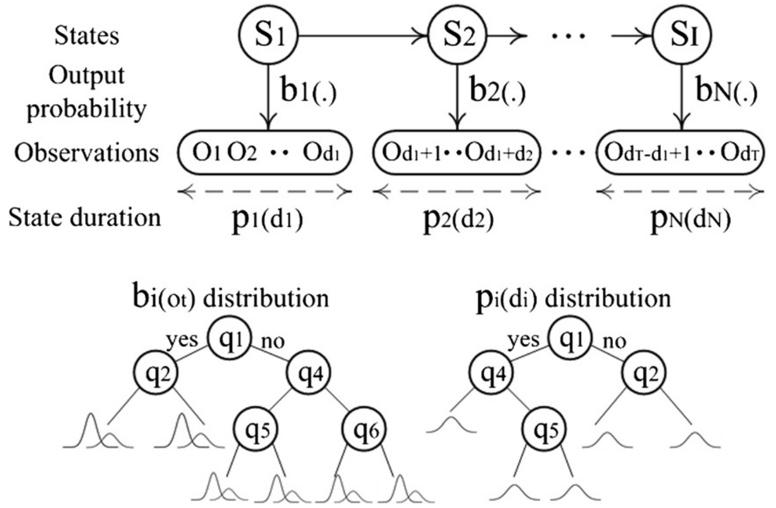


Figure 1 **HSMM structure and its output and duration probability distribution.**

unsupervised manner. An N-state HSMM λ is specified by a set of state output probability distributions $\{b_i(\cdot)\}_{i=1}^{N}$ and a complementary set of state duration probability distributions $\{p_i(\cdot)\}_{i=1}^{N}$. To model these distributions, a number of distinct decision trees are used for output and duration probability distributions. Conventionally, different trees are trained for different states [31]. These trees cluster the whole contextual factor space into a large number of tiny regions which are expressed by terminal nodes. Thereafter, in each terminal node, the output distribution $b_i(\cdot)$ is modeled by a multi-space probability distribution, and similarly, a typical Gaussian distribution is considered for the duration probability $p_i(\cdot)$ [14].

To handle the absence of fundamental frequency in unvoiced regions, multi-space probability distribution (MSD) is used for output probability distribution [17]. In accordance with commonly used synthesizers, this paper assumes that acoustic sample space consists of G spaces. Each of these spaces, specified by an index g, represents an n_g dimensional real space, i.e., $\mathscr{R}^{n_g}$. Each observation vector o_t has a probability w_g to be generated by the gth space iff the dimensionality of o_t is identical to n_g. In other words, we have

$$\begin{aligned} b_i(o_t) &= \sum\nolimits_{g \epsilon S(o_t)} w_{ig} b_{i|g}(o_t), \\ b_{i|g}(\mathrm{o}_t) &= \mathcal{N}_{n_g}\left(o_t; \mu_{ig}, \Sigma_{ig}\right), \end{aligned} \tag{1}$$

$$p_i(d) = \mathcal{N}_1\left(d; m_i, \sigma_i^2\right), \tag{2}$$

where $S(o_t)$ represents a set of all space indexes with the same dimensionality of o_t, and where $\mathcal{N}_l(.;\mu,\Sigma)$ denotes an l-dimensional Gaussian distribution with mean μ, and covariance matrix Σ ($\mathcal{N}_0$ is defined to be 1). Furthermore, the output probability distribution of the ith state and gth space is denoted by $b_{i|g}(o_t)$ which is a Gaussian distribution with mean vector μ_{ig} and covariance matrix Σ_{ig}. Also, m_i and σ_i^2 represent mean and variance of the state duration probability.

Regarding the method for providing context dependency, it should be noted that HSMM normally offers binary decision trees and acoustic models are established for each leaf of these trees, separately [45,46]. Suppose f and L are contextual functions based on a decision tree Υ and are defined as

$$\begin{aligned} f_l(i;\Upsilon) &\overset{\text{def}}{=} \begin{cases} 1 \text{ if context of the } i\text{th state} \in l\text{th leaf of } \Upsilon \\ 0 \text{ if context of the } i\text{th state} \notin l\text{th leaf of } \Upsilon \end{cases}, \\ L(\Upsilon) &\overset{\text{def}}{=} \text{number of leaves in } \Upsilon \end{aligned} \tag{3}$$

Applying the above functions, all model parameters of Equations 1 and 2 can be expressed by linear combinations of model parameters defined for each terminal node. More precisely,

$$\begin{aligned} \Sigma_{ig} &= \sum\nolimits_{l=1}^{L(\Upsilon_o)} f_l(i;\Upsilon_o)\Sigma_{ig}^l, \\ \mu_{ig} &= \sum\nolimits_{l=1}^{L(\Upsilon_\mathrm{o})} f_l(i;\Upsilon_o)\mu_{ig}^l, \\ w_{ig} &= \sum\nolimits_{l=1}^{L(\Upsilon_o)} f_l(i;\Upsilon_o)w_{ig}^l, \\ \sigma_i &= \sum\nolimits_{l=1}^{L(\Upsilon_o)} f_l(\mathrm{i};\Upsilon_d)\sigma_i^l, \\ m_i &= \sum\nolimits_{\mathrm{l}=1}^{L(\Upsilon_o)} f_l(i;\Upsilon_d)m_i^l, \end{aligned} \tag{4}$$

where $\mathrm{Y_o}$ and $\mathrm{Y_d}$ are decision trees trained for modeling output observation vectors and state durations. All symbols with superscript l indicate model parameters defined for the lth leaf.

2.2 HSMM likelihood

Having described the HSMM structure, we can now probe the exact expression for model likelihood or the probability of the observation sequence $O = [o_1, o_2, \ldots, o_T]$ as [14]:

$$P(O|\lambda) = \sum_{i=1}^{N} \sum_{j=1,j\neq i}^{N} \sum_{d=1}^{t} \alpha_{t-d}(j) p_i(d) \prod_{s=t-d+1}^{t} b_i(o_s) \beta_t(i) \tag{5}$$

where this equality is valid for every value of $t \in [1,T]$. Also, $\alpha_t(i)$ and $\beta t(i)$ are partial forward and backward probability variables that are calculated successively from their previous or next values as follows [3,14]:

$$\alpha_t(i) = \sum_{d=1}^{t} \sum_{j=1,j\neq i}^{N} \alpha_{t-d}(j) p_i(d) \prod_{s=t-d+1}^{t} b_i(o_s), \tag{6}$$

$$\beta_t(i) = \sum_{d=1}^{T-t} \sum_{j=1,j\neq i}^{N} p_j(d) \prod_{s=t+1}^{t+d} b_j(o_s) \beta_{t+d}(j), \tag{7}$$

where the initial forward and backward variables for every state indexes i are $\alpha_0(i)$-1 and $\beta_T(i) = 1$.

2.3 HSMM parameter re-estimation

The ML criterion is commonly used to estimate model parameters of HSMM. However, we are not aware of latent variables, i.e., state durations and space indexes; therefore, an expectation maximization (EM) algorithm has to be adopted. Applying EM algorithm leads to the following re-estimation formulas [14]:

$$\begin{aligned}
\hat{\mu}_{ig}^{l} &= \frac{\sum_k f_l(i;\Upsilon_o) \sum_{t=1}^{T} \gamma_t(i,g) o_t}{\sum_k f_l(i;\Upsilon_o) \sum_{t=1}^{T} \gamma_t(i,g)}, \\
\hat{\sum}_{ig}^{l} &= \frac{\sum_k f_l(i;\Upsilon_o) \sum_{t=1}^{T} \gamma_t(i,g) \left(o_t - \hat{\mu}_{ig}^{l}\right) \left(o_t - \hat{\mu}_{ig}^{l}\right)^T}{\sum_k f_l(i;\Upsilon_o) \sum_{t=1}^{T} \gamma_t(i,g)}, \\
\hat{w}_{ig}^{l} &= \frac{\sum_k f_l(i;\Upsilon_o) \sum_{t=1}^{T} \gamma_t(i,g)}{\sum_{h=1}^{G} \sum_k f_l(i;\Upsilon_o) \sum_{t=1}^{T} \gamma_t(i,h)}, \\
\hat{m}_{i}^{l} &= \frac{\sum_k f_l(i;\Upsilon_d) \sum_{t=1}^{T} \sum_{d=1}^{t} \chi_t^d(i) d}{\sum_k f_l(i;\Upsilon_d) \sum_{t=1}^{T} \sum_{d=1}^{t} \chi_t^d(i)}, \\
\hat{\sigma}_{i}^{l^2} &= \frac{\sum_k f_l(i;\Upsilon_d) \sum_{t=1}^{T} \sum_{d=1}^{t} \chi_t^d(i) \left(d - \hat{m}_i^l\right)^2}{\sum_k f_l(i;\Upsilon_d) \sum_{t=1}^{T} \sum_{d=1}^{t} \chi_t^d(i)},
\end{aligned} \tag{8}$$

where $\gamma_t(i,g)$ denotes the posterior probability of being in state i and space g at time t, and $\chi_t^d(i)$ is the probability of occupying the ith state from time t–$d + 1$ to t. The following equations calculate the above probabilities:

$$\begin{aligned}
{}_{\gamma}t(i,g) &= \frac{1}{P(o|\lambda)} \sum_{t0=1}^{t-1} \sum_{t1=\mathrm{t}}^{T} \sum_{j=1,j\neq i}^{N} \alpha_{t-d}(j) p_i(d) b_{i|g}(o_t) \\
&\prod_{s=t0,s\neq t}^{t1} b_i(o_s) \beta_{t_1}(i), x_t^d(i) = \frac{1}{p(o|\lambda)} \\
&\sum_{j=1,j\neq i}^{N} \alpha_{t-d}(j) p_i(d) \prod_{s=t-d+1}^{t} b_i(o_s) \beta_t(i).
\end{aligned} \tag{9}$$

2.4 Inefficient context generalization

A major drawback of decision tree-clustered HSMM can now be clarified. Suppose we have only two real contextual factors, f_1 and f_2. Figure 2 shows a sample decision tree and the regions represented by its terminal nodes. By training HSMM, the model matches training data in all non-overlapped regions expressed by the terminal nodes. However, there is no guarantee that this accordance is held for overlapped regions such as the region R in Figure 2.

It can be noticed from the definition of function $f_i(c;\mathrm{Y})$ in Equation 3 that this function can be viewed as a set of $L(Y)$ *non-overlapped* binary contextual factors. The fact that these contextual factors are non-overlapped leads to the insufficient context generalization, because this fact makes each training sample contribute to the model of only one leaf and only one Gaussian distribution. Hence, by extending $f_i(c;\mathrm{Y})$ to *overlapped* contextual factors, more efficient context generalization capabilities could be achieved. Section 3 proposes an approach which enables the conventional structure to model the overlapped contextual factors and thus improves the modeling performance of unseen contexts.

3. Hidden maximum entropy model

The goal of this section is to develop a context-dependent statistical model for acoustic parameters with adequate context generalization. The previous section on HSMM revealed that inappropriate generalization stemmed from the application of *non-overlapped* features only. Consequently, relating acoustic parameters to contextual information by incorporating *overlapped* features could improve generalization efficiency. This section proposes *HMEM* to establish this relation.

