Soft context clustering for F0 modeling in HMM-based speech synthesis

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Soft context clustering for F0 modeling in HMM-based speech synthesis

Soheil Khorram[1*], Hossein Sameti[1] and Simon King[2]

Abstract

This paper proposes the use of a new binary decision tree, which we call a *soft decision tree*, to improve generalization performance compared to the conventional 'hard' decision tree method that is used to cluster context-dependent model parameters in statistical parametric speech synthesis. We apply the method to improve the modeling of fundamental frequency, which is an important factor in synthesizing natural-sounding high-quality speech. Conventionally, hard decision tree-clustered *hidden Markov models (HMMs)* are used, in which each model parameter is assigned to a single leaf node. However, this 'divide-and-conquer' approach leads to data sparsity, with the consequence that it suffers from poor generalization, meaning that it is unable to accurately predict parameters for models of unseen contexts: the hard decision tree is a weak function approximator. To alleviate this, we propose the soft decision tree, which is a binary decision tree with soft decisions at the internal nodes. In this soft clustering method, internal nodes select both their children with certain membership degrees; therefore, each node can be viewed as a fuzzy set with a context-dependent membership function. The soft decision tree improves model generalization and provides a superior function approximator because it is able to assign each context to several overlapped leaves. In order to use such a soft decision tree to predict the parameters of the HMM output probability distribution, we derive the smoothest (maximum entropy) distribution which captures all partial first-order moments and a global second-order moment of the training samples. Employing such a soft decision tree architecture with maximum entropy distributions, a novel speech synthesis system is trained using *maximum likelihood (ML)* parameter re-estimation and synthesis is achieved via maximum output probability parameter generation. In addition, a soft decision tree construction algorithm optimizing a log-likelihood measure is developed. Both subjective and objective evaluations were conducted and indicate a considerable improvement over the conventional method.

Keywords: Context clustering; Decision tree-based clustering; F0 modeling; Hidden Markov model; HMM; HMM-based speech synthesis; Maximum entropy model; Soft decision tree; Soft context clustering; Statistical parametric speech synthesis

1 Introduction

Demand for natural and high-quality speech-based human-computer interaction is increasing due to applications including speech-based virtual assistants for mobile devices. Speech synthesis plays a significant role, not only in transmitting factual information, but also as the outward 'face' of the application: the naturalness of the synthesis affects overall user satisfaction. Speech synthesis from text is usually achieved via an intermediate linguistic specification [1], which can be thought of as a collection of contextual factors - such as phonetic and prosodic properties of the current, preceding, and following segment - which have been derived from the text. Here, we are concerned only with the conversion of this linguistic specification to a speech waveform. In order to perform this conversion, several methods have been proposed [2], of which *statistical parametric speech synthesis (SPSS)* [3,4] has been dominant, at least in research terms, for the last decade or more.

Figure 1 portrays the overall architecture of a typical SPSS system, which comprises two distinct phases [3,4], namely, *training* and *synthesis*. The training phase starts with the extraction of acoustic features and the linguistic specification (i.e., contextual factors) for all training

* Correspondence: khorram@ce.sharif.edu
[1]Department of Computer Engineering, Sharif University of Technology, Azadi Avenue, Tehran 14588, Iran
Full list of author information is available at the end of the article

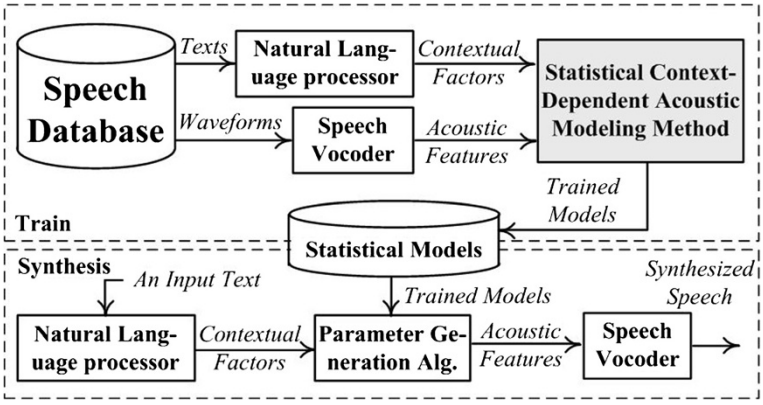

Figure 1 Overall block diagram of a typical SPSS.

utterances. Waveforms are converted to a compact set of acoustic features using a vocoder (such as *MELP* [5], *STRAIGHT* [6], *DSM* [7,8], or *HNM* [9]), and simultaneously, all texts are expanded into contextual factors using a natural language pre-processing front end [10]. Thereafter, the training phase proceeds to the context-dependent statistical modeling step in which the dependencies between extracted acoustic features and contextual factors are modeled through context-dependent statistical models [11]. It is important to note that, because of the extreme sparsity in the contextual feature space, the decision tree (DT) used to cluster the model parameters is the critical component in the statistical modeling.

In the synthesis phase, contextual factors are obtained for the input text, and the decision tree is used to obtain the corresponding trained model parameters, using which a *parameter generation (PG)* algorithm [11-14] generates acoustic feature trajectories. These are then converted to a speech waveform using the vocoder.

In contrast to concatenative synthesis [15], which stores speech waveforms, the parametric representation in SPSS has several potential advantages, including flexibility in changing voice characteristics [3], speaker and style adaptation [16-19], easier multilingual support [20-22], superior coverage of acoustic space [3], reduced memory footprint [3], and better robustness to low-quality speech recordings [23].

Though compressing a human voice into a compact statistical model offers the abovementioned advantages over concatenation of waveforms, there remains one major shortcoming: lower quality synthetic speech. This is caused by one or more of the blocks shown in Figure 1, e.g., inadequate acoustic coverage of training utterances, noisy speech database [24,25], errors in natural language processing, inadequate contextual factors [26], inaccurate statistical modeling [3], the PG algorithm [11-14], or vocoding distortion [5-9]. Here, we propose improvements to the statistical modeling (the shaded block in Figure 1) and specifically for the F0 stream.

Hidden Markov model (HMM)-based speech synthesis [27-33] models not only the spectrum, but also the excitation and duration in a unified framework of *context-dependent* [34,35] *multi-space probability distribution* [36] *hidden semi-Markov models (HSMMs)* [37]. More precisely, an independent binary-branching hard decision tree is constructed for each stream of acoustic features (spectrum, aperiodic energy, and fundamental frequency). In the case of F0, a multi-space probability distribution [36] is associated with each leaf of the decision tree. Contextual space (which is very large and very sparse due the great number of contextual factors typically employed) is divided by the decision trees into multiple hard (i.e., non-overlapping) clusters; each cluster is a group of context-dependent HMM states that share the same output probability distribution.

Hard decision tree-based context clustering, which is the standard approach to F0 modeling, has *poor generalization* [38]. In other words, this structure cannot accurately predict the parameters of models of unseen contexts, given the very limited subset of contexts observed in the training data. In order to predict acoustic variations with high generalization capability, the model has to be able to express a large number of robust distributions (i.e., a large number of distributions, but such that each one can be trained from a sufficient number of training samples). In hard decision trees, increasing the number of distributions by growing the depth of the tree reduces the number of training samples assigned to each leaf, and thus, the robustness of the distributions is weakened. This problem stems from the fact that the hard decision tree structure assigns each model parameter to exactly one cluster (corresponding to a small part of the large contextual space): each training sample contributes to the estimation of only one set of model parameters (one mean vector and one covariance matrix). Our hypothesis is that by enabling each training

sample to influence multiple sets of model parameters (and thus cover a larger portion of contextual space), generalization to unseen contexts would be improved.

1.1 Related work

Several attempts have already been made to alleviate the limitations of F0 modeling in standard decision tree-clustered H(S)MMs. One of these is the use of *deep neural networks (DNNs)* [38,39] which are able to approximate complex acoustic feature-to-linguistic context dependencies by employing many hidden layers - contrast this with decision trees that cannot efficiently represent something as simple as XOR or multiplexing [38], i.e., they must use an excessive number of leaves to capture such relationships and thus over-fragment the already sparse training data [38]. DNNs are also able to represent non-binary contextual features, whereas decision trees generally only use binary splits. Other deep learning approaches such as *restricted Boltzmann machines (RBMs)* and *deep belief networks (DBNs)* have also been demonstrated to be effective generative models when applied to speech synthesis [40,41].

Speech synthesis based on *Gaussian process regression (GPR)* [42] is another new technique that has recently been introduced to alleviate basic limitations of HMM-based speech synthesis. The goal of GPR is to remove the incorrect stationarity assumption of state output distribution in HMM-based speech synthesis. GPR uses frame-level contextual factors - such as position of the frame within the phone, and articulatory features - to estimate frame-level acoustic trajectories [42]. GPR can directly express complex context dependencies without needing decision tree structures and is able to use all contextual factors of all types simultaneously; therefore, it has the potential to provide better generalization.

In [43], a new system is proposed that replaces the usual *maximum likelihood (ML)* point estimate of the model parameters with a *variational Bayesian* method. Their system outperforms the usual approach when the amount of training data is limited, thus demonstrating superior generalization.

F0 modeling with *additive* structures has also been used to express the relationship between contextual factors and the F0 trajectory [44-54]. *Contextual additive* modeling [45-48] assumes model parameters to be a sum of multiple independent components, each having different context dependencies; therefore, different decision trees have to be trained for them. The contextual additive model is able to exploit contextual factors more efficiently, because mean vectors and covariance matrices of the predicted distributions are the sum of mean vectors and covariance matrices of the additive components [45]: each training sample contributes to more than one model parameter. Takaki et al. [46-48] used an additive structure for spectral modeling and reported that it has a high computational cost. To alleviate this, they proposed *covariance parameter tying* and a simplified likelihood calculation algorithm using the *matrix inversion lemma*. Though the contextual additive model [45-48] was originally proposed for spectral modeling, it could be used for F0 trajectories.