3.1 HMEM structure

The proposed HMEM technique exploits exactly the same structure and graphical model as the original HSMM, and thus, the model likelihood expression given by Equation 5 is also valid for HMEM. The only difference between HSMM and HMEM is the way they incorporate contextual factors in output and duration probability distributions

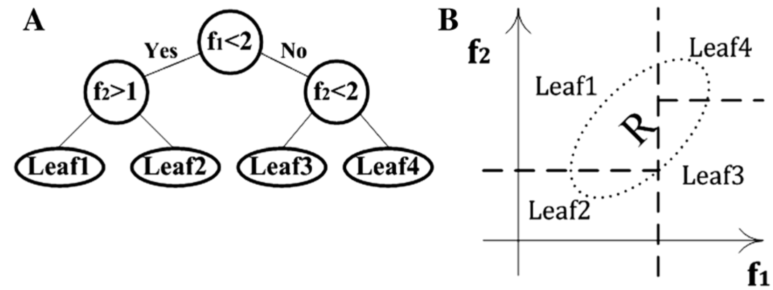


Figure 2 Sample decision tree and regions represented by its terminal nodes. (A) An example of decision tree with just three questions. **(B)** Regions that are classified by the tree and an arbitrary region (named R).

(i.e., $\{b_i(\cdot)\}_{i=1}^{N}$, $\{p_i(\cdot)\}_{i=1}^{N}$). HSMM builds a decision tree and then trains a Gaussian distribution for each leaf of the tree. On the contrary, HMEM obeys the maximum entropy modeling approach which will be described in the next subsection.

3.1.1 Maximum entropy modeling

Let us now derive a simple maximum entropy model. Suppose an ℓ-dimensional random vector process with output x that may be influenced by some contextual information c. Our target is to construct a stochastic model that precisely predicts the behavior of x, when c is given, i.e., $P(x|c)$. Maximum entropy principle first imposes a set of constraints on $P(x|c)$ and then chooses a distribution as close as possible to a uniform distribution by maximizing the entropy criterion [41]. In fact, this method will find the least biased distribution among all distributions that satisfy our constraints. In other words,

$$\hat{P}(x|c) \overset{\text{def}}{=} \operatorname{argmax}_P \mathscr{H}(P) \text{ subject to a set of constraint,} \tag{10}$$

employed constraints make the model preserve some context-dependent statistics of the training data. $\mathscr{H}(P)$ represents entropy criterion [41] that is calculated as

$$\mathscr{H}(P) \overset{\text{def}}{=} -\int_x \sum\nolimits_{\text{all possible } c} P(x,c) \log P(x,c) \mathrm{dx}. \tag{11}$$

Computing the above expression is extremely complex because there are a large number of contextual factors and all possible values of c are not calculable. However, authors in [41] applied the following approximation for $P(x,c)$:

$$P(x,c) = \tilde{P}(c)P(x|c). \tag{12}$$

where $\tilde{P}(c)$ denotes empirical probability which can be calculated directly using the training database [41]. The above approximation simplifies the entropy expression as

$$\mathscr{H}(P) = -\int_x \sum\nolimits_{\text{all } c \text{ in database}} \tilde{P}(c)P(x|c) \log P(x|c)dx \\ -\sum\nolimits_{\text{all c in database}} \tilde{P}(c) \log \tilde{P}(c), \tag{13}$$

where the second term is constant and does not affect the optimization problem. Therefore, we have

$$\mathscr{H}(P) = -\sum\nolimits_{\text{all } c \text{ in database}} \tilde{P}(c) \int_x P(x|c) \log P(x|c)dx. \tag{14}$$

Additionally, we adopt a set of L_f predefined binary contextual factors, $f_l(c)$, and another set of L_g binary contextual factors, $g_l(c)$, that both of them may be highly overlapped. In order to obtain a Gaussian distribution for $\hat{P}(x|c)$ and extend the conventional HSMM distribution, first- and second-order context-dependent moments expressed in Equation 14 are considered for the constraints.

$$\hat{P}(x|c) \overset{\text{def}}{=} \operatorname{argmax}_P \mathscr{H}(P) \tag{15}$$

subject to following constraints:

$$\left\{ \begin{array}{c} \forall 1 \le l \le L_f \; E\{f_l(c)x\} = \tilde{E}\{f_l(c)x\} \\ \forall 1 \le l \le L_g \; E\{g_l(c)x\,x^T\} = \tilde{E}\{g_l(c)x\,x^T\} \\ \text{For all possible } c \; \int_x P(x|c)dx = 1 \end{array} \right\},$$

where E and $\hat{E}$ indicate real and empirical mathematical expectations given in the following equations:

$$\tilde{E}\{f_l(c)x\} = \sum\nolimits_{\text{all } c \text{ in database}} \tilde{P}(c) f_l(c) x(c), \tag{16}$$

$$E\{f_l(c)x\} = \sum\nolimits_{\text{all } c \text{ in database}} \tilde{P}(c) f_l(c) \int_x xP(x,c)dx$$

where $x(c)$ denotes the realization of ℓ-dimensional random vector x for the context c in the database. If there are multiple realizations for x, $x(c)$ will be obtained by taking the average over those values. In sum, the proposed context-dependent acoustic modeling approach obtains the smoothest (maximum entropy) distribution that captures first-order moments of training data in L_f regions indicated by $\{f_l(c)\}_{l=1}^{L_f}$ and second-order moments of data computed in $\{g_l(c)\}_{l=1}^{L_g}$.

In order to solve the optimization problem expressed by Equation 10, the Lagrange multipliers method is applied. This method defines a new optimization function as follows:

$$\hat{P}(x|c) = \mathrm{argmax}_P \mathscr{H}(P) + \sum\nolimits_{l=1}^{L_f} u_l^T \left(E\{f_l(c)x\} - \tilde{E}\{f_l(c)x\}\right) + \sum\nolimits_{l=1}^{L_g} \left(E\{g_l(c)x^T H_l x\} - \tilde{E}\{g_l(c)x^T H_l x\}\right), \tag{17}$$

where u_l denotes a vector of Lagrange multipliers for satisfying the lth first-order moment constraints and H_l is a matrix of Lagrange multipliers for satisfying the lth second-order moment constraints. Taking derivatives of the above function with respect to P leads to the following equality.

$$\sum\nolimits_{\text{all } c} \tilde{P}(c) \int_x \left(-\log P(x|c) + u^T x + x^T H x + \text{const.}\right) dx = 0$$
$$H \overset{\text{def}}{=} \sum\nolimits_{l=1}^{L_g} g_l(c) H_l,\ u \overset{\text{def}}{=} \sum\nolimits_{l=1}^{L_f} f_l(c) u_{l.} \tag{18}$$

Therefore, one possible solution that maximizes entropy with the constraint of Equation 15 using Lagrange multipliers can be expressed as:

$$\hat{P}(x|c) = \frac{1}{\left(\det\left(2\pi H^{-1}\right)\right)^{0.5}} \exp \times \left[-\frac{1}{2}\left(x + \frac{1}{2}H^{-1}u\right)^T H \left(x + \frac{1}{2}H^{-1}u\right)\right],$$
$$H \overset{\text{def}}{=} \sum\nolimits_{l=1}^{L_g} g_l(c) H_l,\ u \overset{\text{def}}{=} \sum\nolimits_{l=1}^{L_f} f_l(c) u_{l,} \tag{19}$$

where H_l and u_l are model parameters related to the lth contextual factors $g_l(c)$ and $f_l(c)$, respectively. H_l is an ℓ-by-ℓ matrix and u_l is an ℓ-dimensional vector. When $f_l(c)$ becomes 1 (i.e., it is active), u_l affects the distribution; otherwise, it has no effect on the distribution. In fact, Equation 19 is nothing but the well-known Gaussian distribution with mean vector $-0.5H^{-1}u$, and covariance matrix H^{-1}, both calculated from a specific context-dependent combination of model parameters. Indeed, the main difference of MEM in comparison with other methods such as spectral additive structure [19,20] is that mean and variance in MEM are not a linear combination of other parameters. This type of combination enables MEM to match training data in all overlapped regions.

This form of context-dependent Gaussian distribution presents a promising flexibility in utilizing contextual information. On one hand, using detailed and non-overlapped contextual factors such as features defined by Equation 3 (decision tree terminal node indicators) generates context-dependent Gaussian distributions which are identical to those used in conventional HSMM. These distributions have straightforward and efficient training procedure but suffer from insufficient context generalization capabilities. On the other hand, incorporating general and highly overlapped contextual factors overcomes the latter shortcoming and provides efficient context generalization, but its training procedure becomes more computationally complex. In the case of highly overlapped contextual factors, an arbitrary context activates several contextual factors, and hence, each observation vector is involved in modeling several model parameters.

3.1.2 ME-based modeling vs. additive modeling

At first glance, the contextual additive structure [19,20,32,37] seems to have the same capabilities as the proposed ME-based context-dependent acoustic modeling. Therefore, to clarify their differences, this section compares HMEM with the additive structure through a very simple example.

In this example, the goal is to model a one-dimensional observation value using both ME-based modeling and a contextual additive structure. Due to the prime importance of mean parameters in HMM-based speech synthesis [47], we investigate the difference between mean values predicted by two systems.

Figure 3A shows a three-dimensional contextual factor space (c_1-c_2-c_3) which is clustered by an additive structure. The additive structure consists of three different additive components with three different decision trees, namely Q_1, Q_2, and Q_3. Each tree has a simple structure with just one binary question that splits a specific dimension of the contextual factor space into two regions. Each region is represented by a leaf node, and inside that leaf node, a mean parameter of each additive component is written. As it is depicted in the figure, these trees split contextual factor space into eight different cubic clusters. Mean values estimated for these cubic clusters are computed by adding mean values of additive components.

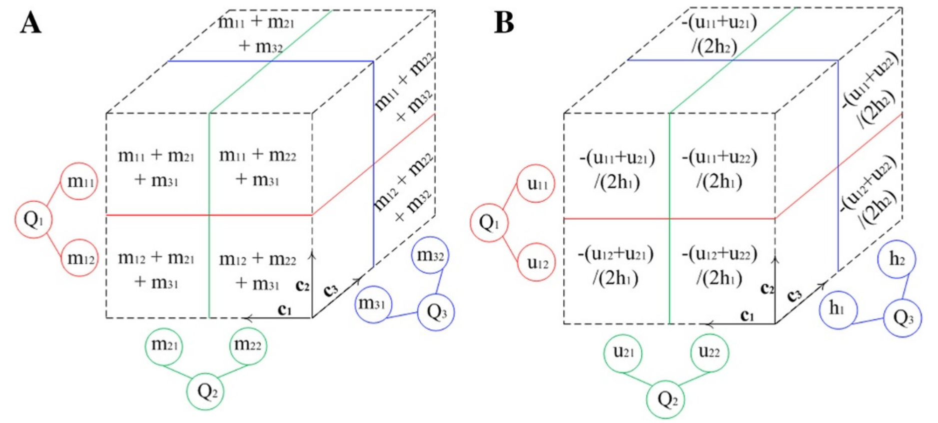


Figure 3 Contextual factor space clustered by (A) contextual additive structure and (B) ME-based context-dependent modeling.

In contrast, Figure 3B shows the corresponding ME-based modeling approach. In the previous subsection, it is described that ME-based context-dependent modeling needs two sets of regions, $\{f_l(c)\}_{l=1}^{L_f}$ and $\{g_l(c)\}_{l=1}^{L_g}$. This example assumes that the leaves of Q_1 and Q_2 are defined as the first set of regions $\{f_l(c)\}_{l=1}^{L_f}$, and the leaves of Q_3 are defined as the second set $\{g_l(c)\}_{l=1}^{L_g}$. Therefore, according to the explanation of the previous subsection, first empirical moments of Q_1 and Q_2, in addition to the second empirical moments of Q_3, are captured by ME-based modeling. Figure 3B shows the estimated model mean values for all eight cubic clusters. As it is realized from the figure, model mean values estimated by ME-based modeling is a combination of adding parameters live in the regions $\{f_l(c)\}_{l=1}^{L_f}$ divided by the parameters defined for the regions $\{g_l(c)\}_{l=1}^{L_g}$. In fact, the proposed ME-based modeling is an extension to the additive structure that ties all covariance matrices [19]. This extension is clear because if $\{g_l(c)\}_{l=1}^{L_g}$ is defined with one region containing all contextual feature space, the ME-based modeling converts to the additive structure that ties all covariance matrices [19].

3.1.3 HMEM-based speech synthesis

HMEM improves both state duration distribution $\{p_i(\cdot)\}_{i=1}^{N}$ and output observation distribution $\{b_i(\cdot)\}_{i=1}^{N}$ using maximum entropy modeling. According to the discussion presented in Section 3.1.1, MEM requires two sets of contextual factors. In this section, for the sake of simplicity, it is assumed that the contextual regions defined for first-order moment constraints $\{f_l(c)\}_{l=1}^{L_f}$ are identical to the regions defined for second-order moment constraints $\{g_l(c)\}_{l=1}^{L_g}$. All equations presented in this section is based on this assumption; however, their extension to the general case (different $\{f_l(c)\}_{l=1}^{L_f}$ and $\{g_l(c)\}_{l=1}^{L_g}$) is straightforward. Therefore, we define $f_l^{\,d}(i)$ and $f_l^{\,o}(i)$ as L^d and L^o contextual factors which are designed carefully for the purpose of modeling duration and acoustic parameters of the ith state. Maximum entropy criterion leads to the following duration and output probability distributions.

$$b_i(o_t) = \sum\nolimits_{g\epsilon S(o_t)} w_{ig} b_{i|g}(o_t),$$

$$b_{i|g}(o_t) = \mathcal{N}_{n_g}\left(o_t; -\frac{1}{2} u_{ig} H_{ig}, H_{ig}^{-1}\right),$$

$$P_i(d) = \mathcal{N}_1\left(d; -\frac{1}{2} u_i h_i, \frac{1}{h_i}\right),$$

$$u_i = \sum\nolimits_{l=1}^{L^d} \mathrm{f}_1^{\mathrm{d}}(i) u_i^l, \qquad h_i = \sum\nolimits_{l=1}^{L^d} f_l^{\,d}(i) h_i^l,$$

$$u_{ig} = \sum\nolimits_{l=1}^{L^o} f_l^{\,o}(i) u_{ig}^l, \qquad H_{ig} = \sum\nolimits_{l=1}^{L^o} f_l^{\,o}(i) H_{ig}^l,$$

$$w_{ig} = \frac{\exp\left(\sum\nolimits_{l=1}^{L^o} f_l^{\,o}(i) w_{ig}^l\right)}{\sum\nolimits_{g=1}^{G} \exp\left(\sum\nolimits_{\mathrm{i}=1}^{L^o} f_l^{\,o}(i) w_{ig}^l\right)}. \tag{20}$$

In these equations, $S(o_t)$ is a set of all possible spaces defined for o_t. u_i^l and h_i^l are the duration model parameters, and w_{ig}^l, u_{ig}^l, and H_{ig}^l denote the output model parameters related to the lth contextual factor, gth space, and ith state.

We can now probe the differences between HSMM and HMEM context-dependent acoustic modeling. These two modeling approaches are dramatically close to each other, so that defining HMEM contextual factors based on the decision trees described by Equation 3 would reduce HMEM to HSMM. Accordingly, HMEM extends HSMM and enables its structure to exploit overlapped contextual factors.

Moreover, another significant conclusion that could be drawn from this section is that several HSMM concepts are transposable within the HMEM framework. These concepts involve Viterbi algorithm, methods which calculate forward/backward variables and occupation probabilities, and even all parameter generation algorithms [26-28]. It just needs to define mean vectors, covariance matrices, and space probabilities of HSMM in accordance with Equation 20.

3.2 HMEM parameter re-estimation

In the training phase, we are given a set of K i.i.d. training data $\left\{O^k\right\}_{k=1}^{K}$; the goal is to find the best set of model parameters $\hat{\lambda}$, which maximizes the log likelihood:

$$\begin{aligned}\hat{\lambda} &\overset{\text{def}}{=} \text{argmax}_{\lambda}\, L(\lambda),\\ L(\lambda) &\overset{\text{def}}{=} \frac{1}{K}\sum\nolimits_{k=1}^{K} \ln P\left(O^{(k)}\middle|\lambda\right).\end{aligned} \tag{21}$$

Substituting Equation 5 for the likelihood $P(O^{(k)}|\lambda)$ leads to an excessively complex optimization problem with seemingly impossible direct solution. The major issue is that the distribution wholly depends upon the latent variables which are unknown. The expectation maximization (EM) technique offers an iterative algorithm which overcomes this problem and accurately solves the issue:

$$\begin{aligned}\lambda^{n+1} &= \text{argmax}_{\lambda}\, \mathcal{Q}(\lambda;\lambda^n),\\ \mathcal{Q}(\lambda;\lambda^n) &= \sum\nolimits_k \sum\nolimits_{\text{all } d,\text{all } q} P\left(d,q\middle|O^{(k)};\lambda^n\right) \ln P\left(O^{(k)},d,q\middle|;\lambda\right),\end{aligned} \tag{22}$$

where d and q represent possible state durations and space indexes for the kth training utterance and the second summation is calculated over all possible values of d and q. In general, these functions cannot be minimized in a closed-form expression. Therefore, a numerical optimization technique such as the Broyden-Fletcher-Goldfarb-Shanno (BFGS) [48] method or Newton algorithm has to be derived to find one of the local optima. This paper proposes to exploit the outstanding BFGS algorithm, due to its favorable characteristics. However, BFGS needs solely the first partial derivatives of the cost functions calculated as follows:

$$\begin{aligned}
\frac{\partial\mathcal{Q}}{\partial u_i^l} &= -\frac{1}{2}\sum\nolimits_k f_l^d(i)\sum\nolimits_{t=1}^{T}\sum\nolimits_{d=1}^{t}\chi_t^d(i)\left[d+\frac{u_i}{2h_i}\right],\\
\frac{\partial\mathcal{Q}}{\partial h_i^l} &= -\frac{1}{2}\sum\nolimits_k f_l^d(i)\sum\nolimits_{t=1}^{T}\sum\nolimits_{d=1}^{t}\chi_t^d(i)\left[d^2-\frac{1}{h_i}-\left(\frac{u_i}{2h_i}\right)^2\right],\\
\frac{\partial\mathcal{Q}}{\partial w_{ig}^l} &= \sum\nolimits_k f_l^o(i)\sum\nolimits_{t=1}^{T}\gamma_t(i,g)\left[1-w_{ig}\right],\\
\frac{\partial\mathcal{Q}}{\partial u_{ig}^l} &= -\frac{1}{2}\sum\nolimits_k f_l^o(i)\sum\nolimits_{t=1}^{T}\gamma_t(i,g)\left[o_t+\frac{{H_{ig}}^{-1}u_{ig}}{2}\right],\\
\frac{\partial\mathcal{Q}}{\partial H_{ig}^l} &= -\frac{1}{2}\sum\nolimits_k f_l^o(i)\sum\nolimits_{t=1}^{T}\gamma_t(i,g)\left[o_t o_t^T-{H_{ig}}^{-1}-\frac{{H_{ig}}^{-1}u_{ig}{u_{ig}}^T{H_{ig}}^{-1}}{4}\right],
\end{aligned} \tag{23}$$

where $\gamma_t(i,g)$ and $\chi_t^d(i)$ are defined in Section 2.3. Therefore, at every iteration of BFGS, we need to find the above gradient values and BFGS estimates new parameters which are closer to the optimum ones.

At first glance, calculating the above gradient expressions seems to be computationally expensive, but they can be calculated efficiently if we rewrite them in terms of sufficient statistics as in the following equations. By doing this, the computational complexity no longer depends on the number of training observation vectors, but rather on the total number of states. Furthermore, storing sufficient statistics instead of all observation vectors reduces the amount of main memory usage of the training procedure. These equations are expressed as

$$\begin{aligned}
\frac{\partial\mathcal{Q}}{\partial u_i^l} &= -\frac{1}{2}\sum\nolimits_k f_l^d(i)\,\tilde{X}_i\left(\tilde{m}_i+\frac{u_i}{2h_i}\right),\\
\frac{\partial\mathcal{Q}}{\partial h_i^l} &= -\frac{1}{2}\sum\nolimits_k f_l^d(i)\,\tilde{X}_i\left(\tilde{r}_i-\frac{1}{h_i}-\left(\frac{u_i}{2h_i}\right)^2\right),\\
\frac{\partial\mathcal{Q}}{\partial w_{ig}^l} &= \sum\nolimits_k f_l^o(i)\tilde{\gamma}(i,g)\left[1-w_{ig}\right],\\
\frac{\partial\mathcal{Q}}{\partial u_{ig}^l} &= -\frac{1}{2}\sum\nolimits_k f_l^o(i)\tilde{\gamma}(i,g)\left[\tilde{\mu}(i,g)+\frac{{H_{ig}}^{-1}u_{ig}}{2}\right],\\
\frac{\partial\mathcal{Q}}{\partial u_{ig}^l} &= -\frac{1}{2}\sum\nolimits_k f_l^o(i)\tilde{\gamma}(i,g)\left[\tilde{R}(i,g)+\frac{{H_{ig}}^{-1}u_{ig}}{2}\right],
\end{aligned} \tag{24}$$

where $\tilde{X}_i$, $\tilde{m}_i$, and $\tilde{r}_i$ are sufficient statistics required to train duration distribution and are calculated as

$$\begin{aligned}
\tilde{X}_i &= \sum\nolimits_{t=1}^{T}\sum\nolimits_{d=1}^{t}\chi_t^d(i),\\
\tilde{m}_i &= \frac{1}{\tilde{X}_i}\sum\nolimits_{t=1}^{T}\sum\nolimits_{d=1}^{t}\chi_t^d(i)d,\\
\tilde{r}_i &= \frac{1}{\tilde{X}_i}\sum\nolimits_{t=1}^{T}\sum\nolimits_{d=1}^{t}\chi_t^d(i)d^2.
\end{aligned} \tag{25}$$

Also, $\tilde{\gamma}(i,g)$, $\tilde{\mu}(i,g)$, and $\tilde{R}(i,g)$ are sufficient statistics related to output probability distribution:

$$\begin{aligned}\tilde{\gamma}(i,g) &= \sum\nolimits_{t=1}^{T} \gamma_t(i,g),\\ \tilde{\mu}0(i,g) &= \frac{1}{\tilde{\gamma}(i,g)}\sum\nolimits_{t=1}^{T}\sum\nolimits_{d=1}^{t}\chi_t^d(i)o_t,\\ \tilde{R}(i,g) &= \frac{1}{\tilde{\gamma}(i,g)}\sum\nolimits_{t=1}^{T}\sum\nolimits_{d=1}^{t}\chi_t^d(i)o_t^2.\end{aligned} \tag{26}$$

These equations prove that regardless of calculating sufficient statistics, an EM iteration in HMEM is just equivalent to train three maximum entropy models for state duration distribution, state output distribution for each subspace, and subspace probability.