Zen et al. [49] also developed an additive F0 modeling structure with multiple components for mean vectors and a single component for variance values. Accordingly, multiple decision trees were trained for the mean vectors, and just one decision tree was built for the variance values. In their system, different sets of contextual factors were used for different additive components and all trees were built concurrently. Similarly, [50] proposed another additive structure with multiple decision trees, but a *minimum generation error (MGE)* measure was used as the selection criterion instead of the more common *maximum likelihood (ML)* measure. In [51], an additive model with three different layers, including intonational phrase, word level, and pitch accent, was designed. All three components were trained concurrently using a *regularized least square error* measure. Qian et al. [52] proposed to use a new gradient-based tree-boosting approach with a view to training multiple additive regression trees. Their decision trees were built in successive stages to minimize the squared error.

Some studies [53,54] have also highlighted another important problem of the common decision tree-based F0 modeling: its *deficiency in capturing the effect of contextual features that are poorly represented in the training database*. These features (i.e., questions used in the decision tree splits) have little influence on the likelihood criterion and hence will not be selected by the usual decision tree construction algorithms [53,54]. One obvious technique to solve this problem is to build the decision tree using a two-stage algorithm [53]. In the first stage, all splits are made only with these under-represented contextual factors. This stage captures the influence of such factors, even though they are rare. In the second stage, the well-represented factors are employed. This procedure is not efficient, since the first stage reduces the amount of the training data available for modeling the dependency between well-represented contextual factors and F0 [54]. Context adaptive training with factorized decision trees [54] is another method designed to exploit rare features more effectively. There, cluster adaptive training [55] is employed such that an average model is built and then this general model is adapted using a set of transforms. In fact, well-represented contextual factors contribute to generate the average model, and rare contextual questions are taken into account for the transforms. Due to the use of cluster adaptive training, this structure also is able to improve context generalization.

1.2 Scope of the paper

Numerous binary and non-binary contextual factors are generally taken into account in modeling F0. Conventional HMM-based speech synthesis converts all non-binary contextual factors to multiple binary questions (i.e., potential decision tree splits). As mentioned earlier, this structure may suffer from inadequate context generalization. To alleviate this deficiency, we propose the direct use of non-binary contextual factors in a soft decision tree framework [56,57]. The proposed soft decision tree structure is an innovative binary decision tree with soft questions at each non-terminal node. Both children are selected with a specific membership degree. In contrast to a hard decision tree that partitions contextual factor space into hard contextual regions, the proposed soft decision tree is able to provide soft - i.e., overlapping - clusters. In this structure, each context will be assigned to several terminal leaves with certain membership functions, and consequently, each training sample affects multiple model parameters, and generalization should be improved.

The rest of the paper is organized as follows: Section 2 presents the classical hard decision tree approach to F0 modeling in statistical parametric speech synthesis. The proposed soft context-clustered HMM structure and details of the associated speech synthesis system that employs such trees are provided in Section 3. Section 4 reports the experiments and results, and Section 5 concludes the paper.

2 F0 modeling using hard decision trees

This section describes the predominant framework for F0 modeling in HMM-based speech synthesis, which is the same framework used for the spectral envelope, aperiodic energy, and duration. This section also sets out the notation, algorithms, and structures required for subsequent sections.

2.1 F0 modeling in the HMM framework

Typically, F0 along with its delta and delta-delta derivatives form three streams[a] of a *context-dependent* [34,35] *multi-space probability distribution (MSD)* [36] *left-to-right without skip transitions HSMM* [58,37] (which for obvious reasons, we shorten to simply 'HMM' in this paper). This model structure generates acoustic trajectories of a unit (e.g., phoneme) by emitting observations from hidden states. The output distribution of the state is a context-dependent multi-space Gaussian distribution [36], and these are clustered into groups of related contexts using a decision tree in order to reduce the number of free parameters and allow the modeling of unseen contexts. For notational simplicity, we limit our discussion here to an HMM with just one stream. Generalizing this to the multi-stream case is straightforward.

Figure 2 shows the equivalent dynamic Bayesian network (DBN) for such an HMM [59]. In this figure, q_t, o_t, and g_t respectively represent the state index, the acoustic feature vector, and the space index [36] in time t.

When using MSD output distributions with two spaces - for defined and undefined values - the space index is an observed value equal to the voicing label. The figure also introduces c_j, d_j, and t_j which are the contextual factors, the duration, and the last frame index of the jth state (clearly, $d_j = t_j - t_{j-1}$). Note that state boundaries are latent variables and have to be trained in an unsupervised manner using the expectation maximization (EM) [60] algorithm.

According to this figure, the HMM is simply specified through three sets of fundamental distributions: i) state duration probability distribution ($p_j(d_j|c_j)$), ii) voicing (space) probability distribution ($w_j(g_t|c_j)$), and iii) output probability distribution given voicing labels ($b_j(o_t|g_t,c_j)$). Using these fundamental distributions and considering the graphical model represented by Figure 2, the likelihood of a given utterance with observations (o, g, c) can be factorized as

$$p(o,g|c;\lambda) = \sum_{t_1,t_2,\ldots,t_J} \prod_{j=1}^{J} p_j(d_j|c_j) \prod_{j=1}^{J} w_j(g_t|c_j) b_j(o_t|g_t,c_j), \quad (1)$$

where J and λ denote the total number of states and the model parameters, respectively.

Now, assume g_t takes two values: '1' for voiced frames and '0' for unvoiced regions; also, let b_j and p_j be expressed through Gaussian distributions. Therefore, the above utterance likelihood can be rewritten as

$$p(o,g|c;\lambda) = \sum_{(t_1,t_2,\ldots,t_J)} \prod_{j=1}^{J} \mathcal{N}\left(d_j;\bar{m}_j,\bar{\sigma}_j^2\right) \prod_{t=t_{j-1}}^{t_j} \left[g_t \bar{w}_j \mathcal{N}\left(o_j;\bar{\mu}_j,\bar{\Sigma}_j\right) + (1-g_t)(1-\bar{w}_j)\right], \quad (2)$$

where $\mathcal{N}(.;\mu,\Sigma)$ represents a Gaussian distribution with mean vector μ, and covariance matrix Σ. In this equation, duration and output distributions are parameterized by duration mean $(\bar{m}_j)$, duration variance $\left(\bar{\sigma}_j^2\right)$, voicing probability ($\bar{w}_j$), output mean vector $\left(\bar{\mu}_j\right)$, and observation covariance matrix $\left(\bar{\Sigma}_j\right)$.

As previously mentioned, a canonical decision tree structure is used to express the fundamental distributions. Assume $I_l^d(c_j)$ and $I_l^o(c_j)$ are defined as binary indicator functions of decision trees trained for duration and output distributions where l and c_j are, respectively, the leaf index and the contextual factors extracted for the jth state. In other words, $I_l^d(c_j)$ and $I_l^o(c_j)$ determine whether the jth state is assigned to the lth leaf of the duration and observation decision trees or not. Using these decision tree indicator functions, the HMM parameters can be expressed by

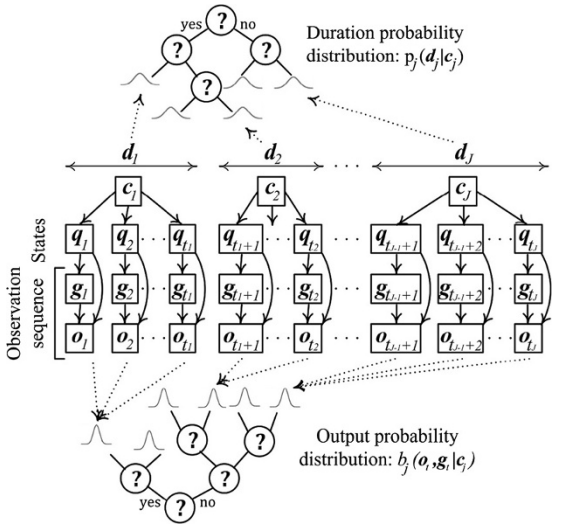

Figure 2 Graphical model of the common HMM. Rectangular nodes and solid arrows demonstrate random variables and their dependencies. Dotted arrows show the tying structure through decision trees.

$$\begin{aligned}\bar{m}_j &= \sum\nolimits_l I_l^d(c_j) m_l,\ \ \bar{\sigma}_j^2 = \sum\nolimits_l I_l^d(c_j)\sigma_l^2,\\ \bar{w}_j &= \sum\nolimits_l I_l^o(c_j) w_l,\ \ \bar{\mu}_j = \sum\nolimits_l I_l^o(c_j)\mu_l,\ \ \bar{\Sigma}_j = \sum\nolimits_l I_l^o(c_j)\Sigma_l,\end{aligned} \tag{3}$$

where m_l and σ_l^2 are duration mean and variance values lying in the lth leaf of the duration decision tree. Similarly, w_l, μ_l, and Σ_l represent parameters of the voicing and output probability distributions that are trained for the lth leaf of the output decision tree.