Having introduced HMEM parameter estimation procedure, we can now proceed to explain the overall structure of HMEM. Figure 4 shows the whole architecture illustrating the HMEM-based speech synthesis system. Just like other statistical parametric approaches, it consists of two phases, training and synthesis. In the training phase, we first extract a parametric representation of the speech signal (i.e., acoustic features) including both spectral and excitation features from training speech database. In parallel, contextual factors are obtained for all states of the database.

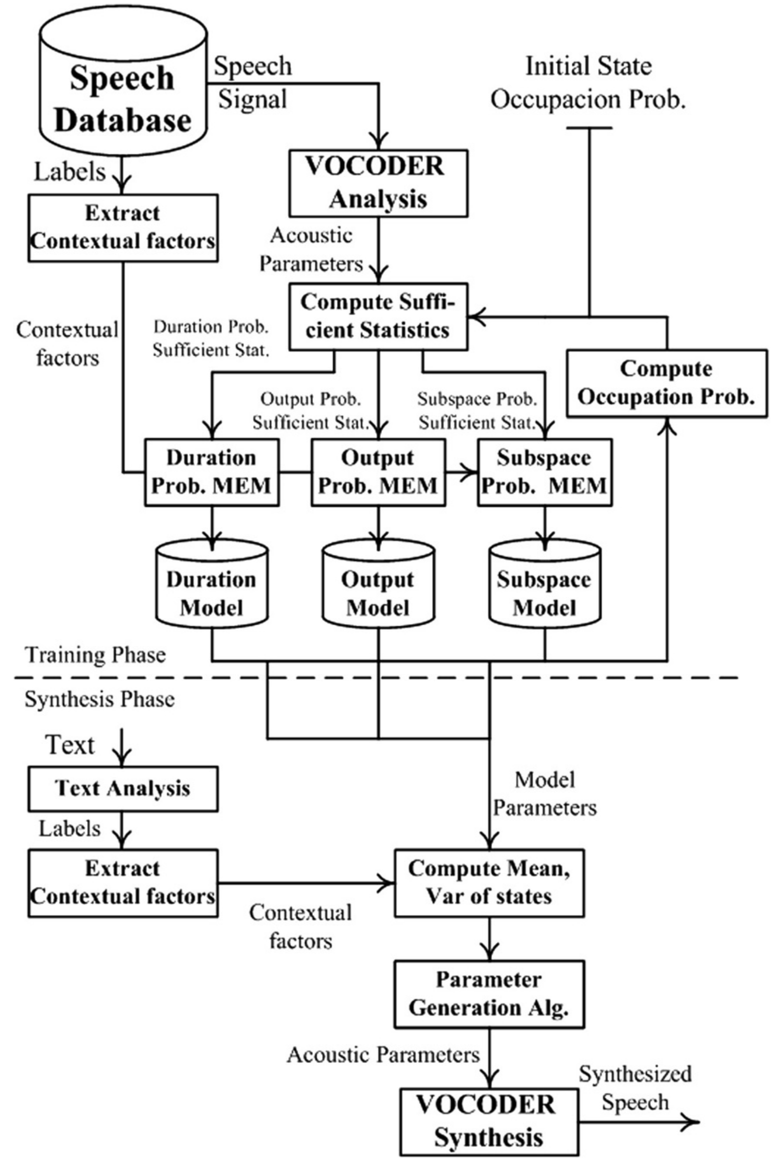


Figure 4 Block-diagram of HMEM-based speech synthesis.

Thereafter, both acoustic and contextual factors are applied for HMEM training. The training procedure is performed by iterating through three steps: computing sufficient statistics, training all maximum entropy distributions, and calculating occupation probabilities. However, the training procedure needs prior information about state occupation probabilities for the first iteration. This paper proposes to utilize a trained HMM for this purpose. Training procedure continues until an amount of increase in likelihood falls below a specific threshold. The synthesis phase is completely identical to a typical HSMM-based speech synthesis system. The only difference is that in HMEM state, mean and covariance parameters are estimated in accordance with Equation 20 instead of tracing a binary decision tree.

3.3 Decision tree-based context clustering

Statistical parametric speech synthesis systems typically exploit around 50 different types of contextual factors [23]. For such system, it is impossible to prepare tanning data covering all context-dependent models, and there are a large number of unseen models that have to be predicted in synthesis phase. Therefore, a context clustering approach such as decision tree-based clustering has to be used to decide about unseen contexts [31,45]. Due to the critical importance of context clustering algorithms in HMM-based speech synthesis systems, this section focuses on designing a clustering algorithm for HMEM.

As it is realized from the discussion in this section, In order to implement the proposed architecture, we initially need to define two sets of contextual regions. These regions are represented by two sets, namely $\{f_l(c)\}_{l=1}^{L_f}$ and $\{g_l(c)\}_{l=1}^{L_g}$. First- and second-order moment constraints have to be satisfied for all regions in $\{f_l(c)\}_{l=1}^{L_f}$ and $\{g_l(c)\}_{l=1}^{L_g}$, respectively. Before training, the first empirical moments of all regions in $\{f_l(c)\}_{l=1}^{L_f}$ and the second empirical moments of all regions in $\{g_l(c)\}_{l=1}^{L_g}$ are computed using training data. Then, HMEM is trained to be consistent with these empirical moments. The major difficulty in defining these regions is to find a satisfactory balance between model complexity and the availability of training data. For limited training databases, a model with a small number of parameters, i.e., small number of regions has to be defined. In this case, bigger (strongly overlapped) contextual regions seem to be more desirable, because they can alleviate the problem of weak context generalization. On the other hand, for large training databases, larger number of contextual regions has to be defined to escape from under-fitting model to training data. In this case, smaller contextual regions can be applied to capture the details of acoustic features. This section introduces an algorithm that defines multiple contextual regions for first- and second-order moments by considering HMEM structure.

Due to the complex relationship between acoustic features and contextual factors, it is extremely difficult to find the optimum sets of contextual regions that maximize likelihood for HMEM. For the sake of simplicity, we have made some simplifying assumptions to find a number of suboptimum contextual regions. These assumptions are expressed as follows:

- We have used conventional binary decision tree structures to define $\{f_l(c)\}_{l=1}^{L_f}$ and $\{g_l(c)\}_{l=1}^{L_g}$. This is a common approach in many former papers [19,20,23]. It should be noted that the decision tree structure is not the only possible structure to express the relationship between acoustic features and contextual factors. For example, other approaches such as neural networks or soft-clustering methods can be applied as well. However, in this paper, we limit our discussion to the conventional binary decision tree structure.
- Multiple decision trees are trained for $\{f_l(c)\}_{l=1}^{L_f}$, and just one decision tree is constructed for $\{g_l(c)\}_{l=1}^{L_g}$. In this way, the final HMEM preserves the first empirical moments of multiple decision trees, and the second moments of just one decision tree. This assumption is a result of the fact that first-order moments seem to be more important than second-order moments [32,47].
- The discussion of current section shows that the ML estimates of parameters defined for $\{f_l(c)\}_{l=1}^{L_f}$ and $\{g_l(c)\}_{l=1}^{L_g}$ significantly depend on each other. Therefore, in each step of decision tree construction, a BFGS optimization algorithm has to be executed to re-estimate both sets of parameters simultaneously, and this procedure leads to an extreme amount of computational complexity. To alleviate this problem, it is proposed to borrow $\{g_l(c)\}_{l=1}^{L_g}$ from a baseline system (conventional HMM-based speech synthesis system) and construct $\{f_l(c)\}_{l=1}^{L_f}$ independently.
- In HMEM structure, $\{f_l(c)\}_{l=1}^{L_f}$ is responsible to provide satisfactory clustering of first-order moments (mean vectors). Similarly, contextual additive structures [19,20,37] that tie all covariance matrices offer multiple overlapped clustering of mean vectors based on the likelihood criterion; therefore, an appropriate method is to borrow $\{f_l(c)\}_{l=1}^{L_f}$ from the contextual additive structure.
- However, training a contextual additive structure using algorithms proposed in [19,20] is still computationally expensive for large training databases (more than 500 sentences). Three modifications are applied to the algorithm proposed by Takaki et al. [19] for computational complexity

reduction: (i) The number of decision trees is considered to be fixed (in our experiments, an additive structure with four decision trees is built). (ii) Questions are selected one by one for different decision trees. Therefore, all trees are grown simultaneously, and the size of all trees would be equal. (iii) In the process of selecting the best pair of question and leaf, it is assumed that just the parameters of candidate leaf will be changed and all other parameters remain unchanged. It should be noted that the selection procedure is repeated until the total number of free parameters reaches the number of parameters trained for the baseline system (HSMM-based speech synthesis system).

In sum, the final algorithm of determining $\{f_l(c)\}_{l=1}^{L_f}$ and $\{g_l(c)\}_{l=1}^{L_g}$ can be summarized as follows. $\{g_l(c)\}_{l=1}^{L_g}$ is simply borrowed from a conventional HMM-based speech synthesis system. $\{f_l(c)\}_{l=1}^{L_f}$ also resulted from an independent context clustering algorithm that is a fast and simplified version of contextual additive structure [19]. This clustering algorithm builds four binary context-dependent decision trees, simultaneously. It should be noted that when the number of clusters reaches the number of leaves of the decision tree trained for an HSMM-based system, the clustering algorithm is finished.

The following algorithm shows the overall procedure of the proposed context clustering.

4 Experiments

We have conducted two sets of experiments. First, the performance of HMEM with heuristic context clusters is examined; second, the impact of the proposed method for decision tree-based context clustering presented in the Section 3.3 is evaluated.

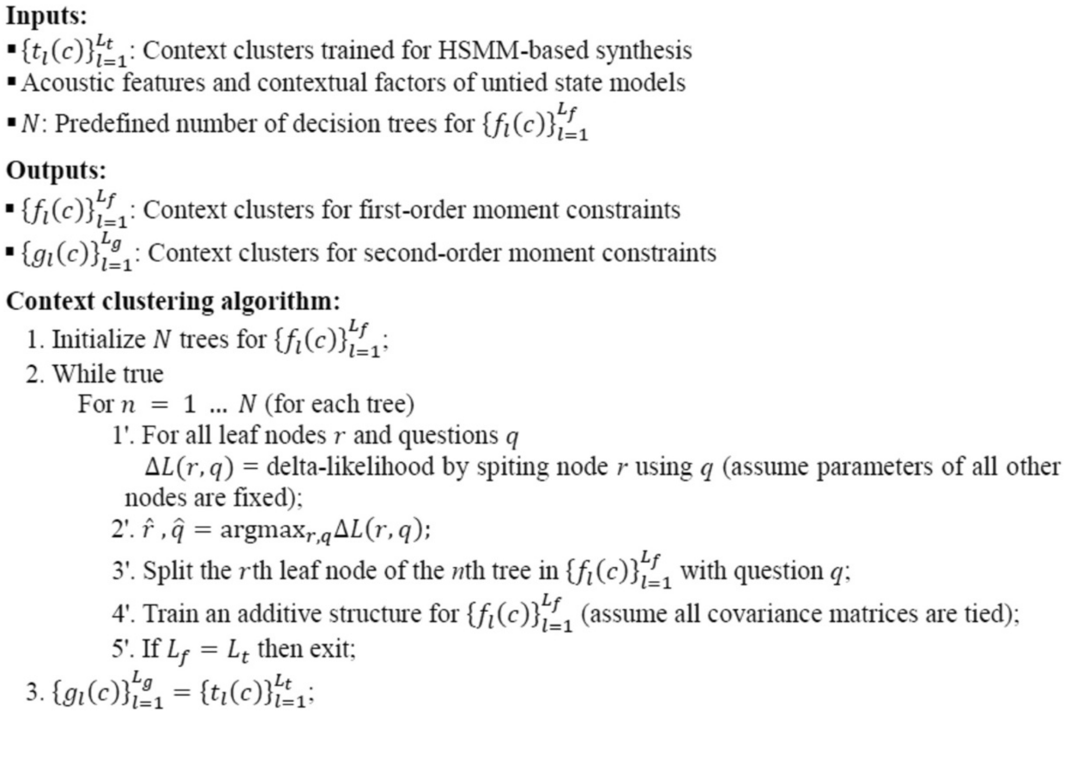
4.1 Performance evaluation of HMEM with heuristic context clusters

This subsection aims to compare HMEM-based acoustic modeling with conventional HSMM-based method. In this subsection, contextual regions of HMEM are defined heuristically and it is fixed for different sizes of training database.

4.1.1 Experimental conditions

A Persian speech database [49] consisting of 1,000 utterances from a male speaker was used throughout our experiments. Sentences were between 5 and 20 words long and have an average duration of 8 s. This database was specifically designed for the purpose of speech synthesis. Sentences in the database covered most frequent Persian words, all bi-letter combinations, all bi-phoneme combinations, and most frequent Persian syllables. In the modeling of the synthesis units, 31 phonemes were used, including silence. As presented in Section 4.1.2, a large variety of phonetic and linguistic contextual factors was considered in this work.

Speech signals were sampled at a rate of 16 kHz and windowed by a 25-ms Blackman window with a 5-ms



```
Inputs:
▪ $\{t_l(c)\}_{l=1}^{Lt}$: Context clusters trained for HSMM-based synthesis
▪ Acoustic features and contextual factors of untied state models
▪ $N$: Predefined number of decision trees for $\{f_l(c)\}_{l=1}^{Lf}$
Outputs:
▪ $\{f_l(c)\}_{l=1}^{Lf}$: Context clusters for first-order moment constraints
▪ $\{g_l(c)\}_{l=1}^{Lg}$: Context clusters for second-order moment constraints
Context clustering algorithm:
  1. Initialize $N$ trees for $\{f_l(c)\}_{l=1}^{Lf}$;
  2. While true
       For $n = 1 \ldots N$ (for each tree)
           1'. For all leaf nodes $r$ and questions $q$
                 $\Delta L(r,q)$ = delta-likelihood by spiting node $r$ using $q$ (assume parameters of all other
             nodes are fixed);
           2'. $\hat{r}, \hat{q} = \mathrm{argmax}_{r,q} \Delta L(r,q)$;
           3'. Split the $r$th leaf node of the $n$th tree in $\{f_l(c)\}_{l=1}^{Lf}$ with question $q$;
           4'. Train an additive structure for $\{f_l(c)\}_{l=1}^{Lf}$ (assume all covariance matrices are tied);
           5'. If $L_f = L_t$ then exit;
  3. $\{g_l(c)\}_{l=1}^{Lg} = \{t_l(c)\}_{l=1}^{Lt}$;
```

shift. 40 Mel-cepstral coefficients, 5 bandpass aperiodicity and fundamental frequency, and their delta and delta-delta coefficients extracted by STRAIGHT [11] were employed as our acoustic features. In this experiment, the number of states was 5, and multi-stream left-to-right with no skip path MSD-HSMM was trained as the traditional HSMM system. Decision trees were built using maximum likelihood criterion, and the size of decision trees was determined by MDL principle [46]. Additionally, global variance (GV)-based parameter generation algorithm [20,26] and STRAIGHT vocoder were applied in the synthesis phase.

Both subjective and objective tests were carried out to compare HMEM that uses some heuristic contextual regions with the traditional HSMM system. In our experiments, two different synthesis systems named HMEM1 and HMEM2 were developed based on the proposed approach. HMEM1 employs a small number of general highly overlapped contextual factors that are designed carefully for each stream, while HMEM2 uses a larger number of contextual factors.

More precisely, a set of 64 initial contextual factors were extracted for each segment (phoneme) of the Persian database. These factors contain both segmental and suprasegmental contextual features. From these contextual factors, a set of approximately 8,000 contextual questions were designed and the HSMM system was trained using these questions. Each question can form two regions; therefore, these 8,000 questions can be converted to 16,000 regions. For each stream of HMEM1, a small number of these contextual regions that seem to be more important for that stream were selected and HMEM1 was trained using them. Contextual factors of HMEM2 contain all contextual factors of HMEM1 in addition to a number of detailed ones. The number of contextual regions in HMEM2 is twice the number of regions in HMEM1. Regions of both HMEM1 and HMEM2 were selected based on the linguistic knowledge of the Persian language. Table 1 shows the number of contextual regions for different synthesis systems (namely HSMM with different training data sizes, HMEM1 and HMEM2).

Experiments were conducted on five different training sets with 50, 100, 200, 400, and 800 utterances. Additionally, a fixed set of 200 utterances, not included in the training sets, was used for testing.

4.1.2 Employed contextual factors

In our experiments, contextual factors contained phonetic, syllable, word, phrase, and sentence level features. In each of these levels, both general and detailed features were considered. Features such as phoneme identity, syllable stress pattern, or word part-of-speech tag are examples of general features, and a question like the position of the current phoneme is a sample of a detailed one. Specific information with regard to contextual features is presented in this subsection.

Contextual factors play a significant role in the proposed HMEM method. As a consequence, they have been designed carefully and are now briefly presented:

➢ Phonetic-level features
- Phoneme identity before the preceding phoneme; preceding, current, and succeeding phonemes; and phoneme identity after the next phoneme
- Position of the current phoneme in the current syllable (forward and backward)
- Whether this phoneme is ‘Ezafe’ [50] or not (Ezafe is a special feature in Persian pronounced as a short vowel ‘e’ and relates two different words together. Ezafe is not written but is pronounced and has a profound effect on intonation)

➢ Syllable-level features
- Stress level of this syllable (five different stress levels are defined for our speech database)
- Position of the current syllable in the current word and phrase (forward and backward)
- Type of the current syllable (syllables in Persian language are structured as CV, CVC, or CVCC, where C and V denote consonants and vowels, respectively)
- Number of the stressed syllables before and after the current syllable in the current phrase
- Number of syllables from the previous stressed syllable to the current syllable
- Vowel identity of the current syllable

➢ Word-level features
- Part-of-speech (POS) tag of the preceding, current and succeeding word
- Position of the current word in the current sentence (forward and backward)

Table 1 The number of leaf nodes for each stream in different speech synthesis systems

		Various speech synthesis systems					
		HSMM-100	**HSMM-200**	**HSMM-400**	**HSMM-800**	**HMEM1**	**HMEM2**
Streams of acoustic features	bap	239	392	581	958	565	1,130
	dur	124	193	319	512	256	512
	log F0	590	904	1,425	2,487	565	1,130
	mgc	267	416	736	1,279	695	1,390
Total parameters		75,628	118,314	204,683	354,133	188,217	377,834

- Whether the current word contains ‘Ezafe’ or not
- Whether this word is the last word in the sentence or not

➢ Phrase-level features

- Number of syllables in the preceding, current, and succeeding phrase
- Position of the current phrase in the sentence (forward and backward)

➢ Sentence-level features

- Number of syllables, words, and phrases in the current sentence
- Type of the current sentence

4.1.3 Illustratory example

Before going further with the objective and subjective evaluations, the superiority of HMEM over HSMM when few training data are available can be already illustrated. Although the improvement will be shown in Sections 4.1.4 and 4.1.5 to be achieved for all speech characteristics (log F0, duration, and spectral features), it is here emphasized for the prediction of log F0 trajectories. Figure 5 shows the trajectory of log F0 generated by HSMM and HMEM1 with 100 training utterances, in contrast to the natural contour. This plot confirms the superiority of HMEM over HSMM in modeling fundamental frequency when the amount of training data is small, as the generated contour by HMEM is far closer to the natural one compared to HSMM.

In limited training sets, HSMM produces sudden transitions between adjacent states. This drawback is the result of decision tree-clustered context-dependent modeling. More specifically, when few data are available for training, the number of leaves in the decision tree is reduced. As a result, the distance between the mean vectors of adjacent states can be large. Even the parameter generation algorithm proposed by [26-28] cannot compensate such jumps. In such cases, the quality of synthetic speech with HSMM is expected to deteriorate.

On the opposite, if we let adjacent states contain common active contextual factors, then the variation of mean vectors in state transitions will be smoother. This is the key idea of HMEM which makes it possible to outperform HSMM when the data are limited. However, the use of overlapped contextual factors in HMEM will result in over-smoothing problem when the size of the training data is increased. Therefore, the detailed contextual factors are additionally considered in HMEM2 to alleviate the over-smoothing issue.

4.1.4 Objective evaluation

The average mel-cepstral distortion (MCD) [51] and root-mean-square (RMS) error of phoneme durations (expressed in terms of number of frames) were selected as relevant metrics for our objective assessment. For the calculation of both average mel-cepstral distance and RMS error of phoneme durations, the state boundaries (state durations) were determined using Viterbi alignment with the speaker's real utterance.

The MCD measure is defined by:

$$\mathrm{MCD} = \frac{10}{\ln(10)} * \sqrt{2\sum_{\mathrm{i}=1}^{40}\left(mc_i^t - mc_i^p\right)^2}, \tag{27}$$

where mc_i is the ith mel-cepstral coefficients in a frame, mc^t is the target coefficient we are comparing against, and mc^p is the generated coefficient. In addition, RMS is defined as the following function:

$$\mathrm{RMS} = \sqrt{\sum_{s=1}^{N}\left(d_s^t - d_s^p\right)^2/N}, \tag{28}$$

where N is the total number of states in a sentence, d_s is the duration of the sth state, d_s^t is the original duration, and d_s^p is the estimated duration.