2.2 HMM parameter re-estimation

The ML criterion is commonly used to estimate model parameters of HMM. However, state boundaries are latent, and therefore, the EM algorithm has to be adopted. Given N i.i.d. utterances, $\{(o^n,\ g^n)\}_{n=1}^N$, along with their corresponding contextual factors, $\{c^n\}_{n=1}^N$, the EM algorithm leads to the following re-estimation formulas:

$$\begin{aligned}\hat{m}_l &= \frac{\sum_{n=1}^N\sum_{j=1}^{J^n} I_l^d\left(c_j^n\right)\sum_{t_j,t_{j-1}} \chi_j^n(t_j,t_{j-1}\ )\left[t_j - t_{j-1}\right]}{\sum_{n=1}^N\sum_{j=1}^{J^n} I_l^d\left(c_j^n\right)\sum_{t_j,t_{j-1}} \chi_j^n(t_j,t_{j-1})},\\ \hat{\sigma}_l^2 &= \frac{\sum_{n=1}^N\sum_{j=1}^{J^n} I_l^d\left(c_j^n\right)\sum_{t_j,t_{j-1}} \chi_j^n(t_j,t_{j-1})\left[t_j - t_{j-1} - \hat{m}_l\right]^2}{\sum_{n=1}^N\sum_{j=1}^{J^n} I_l^d\left(c_j^n\right)\sum_{t_j,t_{j-1}} \chi_j^n(t_j,t_{j-1})},\\ \hat{\mu}_l &= \frac{\sum_{n=1}^N\sum_{j=1}^{J^n} I_l^o\left(c_j^n\right)\sum_t\ \gamma_j^n(t) g_t^n\left[o_t^n\right]}{\sum_{n=1}^N\sum_{j=1}^{J^n} I_l^o\left(c_j^n\right)\sum_t\ \gamma_j^n(t) g_t^n},\\ \hat{\Sigma}_l &= \frac{\sum_{n=1}^N\sum_{j=1}^{J^n} I_l^o\left(c_j^n\right)\sum_t\ \gamma_j^n(t) g_t^n\left[\left(o_t^n - \hat{\mu}_l\right)\left(o_t^n - \hat{\mu}_l\right)^{\mathrm{T}}\right]}{\sum_{n=1}^N\sum_{j=1}^{J^n} I_l^o\left(c_j^n\right)\sum_t\ \gamma_j^n(t) g_t^n},\\ \hat{w}_l &= \frac{\sum_{n=1}^N\sum_{j=1}^{J^n} I_l^o\left(c_j^n\right)\sum_t\ \gamma_j^n(t) g_t^n}{\sum_{n=1}^N\sum_{j=1}^{J^n} I_l^o\left(c_j^n\right)\sum_t\ \gamma_j^n(t)},\end{aligned} \tag{4}$$

where $\hat{m}_l$, $\hat{\sigma}_l^2$, $\hat{\mu}_l$, $\hat{\Sigma}_l$, and $\hat{w}_l$ are new values of m_l, σ_l^2, μ_l, Σ, and w_l during EM algorithm. Also, $\chi_j(t_j, t_{j-1})$ is the probability of occupying the jth state from time t_{j-1} to t_j, and $\gamma_j(t)$ denotes the posterior probability of being in state j at time t. These probabilities are calculated through the well-known *forward-backward* algorithms. It should be noted that the publically available *HMM-based speech synthesis system (HTS)* [61] has been implemented based on the algorithms expressed in [62]. These algorithms were originally proposed by Ferguson [63] and were refined by Levinson [64]. A more efficient version of the forward-backward algorithm has recently been proposed by Yu et al. [65].

2.3 Decision tree-based state clustering

In order to capture the context dependencies inherent in the acoustic features, canonical decision trees are typically incorporated in the HMM framework. Decision trees are constructed iteratively through a greedy and top-down procedure which maximizes the log-likelihood criterion [34,35]. The procedure starts with a single root node representing all contexts. In each iteration, an optimum pair of terminal node and question is selected so that splitting the terminal node by the selected question results in the largest log-likelihood increase. The splitting procedure is continued until a termination criterion (such as *minimum description length (MDL)* [66]) is satisfied. The overall log-likelihood increase $\delta\mathcal{L}$, achieved by splitting a parent node l_1 into two children l_2 and l_3, is simply obtained by the following equation [34]:

$$\begin{aligned}\delta\mathcal{L} = &\frac{1}{2}\log\left(\left|\hat{\Sigma}_{l_1}\right|\right)\sum\nolimits_{n=1}^N\sum\nolimits_{j=1}^{J^n} I_{l1}^o\left(c_j^n\right)\sum\nolimits_t \gamma_j^n(t) -\\ &\sum\nolimits_{l\in\{l_2,l_3\}}\frac{1}{2}\log\left(\left|\hat{\Sigma}_l\right|\right)\sum\nolimits_{n=1}^N\sum\nolimits_{j=1}^{J^n} I_l^o\left(c_j^n\right)\sum\nolimits_t \gamma_j^n(t),\end{aligned} \tag{5}$$

where superscript n is an index defined for the number of training utterances. It should be noted that in order to obtain the above likelihood increase expression, the following simplifying assumptions have to be made [34]: 1 - The values of occupation probabilities are assumed to be fixed during the clustering procedure [34]. 2 - The overall likelihood measure is supposed to be approximated by a simple average of the log likelihoods weighted by the posterior probabilities [34]. These assumptions make the calculation of $\delta\mathcal{L}$ possible for all pairs of terminal nodes and questions.

3 Soft context-clustered HMM

Generally, decision tree is the term for a hierarchical structure consisting of internal nodes and terminal leaves. In a canonical hard binary decision tree, used for acoustic modeling, each terminal node carries a distribution

that captures the statistical characteristics of a context cluster. Also, for a given context c, each internal node m applies a binary test $f_m(c)$ and chooses one of its children based on the result of the test. Let $I_m(c)$ be the indicator function defined for the node m. Also, assume that $I_{m_L}(c)$ and $I_{m_R}(c)$ represent the indicator functions lying on its left and right children; $I_{m_L}(c)$ and $I_{m_R}(c)$ are obtained by

$$I_{m_L}(c) \stackrel{\text{def}}{=\!=} \begin{cases} I_m(c) & \text{if } f_m(c) = \text{true} \\ 0 & \text{if } f_m(c) = \text{false} \end{cases},$$
$$I_{m_R}(c) \stackrel{\text{def}}{=\!=} \begin{cases} 0 & \text{if } f_m(c) = \text{true} \\ I_m(c) & \text{if } f_m(c) = \text{false} \end{cases}. \tag{6}$$

Accordingly, to determine the distribution of a given contextual factor, we need to start from the root node and recursively apply the test at each internal node and select one of the two branches depending on the outcome. This process is repeated iteratively until a leaf node is hit at which point the distribution of the leaf is considered as the output probability distribution. Therefore, for each context, just one path from the root to a terminal node is always traversed, and each context is hereby assigned to one leaf and affects the distribution of that single leaf. In order to improve the performance of the canonical decision tree, this paper proposes the soft binary decision tree structure which is able to establish several fuzzy paths from the root to multiple leaves.

3.1 Soft context-clustered HMM structure

The soft decision tree applies soft decisions $\tilde{f}_m(c)$ in its internal nodes m and redirects all samples to both children, but with certain membership degrees computed by $\tilde{f}_m(c)$ and $1-\tilde{f}_m(c)$. In fact, each node of a soft decision tree represents a fuzzy subset of contextual factor space; therefore, each context belongs to several nodes with a membership degree. More precisely, when we are traversing the node m for the given context c, a soft question $\tilde{f}_m(c)$ represents the membership grade of the left child, and clearly, $1-\tilde{f}_m(c)$ computes the degree of selecting the right child.

In both hard and soft decision tree-based HMMs, initially, a set of contextual factors have to be defined and extracted for all training utterances. Thereafter, as opposed to the hard decision tree that requires hard questions $f_m(c)$, here, we have to design a great number of soft questions (soft tests) $\tilde{f}_m(c)$ for each contextual factor. These questions are finally assigned to the internal nodes of the decision tree and make fuzzy decisions to select among their children instead of the common crisp decisions.

As it is realized from the above discussion, all terminal leaves may be active for an arbitrary context; as a consequence, it is necessary to generalize the indicator function $I_m(c)$ expressed by Equation 3 to the membership function of assigning context c to the node m. This membership function is denoted by $\tilde{I}_m(c)$ and can be computed through the following recursion:

$$\begin{aligned} &\text{Initialization}: \ \tilde{I}_{\text{root}}(c) = 1, \\ &\text{Recursion}: \begin{cases} \tilde{I}_{m_L}(c) = \tilde{f}_m(c)\tilde{I}_m(c) \\ \tilde{I}_{m_R}(c) = \left(1-\tilde{f}_m(c)\right)\tilde{I}_m(c) \end{cases} \end{aligned} \tag{7}$$

where m_L and m_R are the left and right children of node m. According to the above recursion, all the membership degrees can be calculated by traversing the tree in a preorder style. The traversing procedure starts with setting the membership degree of the root to 1. After observing a node m and determining its membership degree $\tilde{I}_m(j)$, its left m_L and right m_R children are observed. If the node is a left child, its membership degree is calculated through $\tilde{f}_m(c)\tilde{I}_m(c)$; otherwise, the procedure returns $\left(1-\tilde{f}_m(c)\right)\tilde{I}_m(c)$, in which m is the parent node.

In the training phase, soft decisions $\tilde{f}_m(c)$ are selected from a set of predefined contextual functions. These functions must hold the following limitation for all contextual factors:

$$\forall \ m, \ c, \ 0 \le \tilde{f}_m(c) \le 1. \tag{8}$$

The above constraint has to be taken into account during the procedure of defining soft questions. That is, we are not allowed to employ soft questions with a value greater than 1 or less than 0; thus, a normalization step is required for some questions before starting decision tree-based clustering.

3.2 Soft context-clustered HMM distribution

The proposed soft context-clustered HMM exploits the same structure and graphical model as the original hard decision tree-based HMM, and thus, the model likelihood expression given by Equation 1 is also valid for the proposed model. The only difference between the conventional and the proposed approaches lies in the method of capturing context dependencies inherent in the F0 trajectory. More specifically, the method of representing output distribution $b_j(\cdot)$ in Equation 1 is different. The goal of this section is to find this probability distribution for the soft decision tree structure described in the previous section. With a view to providing an efficient context generalization, this section derives the smoothest distribution that is able to accurately express the behavior of the F0 trajectory. To estimate the smoothest distribution, the *maximum entropy*

model (MEM) [67,68], presented in the next subsection, is employed.