Figure 6 shows the average mel-cepstral distance between spectra generated from the proposed method and

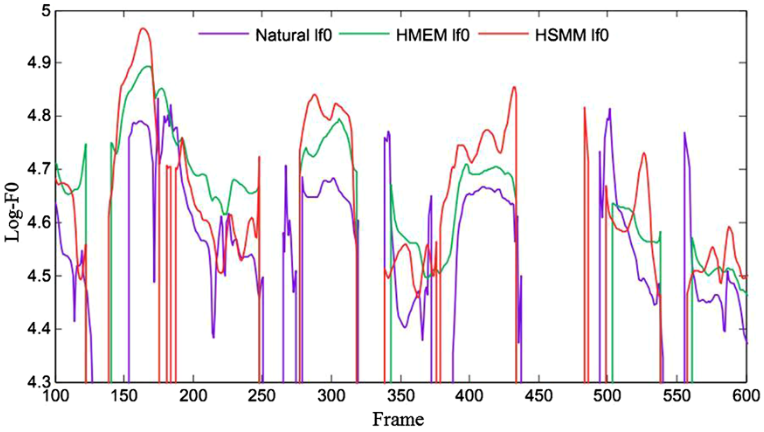
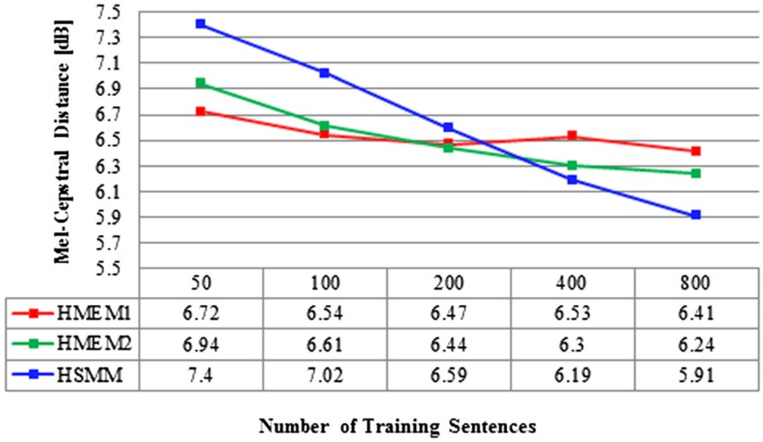


Figure 5 **Trajectory of log F0 generated from the HSMM, HMEM as well as the natural log F0.**



	50	100	200	400	800
HMEM1	6.72	6.54	6.47	6.53	6.41
HMEM2	6.94	6.61	6.44	6.3	6.24
HSMM	7.4	7.02	6.59	6.19	5.91

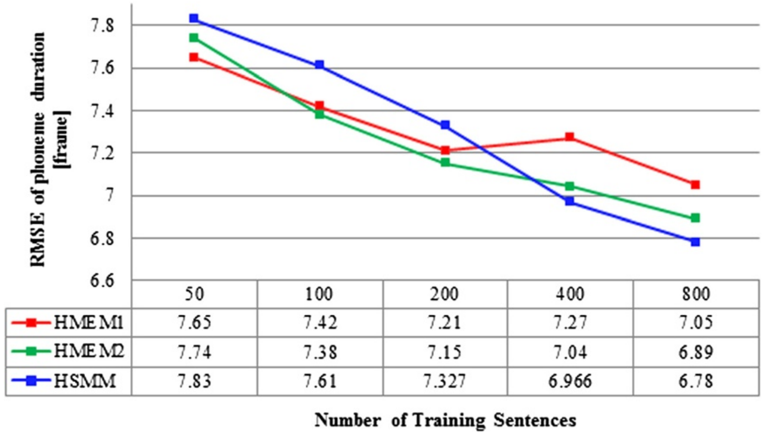
Figure 6 **Comparison of average MCD as an objective measure between the proposed method and the HSMM-based one.**

spectra obtained by analyzing the speaker's real utterance. For comparison, we also present the average distance of spectra generated from the HSMM-based method and the real spectra. In this figure, it is clearly observed that the proposed HMEM systems outperform the standard HSMM approach for limited training datasets. Nonetheless, this advantage disappears when more than 200 utterances are available for training. It can be noticed that a reduction of the size of the training set has a dramatic impact on the performance of HSMM, contrary to HMEM-based systems.

The same conclusions are observed for Figure 7 in which the generated duration of proposed systems is compared against that of HSMM. It can be again noticed that the proposed systems outperform HSMM in small databases. However, when the size of the database increases, HSMM gradually surpasses the proposed HMEM systems. Furthermore, detailed features added in HMEM2 affect the proposed method constructively when the synthesis units model by large databases. Thus, we expect that the proposed method could be comparable with HSMM or outperform it even for large databases if we apply more detailed and well-designed features.

In summary, from these figures and the illustratory example presented before, we can see that when the available data are limited, all features (log F0, duration, and spectra) of synthetic speech generated by HMEM are closer to the original features than those obtained with HSMM. However, when the training database is large, the HSMM-based method performs better than HMEM. Nevertheless, employing more detailed features can assist the proposed method in becoming closer to the HSMM-based synthetic speech.

In addition to the abovementioned objective measurements, we have compared the accuracy of voiced/unvoiced detection in the proposed system with its counterpart in HSMM-based synthesis. Table 2 shows information about the false negative (FN), false positive (FP), true negative (TN), and true positive (TP) rates. Moreover, the data in Table 2 are summarized in Table 3 in which the accuracy of detecting voice/unvoiced regions is presented. As



	50	100	200	400	800
HMEM1	7.65	7.42	7.21	7.27	7.05
HMEM2	7.74	7.38	7.15	7.04	6.89
HSMM	7.83	7.61	7.327	6.966	6.78

Figure 7 **Comparison of RMS error of phoneme durations as objective measure between the proposed method and HSMM one.**

Table 2 FN, FP, TN, and TP rates of detecting voiced/unvoiced regions through HMEM2 and the HSMM-based method

# training data	Implemented systems		Really voiced (%)	Really unvoiced (%)
50	HMEM2	Voiced	77.00	3.70
		Unvoiced	7.09	12.21
	HSMM	Voiced	78.23	5.79
		Unvoiced	5.86	10.12
100	HMEM2	Voiced	75.78	2.75
		Unvoiced	8.31	13.16
	HSMM	Voiced	78.07	5.34
		Unvoiced	6.02	10.57
200	HMEM2	Voiced	77.25	1.54
		Unvoiced	6.84	14.37
	HSMM	Voiced	78.43	4.43
		Unvoiced	5.66	11.48
400	HMEM2	Voiced	77.18	1.34
		Unvoiced	6.91	14.57
	HSMM	Voiced	76.09	2.70
		Unvoiced	8.00	13.21
800	HMEM2	Voiced	77.10	0.83
		Unvoiced	6.99	15.08
	HSMM	Voiced	77.17	2.66
		Unvoiced	6.92	13.25

realized from these tables, the proposed method detects voiced/unvoiced regions more accurately than HSMM regardless of the size of the database. In other words, not only in small databases but also for larger ones, HMEM outperforms HSMM in terms of detecting voiced/unvoiced regions.

4.1.5 Subjective evaluation

Two different subjective methods are employed in order to show the effectiveness of the proposed system and assess the effect of the size of the training database. A comparative mean opinion score (CMOS) test [52] with a 7-point scale, ranging from −3 (meaning that method A is much better than method B) to 3 (meaning the opposite), and a preference scoring [53] are used to evaluate the subjective quality of the synthesized speech. The results of this evaluation are respectively shown in Figures 8 and 9.

Table 3 Accuracy of voiced/unvoiced detector

# training data	HMEM2 accuracy (%)	HSMM accuracy (%)
50	89.21	88.35
100	88.94	88.64
200	91.62	89.91
400	91.75	89.30
800	92.18	90.42

Twenty native participants were asked to listen to ten randomly chosen pairs of synthesized speech samples generated by two different systems (selected arbitrarily among HMEM1, HMEM2, and HSMM).

Remarkably, the proposed systems are noticed to be of a great interest when the training data are limited (i.e., for 50, 100, and 200 utterances) and are in line with the conclusions of the objective assessments. The superiority of HMEM1 over HSMM and HMEM2 is clear in the training sets containing 50 and 100 utterances. In other words, general contextual factors lead the proposed system to a better performance when the amount of training

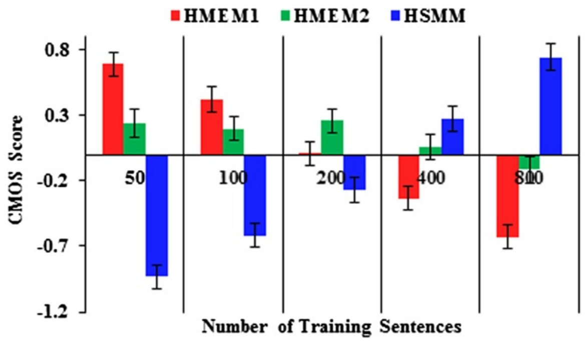


Figure 8 Averaged CMOS scores for the HMEM1, HMEM2, and HSMM. 95% confidence intervals are also indicated.

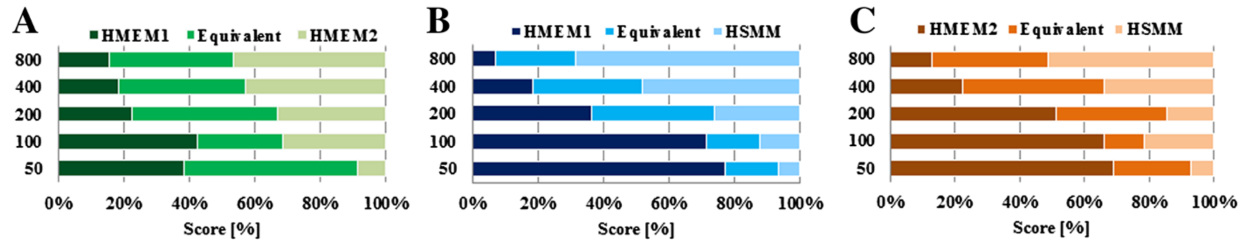


Figure 9 Preference scores as a function of the number of utterances used for training. (A) Comparison between HMEM1 and HMEM2. **(B)** Comparison between HMEM1 and HSMM. **(C)** Comparison between HMEM2 and HSMM.

data is very small. Gradually, as the number of utterances in the training set increases, detailed features assist the proposed system in achieving more effective synthetic speech. Therefore, HMEM2 surpasses HMEM1 for training sets with 200 and more utterances. However, for relatively large training sets (400 and 800), the use of HSMM is recommended.

Table 1 compares the number of leaf nodes in different speech synthesis systems. It can be seen from the table that to model mgc stream, HMEM2 exploits more parameters than HSMM-400 and HSMM-800, but the objective evaluations presented in Figure 6 show that HSMM-400 and HSMM-800 results in better mel-cepstral distances. The above argument shows that HMEM with some heuristic contextual clusters cannot exploit model parameters efficiently. In fact, a great number of contextual regions in HMEM1 and HMEM2 are redundant; therefore, their corresponding parameters are not useful. The next section evaluates the performance of HMEM with the suboptimum context clustering algorithm proposed in Section 3.3. This proposed clustering algorithm selects appropriate contextual regions and consequently solves the aforementioned problem.

4.2 Performance evaluation of HMEM with decision tree-based context clustering

This section is dedicated to the second set of experiments conducted to evaluate the performance of HMEM with decision tree construction algorithm proposed in Section 3.3. As it is realized from the first set of experiments, HMEM with heuristic and naïve contextual regions cannot outperform HSMM in large training databases. This section proves that by employing appropriate sets of $\{f_l(c)\}_{l=1}^{L_f}$ and $\{g_l(c)\}_{l=1}^{L_g}$, HMEM outperforms HSMM even for large databases.