3.2.1 Maximum entropy-based distributions

Our task is to estimate the distribution of the observation vectors. The maximum entropy principle states that an efficient estimate is the one that maximizes entropy (uncertainty) subject to our knowledge about the observation vectors. This knowledge normally appears in the form of some constraints that make the distribution consistent with sufficient statistics of the observation vectors [67]. Let us now derive a simple maximum entropy model for the output distribution given voicing labels, $b_j(o_t|g_t, c_j)$. Suppose the training utterances consist of T i.i.d. voicing labels $\{g_t\}_{t=1}^{T}$ and D-dimensional output feature vectors $\{o_t\}_{t=1}^{T}$ that may be influenced by some contextual information $\{c_t\}_{t=1}^{T}$. Also, the contextual information is clustered through a soft decision tree structure with the total number of L leaves partitioning the contextual factor space through the membership functions $\{\tilde{I}_l(.)\}_{l=1}^{L}$. The maximum entropy principle first imposes a set of constraints on the distribution and then chooses a distribution as close as possible to a uniform distribution by optimizing the entropy criterion [67]. Indeed, this modeling scheme finds the least biased distribution among all distributions that satisfy our constraints. In other words,

$$b(o|g,c)\overset{\text{def}}{=}\ \arg\max_b\ \mathcal{H}\{b(o|g,c)\}\ \ S.T.\ \text{constraints}. \tag{9}$$

where $\mathcal{H}$ is the entropy measure which is defined by

$$\mathcal{H}\{b(o|g,c)\}\overset{\text{def}}{=}-\sum_{t=1}^{T}\int_o b(o|g_t,c_t)\log\ b(o|g_t,c_t)do. \tag{10}$$

The constraints play a crucially important role in maximum entropy modeling. They ensure that the model captures the statistical characteristics of the training samples. In this paper, the following constraints are taken into account:

$$b(o|g,c)\overset{\text{def}}{=}\arg\max_b\mathcal{H}\{b(o|g,c)\}\ \ S.T.\ \left\{\begin{array}{c}\forall\ c,g\ \int_o b(o|g,c)=1\\ E\{goo^T\}=\bar{E}\{goo^T\}\\ \forall 1\le l\le L\ \ E\{\tilde{I}_l(c)go\}=\bar{E}\{\tilde{I}_l(c)go\}\end{array}\right\} \tag{11}$$

The first constraint ensures that the distributions sum to 1. Also, E and $\bar{E}$ indicate real and empirical mathematical expectations given by the following equations:

$$\begin{aligned}&E\{\tilde{I}_l(c)go\}=\sum_{t=1}^{T}\tilde{I}_l(c_t)g_t\int_o ob(o|g_t,c_t)do,\\ &\bar{E}\{\tilde{I}_l(c)go\}=\sum_{t=1}^{T}\tilde{I}_l(c_t)g_to_t,\\ &E\{goo^T\}=\sum_{t=1}^{T}g_t\int_o oo^Tb(o|g_t,c_t)do,\\ &\bar{E}\{goo^T\}=\sum_{t=1}^{T}g_to_to_t^T\end{aligned} \tag{12}$$

These constraints make the estimated distribution capture the partial first-order moments $E\{\tilde{I}_l(c)go\}$ and the global second-order moment $E\{goo^T\}$ of the training data in voiced frames (i.e., in frames where observation features o_t are defined and voicing label g_t is 1); therefore, the training phase of the maximum entropy model estimates the smoothest distribution that preserves the first- and second-order moments, expressed in Equation 9, of the training database. Moreover, the selected constraints lead to a simple expression for the output probability distributions that can be estimated efficiently.

In order to solve optimization problems with equality constraints, the Lagrange multipliers method can be applied. This method defines a new optimization function as follows:

$$\begin{aligned}&b(o|g,c)\overset{\text{def}}{=}\arg\max_b\mathcal{J}(b),\\ &\mathcal{J}(b)=\mathcal{H}\{b(o|g,c)\}+\lambda_{b0}\left[\int_o b(o|g,c)do-1\right]+\\ &\left[E\{go^T\Lambda o\}-\bar{E}\{go^T\Lambda o\}\right]+\sum_{l=1}^{L}\lambda_{bl}^T\left[E\{\tilde{I}_l(c)go\}\right.\\ &\qquad\left.-\bar{E}\{\tilde{I}_l(c)go\}\right]\end{aligned} \tag{13}$$

where $\mathcal{J}(b)$ represents the new optimization function; Also, λ_{b0}, λ_{b1}, and Λ are Lagrange multiplayers incorporated in the optimization function to remove the equality constraints.

Taking the derivatives of the optimization function $\mathcal{J}(b)$ with respect to the output probability distribution $b(o|g,c)$, and setting it to zero leads to the following equation:

$$\begin{aligned}\frac{\partial\mathcal{J}(b)}{\partial b}=\sum_{t=1}^{T}\int_o\left[-\log b(o|g_t,c_t)-1+\lambda_{b0}+g_to^T\Lambda o\right.\\ \left.+\sum_{l=1}^{L}\lambda_{bl}^T\tilde{I}_l(c_t)g_to\right]do=0\end{aligned}$$

An obvious solution satisfying the above equality is

$$\log\ b\ (o|g_t,\ c_t)=g_to^T\Lambda o+\sum_{l=1}^{L}\lambda_{bl}^T\tilde{I}_l(c_t)g_to+\lambda_{b0}-1.$$

Therefore, $b(o|g_t,c_t)$ is a simple Gaussian distribution that can be expressed by

$$b(o|g,c) = \mathcal{N}\left(o; \sum_{l=1}^{L} \tilde{I}_l(c)\mu_l, \Sigma\right) \quad (14)$$

where $\mathcal{N}$ indicates the Gaussian distribution; μ_l is a D-dimensional vector of mean parameters defined for the lth leaf; Also, Σ is a D-by-D covariance matrix that is used for all leaves.

In sum, each leaf of the soft decision tree carries a set of model parameters represented by μ_l that contributes to express the output probability distribution $b(o|g,c)$. The output probability $b(o|g,c)$ is simply approximated by a Gaussian distribution. This Gaussian distribution uses a unique context-independent covariance matrix Σ and a context-dependent mean vector. The mean component is obtained by linearly combining μ_l parameters (i.e., $\sum_{l=1}^{L} \tilde{I}_l(c)\mu_l$) and the weights of the linear combination are determined by the membership functions $\tilde{I}_l(c)$. In fact, the proposed maximum entropy-based output probability distribution is remarkably similar to the distribution expressed by the contextual additive structure that ties all covariance matrixes [46-48]. In the contextual additive method, similar to the proposed method, the output distribution has the form of Equation 14, but the contextual additive method exploits multiple hard decision trees [46] or a hard decision tree with overlapped leaves [47] instead of the proposed soft decision tree. In other words, in contextual additive structure, $\tilde{I}_l(c)$ indicates a leaf indicator function that may be 1 for multiple overlapped leaves, but in the proposed model, $\tilde{I}_l(c)$ is a real number, ranging from 0 to 1, that represents the membership degrees of a soft decision tree terminal node.

3.3 Parameter re-estimation

Having described the soft context-clustered HMM structure, it is now time to discuss its parameter re-estimation procedure. In the training phase, we are given a set of N i.i.d. training utterances containing acoustic features $\{o^n\}_{n=1}^{N}$, voicing labels $\{g^n\}_{n=1}^{N}$, and contextual factors $\{c^n\}_{n=1}^{N}$. The goal is to find the optimum set of model parameters $\hat{\lambda}$ which maximizes the log-likelihood measure:

$$\begin{aligned} &\hat{\lambda} \stackrel{\text{def}}{=} \operatorname{argmax}_{\lambda} \mathcal{L}(\lambda), \\ &\mathcal{L}(\lambda) \stackrel{\text{def}}{=} \sum_{n=1}^{N} \ln\ p(o^n, g^n | c^n; \lambda). \end{aligned} \quad (15)$$

This section assumes that the soft decision tree structure has been trained earlier and we just try to find the maximum log-likelihood estimate of its parameters λ, including $\{\mu_l\}_{l=1}^{L}$ and Σ. Training the optimum soft decision tree structure will be described in the next section. Similar to the classical HMM, the likelihood expression of Equation 1 leads to an extremely complex optimization problem with seemingly impossible direct solution. The main problem is that the distribution depends on the state boundaries which are latent. The EM technique offers an iterative algorithm which is able to overcome this problem. According to the EM technique, $\hat{\lambda}$ is obtained by iteratively maximizing an axillary function $\mathcal{Q}(\lambda;\lambda^r)$:

$$\begin{aligned} &\lambda^{r+1} = \operatorname{argmax}_{\lambda} \mathcal{Q}\ (\lambda;\ \lambda^r) \\ &\mathcal{Q}(\lambda;\lambda^r) = \sum_n \Big[\sum_{t_{j-1},t_j} \chi_j^n(t_j, t_{j-1}; \lambda^r) \log\ p_j(t_j - t_{j-1} | c_j^n) + \\ &\qquad \sum_t \sum_j \gamma_j^n(t;\lambda^r) \Big\{ \log\ w_j(g_t^n | c_j^n) \\ &\qquad + \log\ b_j(o_t^n | g_t^n, c_j^n) \Big\} \Big], \end{aligned} \quad (16)$$

where χ_j and γ_j are occupation probabilities defined in Section 2.2. Also, r is the index of the EM iterations, and n ranges over the utterance numbers. In order to estimate the optimum set of parameters, the partial derivatives of $\mathcal{Q}$ with respect to all model parameters λ have to be set to zero. These partial derivatives are calculated by considering the distribution introduced in Section 2.2 as follows:

$$\begin{aligned} \frac{\partial \mathcal{Q}(\lambda;\lambda^r)}{\partial \mu_\ell} &= \Sigma^{-1} \sum_n \sum_t \sum_j \gamma_j^n(t;\lambda^r) \\ &\qquad \tilde{I}_l\left(c_j^n\right) \left(o_t^n - \sum_{l=1}^{L} \tilde{I}_l\left(c_j^n\right) \mu_l\right), \\ \frac{\partial \mathcal{Q}(\lambda;\lambda^r)}{\partial \Sigma} &= \frac{1}{2} \Sigma^{-1} \sum_n \sum_t \sum_j \gamma_j^n(t;\lambda^r) \\ &\qquad \left\{ -1 + \left(o_t^n - \sum_{l=1}^{L} \tilde{I}_l\left(c_j^n\right) \mu_l\right)^T \right. \\ &\qquad \left. \left(o_t^n - \sum_{l=1}^{L} \tilde{I}_l\left(c_j^n\right) \mu_l\right) \Sigma^{-1} \right\} \end{aligned} \quad (17)$$

By setting these equations to zero, the maximum likelihood estimate of model parameters is obtained. According to these equations, the optimum vectors for mean parameters $\{\hat{\mu}_l\}_{l=1}^{L}$ can be simply calculated through solving the following system of equations:

$$R\hat{\mu} = P, \quad (18)$$

where $\hat{\mu}$ is a L-by-D matrix containing all mean parameters as

$$\hat{\mu} = [\hat{\mu}_1,\ \hat{\mu}_2, \ldots \hat{\mu}_L]^T. \quad (19)$$

Also, R and P are L-by-L and L-by-D matrixes defined by

$$
\begin{aligned}
&R = [r_{uv}]_{L\times L},\\
&r_{uv} = \sum_n \sum_j \tilde{I}_u\left(c_j^n\right)\tilde{I}_v\left(c_j^n\right)\sum_t \gamma_j^n(t;\lambda^r),\\
&P = [p_u]_{L\times L},\ \ p_u = \sum_n \sum_j \tilde{I}_u\left(c_j^n\right)\sum_t o_t^n \gamma_j^n(t;\lambda^r).
\end{aligned}
\tag{20}
$$

As it is realized from Equation 20, R represents the cross-correlation matrix of membership functions. This matrix is symmetric and positive definite; therefore, it is possible to solve the above system of equations efficiently using Cholesky decomposition.

Furthermore, by setting zero the partial derivatives of the auxiliary function $\mathcal{Q}(\lambda;\ \lambda^r)$ with respect to the globally tied covariance matrix Σ, the maximum likelihood estimate of Σ is calculated as follows:

$$
\hat{\Sigma} = \frac{\sum_n \sum_t \sum_j \gamma_j^n(t;\lambda^r)\left(o_t^n - \sum_{l=1}^{L}\tilde{I}_l\left(c_j^n\right)\mu_l\right)^T\left(o_t^n - \sum_{l=1}^{L}\tilde{I}_l\left(c_j^n\right)\mu_l\right)}{\sum_n \sum_t \sum_j \gamma_j^n(t;\lambda^r)}.
\tag{21}
$$

The above equations introduce a straightforward procedure to train the parameters of the output probability distribution factorized by a soft decision tree.

The next section discusses the procedure of constructing the proposed soft decision tree. In order to conduct a soft decision tree clustering algorithm, it is required to calculate the log-likelihood measure for the optimum model parameters. This optimum log-likelihood measure is expressed by

$$
\mathcal{L} \propto -\frac{1}{2}\ \log\left(\left|\hat{\Sigma}\right|\right)\sum_n \sum_t \sum_j \gamma_j^n(t)
\tag{22}
$$

where $|.|$ denotes the matrix determinant operator.

3.4 Soft context clustering algorithm

To automatically capture the dependencies between acoustic features and contextual factors, this section proposes a soft decision tree construction algorithm. Similar to the classical hard decision tree building algorithm, the soft decision tree is built iteratively through a greedy and top-down procedure which maximizes the log-likelihood measure.

The major advantage of the classical hard decision tree construction algorithm is that its terminal nodes can be split independently. In hard decision tree, terminal nodes represent non-overlapped regions of the contextual factor space; therefore, after splitting a leaf, all values obtained for other leaves are still valid, and it is not required to calculate them once again. This advantage causes the algorithm to be computationally tractable. However, in the soft decision tree construction procedure, the different terminal nodes may cover overlapped regions of the contextual space and splitting a leaf using a soft question affects the parameters of all other leaves. Consequently, as opposed to the conventional hard decision tree structure, here, after splitting a leaf, it is required to update all values obtained for all terminal nodes, and it needs tremendous amount of computations.

The procedure of the proposed soft decision tree construction algorithm is stated as follows:

Step 1. Create the root node embarrassing all samples of the training database.

Step 2. Split all terminal nodes using all questions and compute their optimum log-likelihood value. To compute the optimum log-likelihood value for each possible pair of leaf and question, the maximum likelihood estimate of mean parameters $\hat{\mu}$ has to be first obtained by Equation 18. Then, $\hat{\Sigma}$ is calculated through Equation 21, and Equation 22 is finally employed to find the optimum log-likelihood value.

Step 3. Select the best pair of terminal node and question that provides the maximum increase in log-likelihood measure. Thereafter, split the node using the question and estimate the maximum likelihood estimate of all model parameters.

Step 4. Stop the splitting procedure, if a predefined condition is satisfied (e.g., the increase in log-likelihood falls below a certain threshold).

Algorithm 1 summarizes the overall procedure of the proposed soft context clustering. As it is realized from the explained clustering algorithm, the proposed soft clustering procedure is dramatically similar to the classical clustering algorithm. Their main difference is in the number of evaluations that has to be performed during each iteration of the clustering procedure. In hard clustering, both newly generated leaves are just required to be evaluated, but in the soft clustering, all leaf nodes have to be evaluated. This fact increases the computational complexity of the soft clustering by an order of magnitude. Assume we intend to build a decision tree with L leaves. Also, we have defined Q questions. In this case, hard clustering requires $(2L-3)Q$ likelihood calculations to be performed, while soft clustering will be finished after $[L(L-1)/2]Q$ likelihood calculations.

It should be noted that the likelihood calculation in soft decision tree-based clustering is more complicated than the hard clustering; it is mainly due to the fact that calculating the inverse of the matrix R to solve the system of equations expressed by Equation 18 is computationally intractable. Takaki et al. [46] proposed a method to reduce the computational complexity of calculating this inverse problem. Their method exploits the matrix inversion lemma and can also be incorporated in the soft decision tree clustering procedure.

Algorithm 1 Proposed soft clustering algorithm

Inputs:

- c_j^n: Contextual factors of training utterances, $\forall 1 \le n \le N$ and $\forall 1 \le j \le J^n$, where n and j are utterance and state indexes.
- o_t^n: Observation features of training utterances, $\forall 1 \le n \le N$ and $\forall 1 \le t \le T^n$, where t denotes the frame index.
- $\gamma_j^n(t)$: State occupation probabilities, $\forall 1 \le n \le N$, $\forall 1 \le j \le J^n$, and $\forall 1 \le t \le T^n$.
- $\tilde{f}_m(c)$: predefined soft decisions, $\forall 1 \le m \le M$.

Outputs:

- L: Total number of decision tree terminal nodes.
- $\left\{\tilde{I}_l(c)\right\}_{l=1}^{L}$: Context-dependent membership functions of soft decision tree terminal nodes.

Soft context clustering algorithm:

Initialize $L=1$ and $f_1(c)=1$;

While the stopping criterion is not satisfied

1) For $l=1,2\ldots L$ (for each leaf)

For $m=1,2\ldots M$ (for each soft decision)

1') $\forall j,n \quad \tilde{I}_{L+1}(c_j^n) = \left(1-\tilde{f}_m(c_j^n)\right)\tilde{I}_l(c_j^n)$;

2') $\forall j,n \quad \tilde{I}_l(c_j^n) = \tilde{f}_m(c_j^n)\tilde{I}_l(c_j^n)$;

3') Compute *R*and *P* according to Equation 20;

4') Compute $\hat{\mu}$ according to Equation 18;

5') Compute $\hat{\Sigma}$ according to Equation 21;

6') Compute $\mathcal{L}(l,m)$ according to Equation 22;

2) $\hat{l}, \hat{m} = \arg\max_{l,m} \mathcal{L}(l,m)$;

3) $\forall j,n \quad \tilde{I}_{L+1}(c_j^n) = \tilde{f}_{\hat{m}}(c_j^n)\tilde{I}_{\hat{l}}(c_j^n)$;

4) $\forall j,n \quad \tilde{I}_{\hat{l}}(c_j^n) = \tilde{f}_{\hat{m}}(c_j^n)\tilde{I}_{\hat{l}}(c_j^n)$;

3.5 Simple sinusoidal regression

In order to clarify the soft clustering advantages, a simple sinusoidal regression problem is solved using both soft and hard decision tree structures in this section. Assume we have just one continuous contextual factor named c ranging from 0 to 1, and our goal is to approximate the following sinusoidal function:

$$o(c) = \sin(3\pi c - \pi) \times \sin(\pi c) + 0.05 \times r(c) \qquad (23)$$

where $o(c)$ represents the observation value for a given context c, and $r(c)$ is a normally distributed random noise with zero mean and unit variance. The 200 training samples shown in Figure 3a are independently drawn from Equation 23. Nineteen different contextual questions are defined to train the hard decision tree as follows:

$$\forall i \in \{1,2,\ldots 19\} q_i(c) = \begin{cases} 1 & c < i/20 \\ 0 & c \ge i/20 \end{cases} \qquad (24)$$

Therefore, each internal node of the hard decision tree structure has to select one of these hard