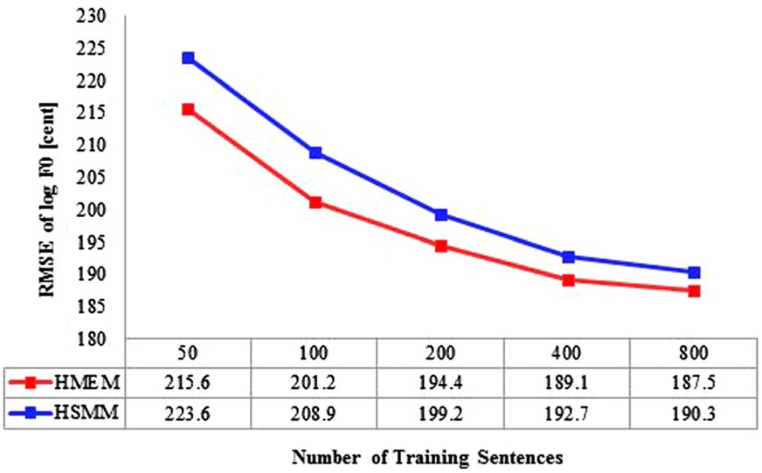
4.2.1 Experimental conditions

Experiments were carried out on Nick [54], a British male database collected in Edinburgh University. This database consists of 2,500 utterances from a male speaker. We considered five sets including 50, 100, 200, 400, and 800 utterances for training, and 200 sentences that were not included in training sets were used as test data. Each sentence in the database is about 5 s of speech. Speech signals are sampled at 48 kHz, windowed by a 25-ms Blackman window with 5-ms shift. This database was specifically designed for the purpose of speech synthesis research, and utterances in the database covered most frequent English words. Also, different segmental and suprasegmental contextual factors were extracted for this database.

The speech analysis conditions and model topologies of CSTR/EMIME HTS 2010 [54] were used in this experiment. Bark cepstrum was extracted from smooth STRAIGHT trajectories [11]. Also, instead of log F0 and five frequency sub-bands (0 to 1, 1 to 2, 2 to 4, 4 to 6, and 6 to 8 kHz), pitch in mel and auditory-scale motivated frequency bands for aperiodicity measure were applied [54]. The analysis process resulted in 40 bark cepstrum coefficients, 1 mel in pitch value, and 25 auditory-scale motivated frequency bands aperiodicity parameters for each frame of training speech signals. These parameters incorporated with their delta and delta-delta parameters considered as the observation vectors of the statistical parametric model.

A five-state multi-stream left-to-right with no skip path MSD-HSMM was trained as the baseline system. Conventional maximum likelihood-based decision tree clustering algorithm was used to tie HMM states, but MDL criterion is used to determine the size of decision trees.

In order to have a fair comparison, the proposed system (HMEM with decision tree structure) was trained with the same number of free model parameters as the baseline system. HMEM was trained based on the decision tree construction algorithm presented in Section 3.3 and parameter re-estimation algorithm proposed in Section 3.2. It should be noted that four decision trees were built for $\{f_l(c)\}_{l=1}^{L_f}$ and one decision tree for $\{g_l(c)\}_{l=1}^{L_g}$. After training acoustic models, in the synthesis phase, GV-based parameter generation algorithm [20,26] and STRAIGHT synthesis module generated synthesized speech signals. Both subjective and objective tests were conducted to compare HMEM that uses decision tree-based clusters with traditional HSMM-based synthesis.



	50	100	200	400	800
HMEM	215.6	201.2	194.4	189.1	187.5
HSMM	223.6	208.9	199.2	192.7	190.3

Figure 10 RMSE as objective measure to compare log F0 trajectories generated by decision tree-based HMEM and conventional HSMM.

It is useful to mention that training the proposed HMEM structure with decision tree-based context clustering took approximately 5 days for 800 training sentences, while training its corresponding HSMM-based synthesis system took approximately 16 h.

4.2.2 Employed contextual factors

In this experiment, employed contextual factors contained phonetic, syllable, word, phrase, and sentence level factors. In each of these levels, all important features were considered. Specific information about these features is presented in this subsection.

➢ Phonetic-level features

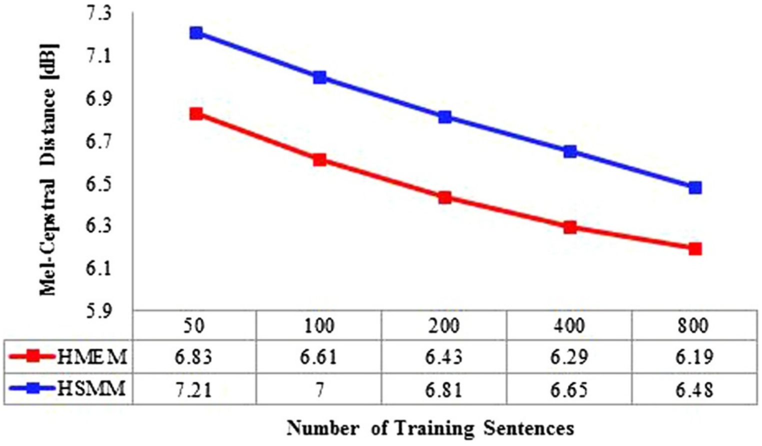
- Phoneme identity before the preceding phoneme; preceding, current, and succeeding phonemes; and phoneme identity after the next phoneme
- Position of the current phoneme in the current syllable, word, phrase, and sentence

➢ Syllable-level features

- Stress level of previous, current, and next syllable (three different stress levels are defined for this database)
- Position of the current syllable in the current word, phrase, and sentence
- Number of the phonemes of the previous, current, and next syllable
- Whether the previous, current, and next syllable is accented or not
- Number of the stressed syllables before and after the current syllable in the current phrase
- Number of syllables from the previous stressed syllable to the current syllable
- Number of syllables from the previous accented syllable to the current syllable

➢ Word-level features

- Part-of-speech (POS) tag of the preceding, current, and succeeding word



	50	100	200	400	800
HMEM	6.83	6.61	6.43	6.29	6.19
HSMM	7.21	7	6.81	6.65	6.48

Figure 11 Result of the MCD measure that compares decision tree-based HMEM and conventional HSMM.

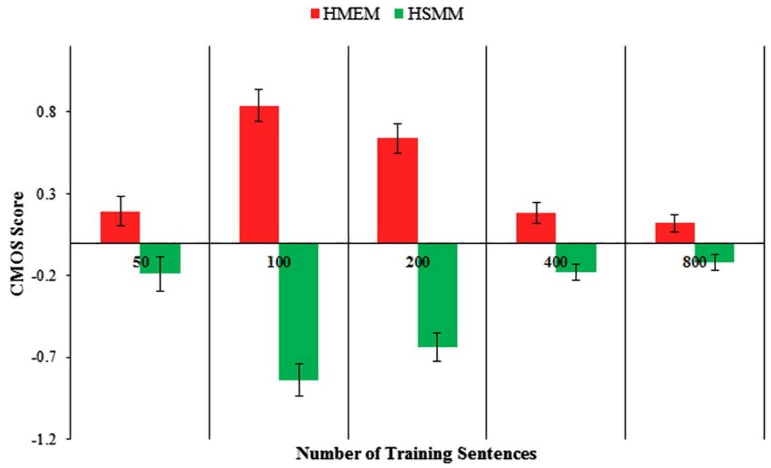


Figure 12 Subjective evaluation of HMEM with decision tree-based context clustering and HSMM through CMOS test with 95% confidence intervals.

- Position of the current word in the current phrase and sentence (forward and backward)
- Number of syllables of the previous, current, and next word
- Number of content words before and after current word in the current phrase
- Number of words from previous and next content word

➢ Phrase-level features

- Number of syllables and words of the preceding, current, and succeeding phrase
- Position of the current phrase in the sentence
- Current phrase ToBI end tone

➢ Sentence-level features

- Number of phonemes, syllables, words, and phrases in the current utterance
- Type of the current sentence

4.2.3 Objective evaluation

Two well-known measures were applied for objective evaluation of the proposed decision tree-based HMEM in comparison with conventional HSMM. The first measure computes RMS error of generated log F0 trajectories, and the second one compares synthesized spectrograms using average MCD criterion. The results of these measures are shown in Figures 10 and 11. As it is realized from Figure 10 that shows the RMS error of the log F0 in terms of cent for different sizes of training data, the log F0 trajectories generated from the proposed approach are more similar to the natural log F0 trajectories, and therefore, HMEM improves the performance of log F0 modeling. However, by increasing the size of the database, the amount of this improvement is slightly reduced. Hence, it can be implied from this figure that in log F0 modeling, the effect of applying overlapped regions for small databases

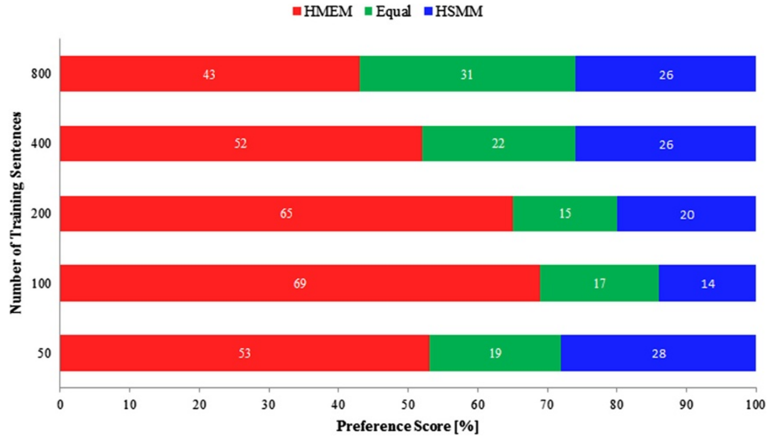


Figure 13 Preference scores as a subjective comparison between HMEM with decision tree-based context clustering and HSMM.

is relatively more than its effect on big databases. Additionally, Figure 11 shows the result of average MCD test. This result also confirms the improvement of HMEM performance in contrast to conventional HSMM for all training databases. As it is clear from the figure, the improvement in average MCD test is fixed for all databases.

4.2.4 Subjective evaluation

We conducted paired comparison tests and reported comparative mean opinion score (CMOS) and preference score as subjective evaluations. Fifteen non-professional native listeners were presented with 30 randomly chosen pairs of synthesized speech generated by HMEM and HSMM. Listeners selected the synthesized speech which sounds better and determined how much is better (much better, better, slightly better, or about the same). The results are shown in Figures 12 and 13.

Both CMOS test and preference score confirm the superiority of the proposed method over HSMM in all databases. Thus, if context clusters are determined through an effective approach, the proposed HMEM will outperform HSMM.

5. Conclusions

This paper addressed the main shortcomings of HSMM in context-dependent acoustic modeling, namely inadequate context generalization. HSMM uses decision tree-based context clustering that does not provide efficient generalization, because each acoustic feature vector is associated with modeling only one context cluster. In order to alleviate this problem, this paper proposed HMEM as a new acoustic modeling technique based on maximum entropy modeling approach. HMEM improves HSMM by enabling its structure to take advantage of overlapped contextual factors, and therefore, it can provide superior context generalization. Experimental results using objective and subjective criteria showed that the proposed system outperforms HSMM.