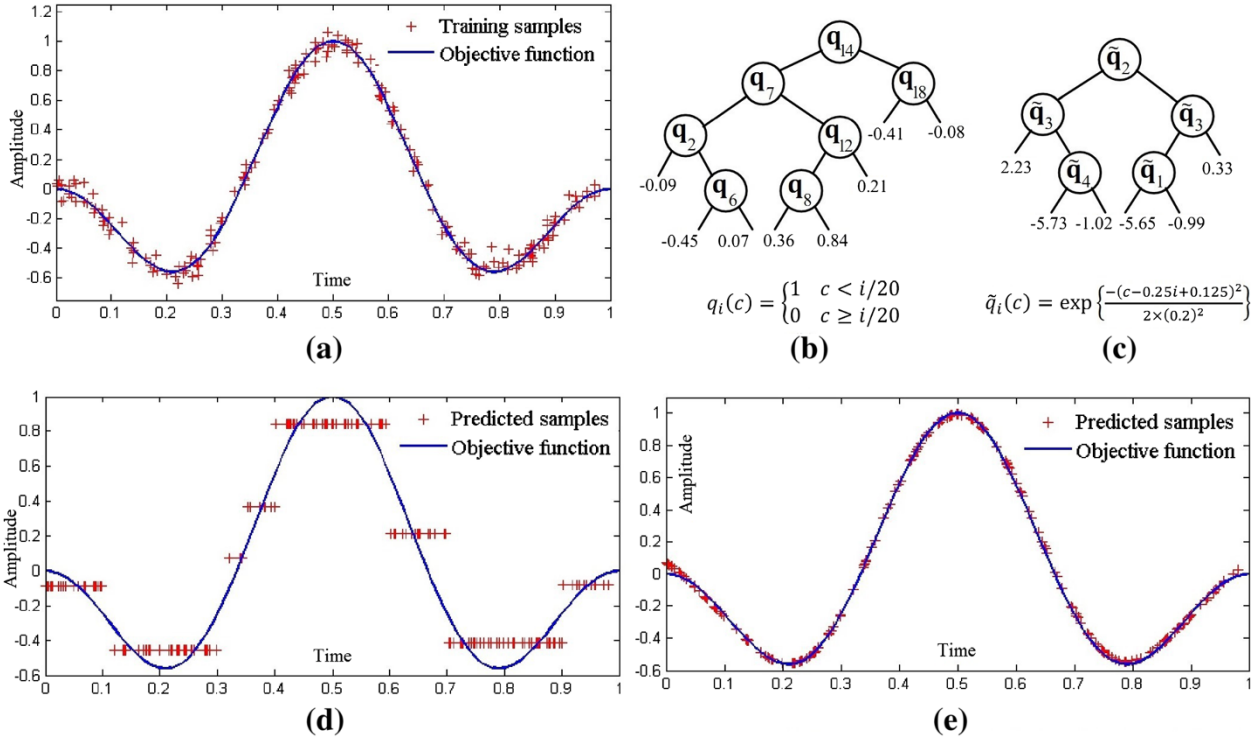

Figure 3 Samples and decision tree structures. (a) Data samples used to train the decision trees. The red dots and blue line are training samples and objective function, respectively. **(b)** Trained hard decision tree structure. **(c)** Trained soft decision tree structure. **(d)** Predicted samples using the hard decision tree. **(e)** Predicted samples using the soft decision tree.

questions denoted by $\{f_i(c)\}_{i=1}^{19}$. Additionally, the soft decision tree is trained by exploiting four distinct soft questions defined by

$$\forall i \in \{1,2,3,4\}\ \tilde{q}_i(c) = \exp\left\{\frac{-(c-0.25i+0.125)^2}{2\times(0.2)^2}\right\} \quad (25)$$

Figure 3b,c shows the hard and soft decision tree structures trained based on the maximum likelihood decision tree construction algorithms. As can be seen from the figures, the hard decision tree requires eight leaves to have an acceptable mean square error of 0.015, but the soft decision tree is able to accurately estimate the objective function with a small mean square error of 0.0002 using just six terminal nodes.

Figure 3d,e shows the approximated functions using hard and soft decision trees, respectively. As an obvious consequence of this simple experiment, the hard decision tree structures are not efficient to exploit the continuous attributes (contextual factors), and incorporating the soft decisions in their internal nodes significantly improves their predictive capabilities.

3.6 Defining soft questions

As it was mentioned earlier, in order to construct the soft decision tree structure, a set of basic contextual factors has to be extracted initially for all training and test datasets. Section 4.1.1 gives the details of the basic contextual factors employed in our experiments. These basic contextual factors have been denoted by c in this paper and can be grouped into two types of factors: categorical and numerical factors. 'Phoneme identity' is a sample of categorical factors, and 'Position of the current phoneme' is an example of the numerical factors. In fact, a numerical factor returns some ordered values, but a categorical factor provides some unordered symbols. For the categorical factors, we cannot define meaningful soft questions, and therefore, we have no choices but to exploit the conventional hard questions. However, for the numerical factors, it is possible to define a large number of soft questions. This subsection introduces the procedure of defining these soft questions in our experiments.

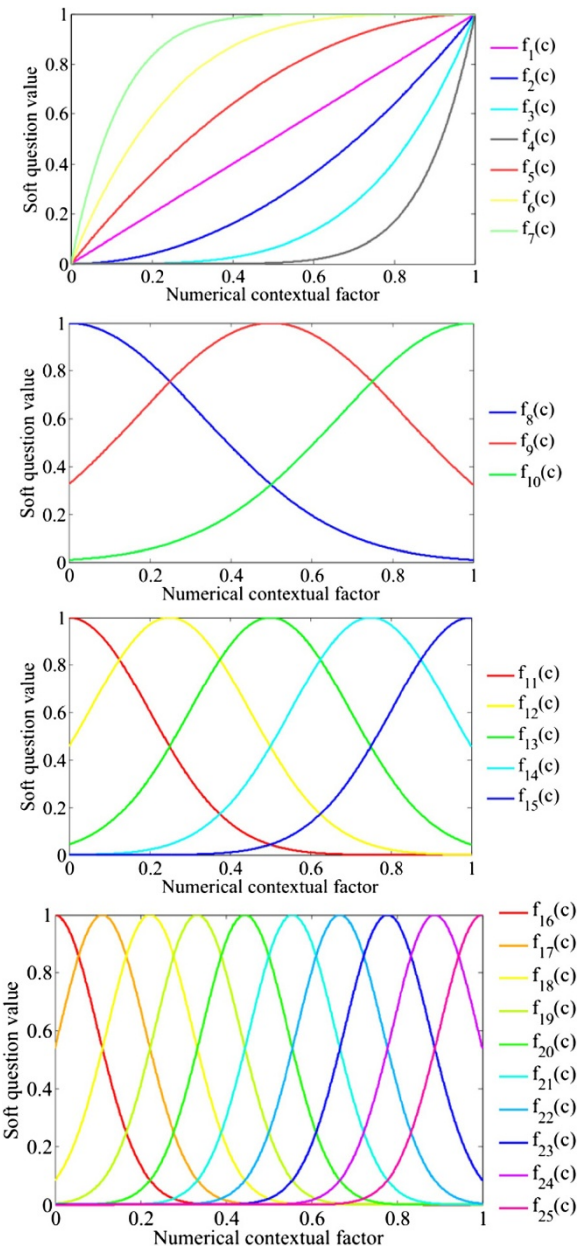

Figure 4 Soft questions defined for each numerical contextual factor in our experiments.

In this study, we first normalize all numerical contextual factors to range between 0 and 1, and then soft questions are obtained by applying a fixed set of candidate functions to the normalized contextual factors. Assume $\tilde{c}$ represents a normalized numerical contextual factor and $\tilde{f}_k(\tilde{c})$ is the kth soft question extracted for $\tilde{c}$. In this study, 25 soft questions have been defined for each numerical contextual factor. These soft questions are shown in Figure 4, and their mathematical expressions are given by Equation 26:

$$\begin{array}{lll}
\tilde{f}_1(\tilde{c})=\tilde{c}, & \tilde{f}_2(\tilde{c})=\tilde{c}^2, & \tilde{f}_3(\tilde{c})=\tilde{c}^4,\\
\tilde{f}_4(\tilde{c})=\tilde{c}^8, & \tilde{f}_5(\tilde{c})=1-(1-\tilde{c})^2, & \tilde{f}_6(\tilde{c})=1-(1-\tilde{c})^4\\
\tilde{f}_7(\tilde{c})=1-(1-\tilde{c})^8 & \tilde{f}_8(\tilde{c})=G_{0,\frac{1}{3}}(\tilde{c}), & \tilde{f}_9(\tilde{c})=G_{0.5,\frac{1}{3}}(\tilde{c}),\\
\tilde{f}_{10}(\tilde{c})=G_{1,\frac{1}{3}}(\tilde{c}), & \tilde{f}_{11}(\tilde{c})=G_{0,\ 0.2}(\tilde{c}), & \tilde{f}_{12}(\tilde{c})=G_{\frac{1}{4},0.2}(\tilde{c}),\\
\tilde{f}_{13}(\tilde{c})=G_{\frac{2}{4},0.2}(\tilde{c}), & \tilde{f}_{14}(\tilde{c})=G_{\frac{3}{4},0.2}(\tilde{c}), & \tilde{f}_{15}(\tilde{c})=G_{1,0.2}(\tilde{c}),\\
\tilde{f}_{16}(\tilde{c})=G_{0,0.2}(\tilde{c}), & \tilde{f}_{17}(\tilde{c})=G_{\frac{1}{9},0.2}(\tilde{c}), & \tilde{f}_{18}(\tilde{c})=G_{\frac{2}{9},0.1}(\tilde{c}),\\
\tilde{f}_{19}(\tilde{c})=G_{\frac{3}{9},0.1}(\tilde{c}), & \tilde{f}_{20}(\tilde{c})=G_{\frac{4}{9},0.2}(\tilde{c}), & \tilde{f}_{21}(\tilde{c})=G_{\frac{5}{9},0.1}(\tilde{c}),\\
\tilde{f}_{22}(\tilde{c})=G_{\frac{6}{9},0.1}(\tilde{c}), & \tilde{f}_{23}(\tilde{c})=G_{\frac{7}{9},0.1}(\tilde{c}), & \tilde{f}_{24}(\tilde{c})=G_{\frac{8}{9},0.1}(\tilde{c}),\\
\tilde{f}_{25}(\tilde{c})=G_{1,0.2}(\tilde{c}), & \text{where } G_{\mu,\sigma}(\tilde{c})= & \exp\left(-\frac{1}{2}\left(\frac{\tilde{c}-\mu}{\sigma}\right)^2\right).
\end{array} \tag{26}$$

In conclusion, all contextual factors were divided into two groups, namely, categorical and numerical. According to the above procedure, a set of soft questions were extracted for numerical factors, and a number of hard questions were obtained for categorical contextual factors. Thereafter, all of the extracted hard and soft questions were grouped together and competed against each other during the soft clustering procedure.

4 Experiments

This section aims to compare the performance of fundamental frequency modeling approaches based on the conventional hard decision tree and the proposed soft clustering method.

4.1 Experimental conditions

Before presenting the experimental results, this section describes the experimental conditions, including database characteristics and employed contextual factors, in detail.

An English speech database called Nick [69] consisting of approximately 2,500 utterances from a British male speaker was used in our experiments. This database is collected in Edinburgh University for the purpose of speech synthesis research. Sentences range in length from 3 to 36 words with an average length of 7.3 words per sentence. Also, the sentences cover most frequent English words, bi-phoneme combinations, and syllables. Totally, 2,944 different words are covered in the sentences.

Speech waveforms were sampled at 48 kHz, windowed by a 25-ms Blackman window with 5-ms shift. The speech analysis and synthesis conditions expressed in CSTR/EMIME HTS 2010 [69] were used in this experiment. In this platform, Bark-cepstrum was extracted from smooth STRAIGHT trajectories [6], since it outperforms predominant Mel-cepstrum coefficients. Also,

the widely used log-F0 and five aperiodicity sub-bands (0 to 1, 1 to 2, 2 to 4, 4 to 6, and 6 to 8 kHz) were replaced with pitch in Mel and auditory-scale motivated frequency bands for aperiodicity measure [69]. The analysis process generated 40 bark cepstrum coefficients, 1 Mel in pitch value, and 25 auditory-scale motivated aperiodicity frequency sub-bands for each frame of training signals. These parameters along with their delta and delta-delta derivatives formed five streams of our observation vectors.

For the baseline system, a five-state multi-stream left-to-right without skip path MSD-HSMM was trained. A conventional maximum likelihood-based decision tree construction algorithm was used to tie HMM states. In the conventional HMM-based speech synthesis framework, a unique tying structure (decision tree) is normally incorporated for both voicing probabilities and F0 output probabilities. As opposed to the conventional HMM-based synthesis system, the proposed method uses a soft decision tree structure for the output probability distribution and a hard decision tree for voicing probabilities; therefore, we cannot apply the same tying structure for both voicing and output probabilities in the proposed system. With a view to having a fair comparison, the baseline system was implemented with two different decision trees for F0 trajectories, one for the voicing labels and the other for the output probability distributions.

The same structure with just one different part was also implemented for the proposed synthesis system. In the proposed system, the soft decision tree structure is trained for F0 and its derivatives output probability distributions instead of the hard decision tree. All other decision trees, including the decision trees trained for state duration, Bark-cepstrum, aperiodicity, and voicing probability, are completely equal to the ones trained for the baseline system. Therefore, all parameters generated for them are equal to the parameters generated for the baseline system. It should be noted that both baseline and proposed synthesis systems employ the MDL criterion [66] to determine the size of all decision trees.

We considered four sets including 100, 200, 400, and 800 utterances for training, and 400 sentences that were not included in the training sets were used as a test data.

4.1.1 Employed contextual factors

Specific information about the contextual factors is presented in this subsection. Employed contextual factors can be categorized into five levels, including phonetic, syllable, word, phrase, and sentence levels. In each of these levels, all important features were considered.

- Phonetic-level factors
  - Phoneme identity before the preceding phoneme, preceding, current, succeeding phoneme, and phoneme identity after the next phoneme
  - Position of the current phoneme in the current syllable, word, phrase, and sentence
- Syllable-level factors
  - 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 factors
  - Part of speech (POS) tag of the preceding, current, and succeeding word.
  - 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 factors
  - 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 factors
  - Number of phonemes, syllables, words, and phrases in the current utterance
  - Type of the current sentence

4.2 Experimental results

Both objective and subjective tests are conducted to evaluate the proposed F0 modeling method. The results of these tests are given in the following subsections.

4.2.1 Objective evaluation

Figure 5 shows the learning curves obtained during building the hard and soft decision trees for 800 training utterances and 400 test sentences. Normalized log-likelihood measure, depicted in this figure, was computed through the following expression:

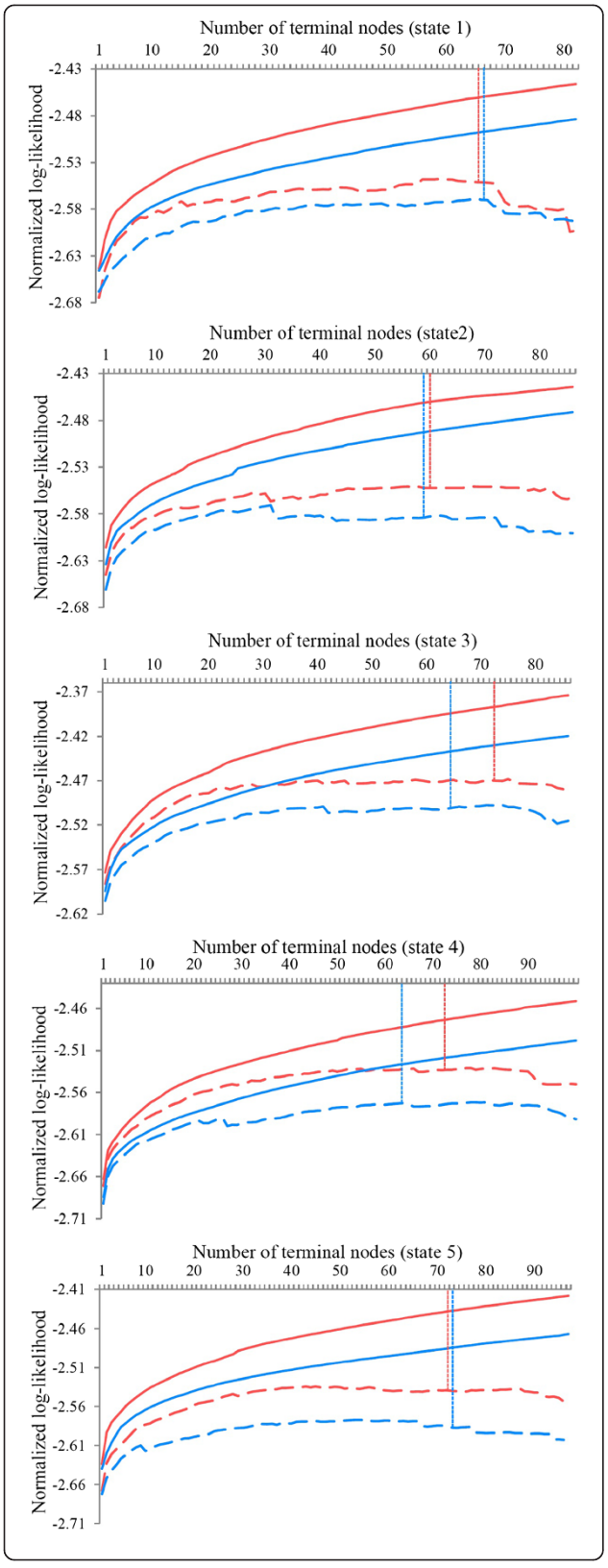

Figure 5 Normalized log-likelihood with respect to the number of leaves computed for each state of HMM. Blue and red curves are the learning curves of the hard and soft decision trees, respectively. In addition, solid curves illustrate the log-likelihood of the training set, and dashed curves are the log-likelihood computed for test data. MDL-based stop points are also shown through vertical dotted lines.

$$\mathcal{L} = \frac{1}{\sum_t \sum_l g_{tl}} \sum_t \sum_l g_{tl} \log\ b(o_{tl}|g_{tl}), \tag{27}$$

where the F0 and its derivatives are represented by o_{tl}, and their voicing labels are denoted by g_{tl}. t is the frame index and l represents the dynamic or static features ranging from 1 to 3. In this figure, the above measure is depicted for both test and train data. Red and blue curves are related to the proposed soft and the conventional hard decision tree structures, respectively. Solid curves are the normalized log-likelihood measure of the training sets, and the dashed curves represent the normalized log-likelihood measure computed for the test data. Also, the optimum number of terminal leaves calculated by the MDL principle is illustrated through vertical dotted lines. As it is realized from Figure 5, all red curves surpass their corresponding blue curves; therefore, the soft decision tree is able to provide superior log-likelihood measure with a smaller number of model parameters. All learning curves confirm the fact that the soft decision tree structure is able to provide better generalization in contrast to the canonical hard decision tree structure.

Another well-known objective measure, reported in this section, is the *root-mean-square error (RMSE)* between synthesized and natural log-F0 trajectories. In order to compute this measure, first, all test utterances were synthesized with natural voicing labels and natural durations (durations obtained through applying the Viterbi algorithm to natural acoustic trajectories). Thereafter, the RMSE measure is computed through the following expression:

$$\mathrm{RMSE} = \sqrt{\frac{1}{\sum_t g_t} \sum_t g_t \left(f_t^P - f_t^T\right)^2}, \tag{28}$$

where g_t, f_t^T, and f_t^P are voicing label, target log-F0 value, and predicted log-F0 value of the tth frame. This measure is computed for four training datasets including 100, 200, 400, and 800 training utterances. Figure 6 shows the calculated RMSE values in terms of cent. As it is realized from this figure, the log-F0 trajectories generated from the proposed approach is more similar to the natural log-F0 trajectories, and therefore, the proposed soft decision tree structure improves the performance of log-F0 modeling. However, by increasing the size of database,

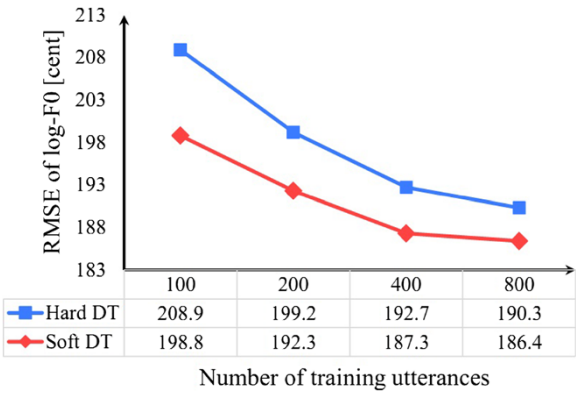

Figure 6 RMSE as an objective measure to compare log-F0 trajectories generated by hard and soft decision trees. Blue and red lines are the results of hard and soft decision trees, respectively.

the amount of this improvement is slightly reduced. Hence, it can be implied that the effect of applying soft clustering for small databases is relatively more than its effect on large databases.