Despite the advantages, which enabled our system to outperform HSMM, a drawback of computationally complex training procedure is noticed in large databases.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. [2]Centre for Speech Technology Research, University of Edinburgh, Edinburgh EH8 9LW, UK. [3]TCTS Lab, Faculte Polytechnique de Mons, Mons, Belgium.

Received: 12 August 2013 Accepted: 3 March 2014
Published: 7 April 2014

References

1. H Zen, K Tokuda, AW Black, Statistical parametric speech synthesis. Speech Comm. **51**(11), 1039–1064 (2009)
2. AW Black, H Zen, K Tokuda, *Statistical parametric speech synthesis, in IEEE International Conference on Acoustics*, vol. 4 (Speech and Signal Processing (ICASSP), Honolulu, Hawaii, USA, 2007), pp. IV1229–IV1232
3. J Yamagishi, T Kobayashi, Average-voice-based speech synthesis using HSMM-based speaker adaptation and adaptive training. IEICE - Trans. Info. Syst. **90**(2), 533–543 (2007)
4. J Yamagishi, T Nose, H Zen, ZH Ling, T Toda, K Tokuda, S King, S Renals, Robust speaker-adaptive HMM-based text-to-speech synthesis, In IEEE Transactions on Audio, Speech, and Language Processing. **17**(6), 1208–1230 (2009)
5. J Yamagishi, T Kobayashi, Y Nakano, K Ogata, J Isogai, Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm, IEEE Transactions on Audio, Speech, and Language Processing. **17**(1), 66–83 (2009)
6. YJ Wu, Y Nankaku, K Tokuda, State mapping based method for cross-lingual speaker adaptation in HMM-based speech synthesis, in *INTERSPEECH* (Brighton, UK, 2009), pp. 528–531
7. H Liang, J Dines, L Saheer, A comparison of supervised and unsupervised cross-lingual speaker adaptation approaches for HMM-based speech synthesis, in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)* (Dallas, Texas, USA, 2010), pp. 4598–4601
8. M Gibson, T Hirsimaki, R Karhila, M Kurimo, W Byrne, Unsupervised cross-lingual speaker adaptation for HMM-based speech synthesis using two-pass decision tree construction, in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)* (Dallas, Texas, USA, 2010), pp. 4642–4645
9. J Yamagishi, Z Ling, S King, Robustness of HMM-based speech synthesis, in *INTERSPEECH* (Brisbane, Australia, 2008), pp. 581–584
10. T Yoshimura, K Tokuda, T Masuko, T Kobayashi, T Kitamura, Mixed excitation for HMM-based speech synthesis, in *INTERSPEECH* (Aalborg, Denmark, 2001), pp. 2263–2266
11. H Kawahara, I Masuda-Katsuse, A de Cheveigné, Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based F0 extraction: possible role of a repetitive structure in sounds. Speech Comm. **27**(3), 187–207 (1999)
12. T Drugman, G Wilfart, T Dutoit, A deterministic plus stochastic model of the residual signal for improved parametric speech synthesis, in *INTERSPEECH* (Brighton, United Kingdom, 2009), pp. 1779–1782
13. T Drugman, T Dutoit, The deterministic plus stochastic model of the residual signal and its applications. IEEE Trans. Audio. Speech. Lang. Process **20**(3), 968–981 (2012)
14. H Zen, K Tokuda, T Masuko, T Kobayasih, T Kitamura, A hidden semi-Markov model-based speech synthesis system. IEICE - Trans. Info. Syst **90**(5), 825 (2007)
15. K Hashimoto, Y Nankaku, K Tokuda, *A Bayesian approach to hidden semi Markov model based speech synthesis, in Proceedings of INTERSPEECH* (Brighton, United Kingdom, 2009), pp. 1751–1754
16. K Hashimoto, H Zen, Y Nankaku, T Masuko, K Tokuda, *A Bayesian approach to HMM-based speech synthesis, in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Taipei, Taiwan, 2009), pp. 4029–4032
17. K Tokuda, T Masuko, N Miyazaki, T Kobayashi, Multi-space probability distribution HMM. IEICE Trans. on Info. Syst **85**(3), 455–464 (2002)
18. H Zen, A Senior, M Schuster, Statistical parametric speech synthesis using deep neural networks, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Vancouver, British Columbia, Canada, 2013), pp. 7962–7966
19. S Takaki, Y Nankaku, K Tokuda, Spectral modeling with contextual additive structure for HMM-based speech synthesis, in *Proceedings of 7th ISCA Speech Synthesis Workshop* (Kyoto, Japan, 2010), pp. 100–105
20. S Takaki, Y Nankaku, K Tokuda, Contextual partial additive structure for HMM-based speech synthesis, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Vancouver, British Columbia, Canada, 2013), pp. 7878–7882
21. MJ Gales, Cluster adaptive training of hidden Markov models. IEEE Trans. Speech. Audio. Process. **8**(4), 417–428 (2000)
22. H Zen, MJ Gales, Y Nankaku, K Tokuda, Product of experts for statistical parametric speech synthesis, IEEE Trans. Audio. Speech. Lang. Process. **20**(3), 794–805 (2012)
23. K Yu, H Zen, F Mairesse, S Young, Context adaptive training with factorized decision trees for HMM-based statistical parametric speech synthesis. Speech Comm. **53**(6), 914–923 (2011)

24. T Toda, S Young, Trajectory training considering global variance for HMM-based speech synthesis, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Taipei, Taiwan, 2009), pp. 4025–4028
25. L Qin, YJ Wu, ZH Ling, RH Wang, LR Dai, Minimum generation error criterion considering global/local variance for HMM-based speech synthesis, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Las Vegas, Nevada, USA, 2008), pp. 4621–4624
26. T Toda, K Tokuda, Speech parameter generation algorithm considering global variance for HMM-based speech synthesis. IEICE - Trans. Info. Syst. Arch **E90-D**(5), 816–824 (2007)
27. K Tokuda, T Yoshimura, T Masuko, T Kobayashi, T Kitamura, *Speech Parameter Generation Algorithms for HMM-based Speech Synthesis, in ICASSP*, vol. 3 (Istanbul, 2000), pp. 1315–1318
28. K Tokuda, T Kobayashi, S Imai, Speech parameter generation from HMM using dynamic features, in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1 (Detroit, Michigan, USA, 1995), pp. 660–663
29. Comparing glottal-flow-excited statistical parametric speech synthesis methods, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Vancouver, British Columbia, Canada, 2013), pp. 7830–7834
30. T Yoshimura, K Tokuda, T Masuko, T Kobayashi, T Kitamura, Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis, in *Proceedings of Eurospeech* (1999), pp. 2347–2350
31. SJ Young, JJ Odell, PC Woodland, Tree-based state tying for high accuracy acoustic modeling, in *Proceedings of the workshop on Human Language Technology, Association for Computational Linguistics* (1994), pp. 307–312
32. CJ Leggetter, PC Woodland, Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. Comput. Speech. Lang. **9**(2), (1995)
33. VV Digalakis, LG Neumeyer, Speaker adaptation using combined transformation and Bayesian methods. IEEE Trans. Speech. Audio. Process. **4**(4), 294–300 (1996)
34. H Zen, N Braunschweiler, S Buchholz, MJ Gales, K Knill, S Krstulovic, J Latorre, Statistical parametric speech synthesis based on speaker and language factorization. IEEE Transactions. Audio. Speech. Lang. Process. **20**(6), 1713–1724 (2012)
35. T Koriyama, T Nose, T Kobayashi, *Statistical parametric speech synthesis based on Gaussian process regression, IEEE Journal of Selected Topics in Signal Processing* (2013), pp. 1–11
36. K Yu, F Mairesse, S Young, Word-level emphasis modeling in HMM-based speech synthesis, in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)* (Dallas, Texas, USA, 2010), pp. 4238–4241
37. H Zen, N Braunschweiler, Context-dependent additive log f_0 model for HMM-based speech synthesis, in *INTERSPEECH* (Brighton, United Kingdom, 2009), pp. 2091–2094
38. S Sakai, *Additive modeling of English f0 contour for speech synthesis, in Proceedings of ICASSP* (Las Vegas, Nevada, USA, 2008), pp. 277–280
39. Y Qian, H Liang, FK Soong, Generating natural F0 trajectory with additive trees, in *INTERSPEECH* (Brisbane, Australia, 2008), pp. 2126–2129
40. YJ Wu, F Soong, Modeling pitch trajectory by hierarchical HMM with minimum generation error training, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Kyoto, Japan, 2012), pp. 4017–4020
41. AL Berger, VJD Pietra, SAD Pietra, A maximum entropy approach to natural language processing. Computer Ling **22**, 39–71 (1996)
42. A Borthwick, *A maximum entropy approach to named entity recognition, PhD dissertation (New York University)*, 1999
43. V Rangarajan, S Narayanan, S Bangalore, *Exploiting acoustic and syntactic features for prosody labeling in a maximum entropy framework, in Proceedings of NAACL HLT*, 2007, pp. 1–8
44. A Ratnaparkhi, A maximum entropy model for part-of-speech tagging, in Proceedings of the conference on empirical methods in natural language processing. **1**, 133–142 (1996)
45. JJ Odell, *The use of context in large vocabulary speech recognition, PhD dissertation (Cambridge University)*, 1995
46. K Shinoda, W Takao, MDL-based context-dependent subword modeling for speech recognition. J. Acoust. Soc. Jpn **21**(2), 79–86 (2000)
47. K Oura, H Zen, Y Nankaku, A Lee, K Tokuda, A covariance-tying technique for HMM-based speech synthesis. J. IEICE **E93-D**(3), 595–601 (2010)
48. J Nocedal, JW Stephen, *Numerical Optimization* (Book of Springer, USA, 1999)
49. M Bijankhan, J Sheikhzadegan, MR Roohani, Y Samareh, C Lucas, M Tebiani, The speech database of Farsi spoken language, in *Proceedings of 5th Australian International Conference on Speech Science and Technology (SST)* (1994), pp. 826–831
50. J Ghomeshi, Non-projecting nouns and the ezafe: construction in Persian. Nat. Lang. Ling. Theor. **15**(4), 729–788 (1997)
51. R Kubichek, Mel-cepstral distance measure for objective speech quality assessment, in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, vol. 1, 1993, pp. 125–128
52. B Picart, T Drugman, T Dutoit, *Continuous control of the degree of articulation in HMM-based speech synthesis, 12th Annual Conference of the International Speech Communication Association (ISCA)* (INTERSPEECH, Florence, Italy, 2011), pp. 1797–1800
53. J Yamagishi, *Average-Voice-Based Speech Synthesis, PhD dissertation* (Tokyo Institute of 1362 Technology, Yokohama, 2006)
54. J Yamagishi, O Watts, *The CSTR/EMIME HTS system for Blizzard challenge, in Proceedings of Blizzard Challenge 2010* (Kyoto, Japan, 2010), pp. 1–6

doi:10.1186/1687-4722-2014-12
Cite this article as: Khorram *et al.*: Context-dependent acoustic modeling based on hidden maximum entropy model for statistical parametric speech synthesis. *EURASIP Journal on Audio, Speech, and Music Processing* 2014 **2014**:12.