4.2.2 Subjective evaluation

Two subjective tests have been selected in order to assess the effectiveness of the proposed system in comparison with the conventional synthesis system. The *comparative mean opinion score (CMOS)* test [7] with a 7-point scale, ranging from −3 to 3, and the paired comparison test [70] have been used to evaluate the subjective similarity of the synthesized and the natural utterances. Eighteen evaluators participated in our subjective evaluations, and each of them was asked to listen to 20 randomly chosen pairs of synthesized waveforms generated by two different synthesizers (i.e., the soft decision tree-based system and the conventional system).

In paired comparison tests, listeners are presented with a number of pairs of waveforms and they are asked to identify which one is more similar to its corresponding natural speech signal. If the two utterances sound equal, listeners are allowed to choose the equality option. The paired comparison test simply reports the percentage of comparisons that a certain synthesizer outperforms the other.

In CMOS tests, listeners not only select the better utterance, but also determine the difference level between two utterances. Four levels are normally defined for this purpose (namely, 0, 1, 2, and 3 which respectively have the meaning of about the same, slightly different, different, and much different). These difference levels are mainly useful in computing CMOS scores which have to be calculated in each comparison for each synthesizer separately. More precisely, a positive score equal to the difference level is computed for the winner of the comparison, and a negative score with equivalent absolute is assigned to the loser. Finally, the value of the CMOS score is obtained by taking an average over all scores.

The results of CMOS and paired comparison evaluations are respectively shown in Figures 7 and 8. Remarkably, the proposed soft context-clustered HMM is noticed to outperform the conventional hard decision tree structure for all training utterances. This result is completely in line with the conclusion of the objective assessments. For small datasets (i.e., 100 and 200 training utterances), more than 58% of the comparisons are in favor of the proposed method and the average CMOS score of the proposed system is more than one unit higher than the baseline system. These results show that the proposed soft decision tree structure is able to improve the F0 estimation accuracy of the baseline system significantly in small training datasets, and therefore, an important application of the proposed system is in low-resource languages when limited amount of data is available for training.

Another considerable conclusion that can be drawn from the results presented in this section is that by increasing the number of training utterances, the

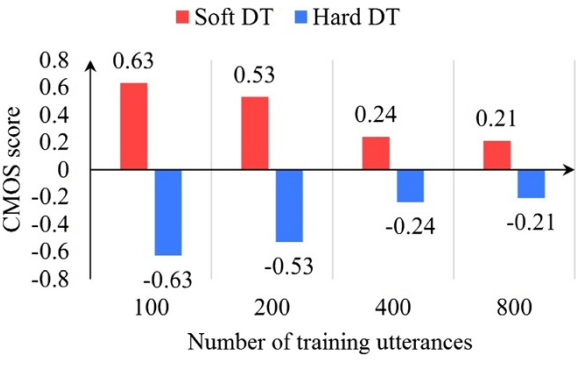

Figure 7 Subjective evaluation of soft context-clustered HMM through CMOS test. Blue and red bars are the CMOS scores of the conventional and the proposed systems.

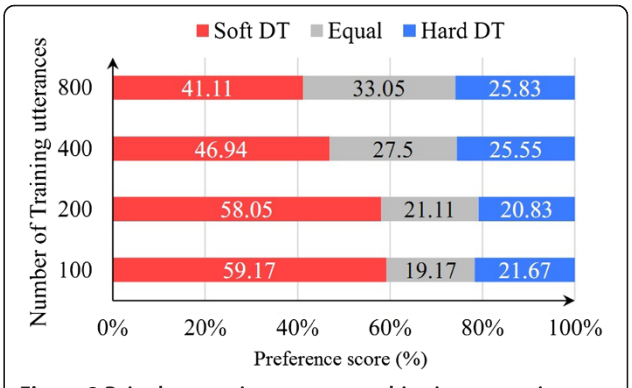

Figure 8 Paired comparison test as a subjective comparison between soft and hard decision tree structures. The scores of hard and soft decision trees are shown with blue and red colors.

improvement achieved through applying soft clustering is slightly reduced; thus, it is more efficient to employ the proposed structure in limited training datasets.

5 Conclusions

This paper addressed one of the most important shortcomings of hard decision tree-based context-dependent F0 modeling, namely, poor context generalization. In the hard decision tree structure, each acoustic feature vector is associated with modeling only one contextual cluster, and it is the main reason of poor generalization. In order to alleviate this problem, the capability of exploiting soft questions was added to the conventional decision tree architecture. The resulting structure, which is called soft decision tree, splits the contextual factor space into several soft clusters; therefore, each context is assigned to several leaves and it can provide superior generalization. In this paper, a maximum entropy model was used to drive the distribution expressed by the soft decision tree architecture. Relying on maximum entropy-based distribution, a speech synthesis system with all details was designed and implemented. Experimental results using both objective and subjective criteria showed that the proposed system outperforms the conventional hard decision tree-based system.

Endnote

[a]The unfortunate need for three separate streams only arises when using MSD output distributions to model F0: it is possible (at the onset or offset of voicing) for the dimensionality of the delta stream to be 0 in the same frame that the dimensionality of F0 is 1. That is, F0 exists, but its delta is undefined.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Computer Engineering, Sharif University of Technology, Azadi Avenue, Tehran 14588, Iran. [2]The Centre for Speech Technology Research, University of Edinburgh, Informatics Forum, 10 Crichton Street, Edinburgh EH8 9AB, UK.

Received: 29 August 2014 Accepted: 5 December 2014
Published: 9 January 2015

References

1. S King, An introduction to statistical parametric speech synthesis. Sadhana **36**(5), 837–852 (2011)
2. T Dutoi, *An Introduction to Text-to-Speech Synthesis, vol. 3* (Springer book (Kluwer Academic Publishers), The Netherlands, 1997)
3. H Zen, K Tokuda, AW Black, Statistical parametric speech synthesis. Speech Comm **51**(11), 1039–1064 (2009)
4. AW Black, H Zen, K Tokuda, *Statistical Parametric Speech Synthesis* (IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Honolulu, Hawaii, USA, 2007), vol 4, pp. IV-1229
5. T Yoshimura, K Tokuda, T Masuko, T Kobayashi, T Kitamura, *Mixed Excitation for HMM-Based Speech Synthesis* (European Conference on Speech Communication and Technology INTERSPEECH, Aalborg, Denmark, 2001). pp. 2263–2266
6. H Kawahara, I Masuda-Katsuse, A 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)
7. T Drugman, T Dutoit, The deterministic plus stochastic model of the residual signal and its applications. IEEE Transactions on Audio, Speech and Language Processing **20**(3), 968–981 (2012)
8. T Drugman, G Wilfart, T Dutoit, *A Deterministic Plus Stochastic Model of the Residual Signal for Improved Parametric Speech Synthesis* (INTERSPEECH, Brighton, United Kingdom, 2009). pp. 1779–1782
9. Y Stylianou, Applying the harmonic plus noise model in concatenative speech synthesis. IEEE Transactions on Speech and Audio Processing **9**(1), 21–29 (2001)
10. MY Liberman, KW Church, Text analysis and word pronunciation in text-to-speech synthesis. Advances in Speech Signal Processing, 791–831 (1992)
11. K Tokuda, T Yoshimura, T Masuko, T Kobayashi, T Kitamura, *Speech Parameter Generation Algorithms for HMM-Based Speech Synthesis* (IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Istanbul, 2000), pp. 1315–1318
12. T Toda, K Tokuda, Speech parameter generation algorithm considering global variance for HMM-based speech synthesis. IEICE - Transactions on Information and Systems **E90-D**(5), 816–824 (2007)
13. S Takamichi, T Toda, Y Shiga, S Sakti, G Neubig, S Nakamura, *Parameter Generation Methods With Rich Context Models for High-Quality and Flexible Text-to-Speech Synthesis* (IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, British Columbia, Canada, 2013)
14. M Shannon, W Byrne, *Fast, low-Artifact Speech Synthesis Considering Global Variance* (IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, British Columbia, Canada, 2013), pp. 7869–7873
15. AJ Hunt, AW Black, *Unit Selection in a Concatenative Speech Synthesis System Using a Large Speech Database* IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Atlanta, Georgia, USA, (373, 376, 1996)
16. J Yamagishi, T Kobayashi, Average-voice-based speech synthesis using HSMM-based speaker adaptation and adaptive training. IEICE - Transactions on Information and Systems **90**(2), 533–543 (2007)
17. 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. IEEE Transactions on Audio, Speech, and Language Processing **17**(6), 1208–1230 (2009)
18. 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)
19. 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 on Audio, Speech, and Language Processing **20**(6), 1713–1724 (2012)
20. YJ Wu, Y Nankaku, K Tokuda, *State Mapping Based Method for Cross-Lingual Speaker Adaptation in HMM-Based Speech Synthesis* (INTERSPEECH, Brighton, United Kingdom, 2009), pp. 528–531
21. H Liang, J Dines, L Saheer, *A Comparison of Supervised and Unsupervised Cross-Lingual Speaker Adaptation Approaches for HMM-Based Speech Synthesis* (IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), Dallas, Texas, USA, 2010). pp. 4598–4601
22. 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* (IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), Dallas, Texas, USA, 2010), pp. 4642–4645
23. J Yamagishi, Z Ling, S King, *Robustness of HMM-Based Speech Synthesis* (INTERSPEECH, Brisbane, Australia, 2008), pp. 581–584
24. R Karhila, U Remes, M Kurimo, *HMM-Based Speech Synthesis Adaptation Using Noisy Data: Analysis and Evaluation Methods* (IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, British Columbia, Canada, 2013), pp. 6930–6934
25. K Yanagisawa, J Latorre, V Wan, MJ Gales, S King, *Noise Robustness in HMM-TTS Speaker Adaptation* (8th ISCA Speech Synthesis Workshop, Barcelona, Spain, 2013), pp. 119–124
26. M Cernak, P Motlicek, PN Garner, *On the (un) Importance of the Contextual Factors in HMM-Based Speech Synthesis and Coding* (IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, British Columbia, Canada, 2013), pp. 8140–8143

27. T Yoshimura, K Tokuda, T Masuko, T Kobayashi, T Kitamura, *Simultaneous Modeling of Spectrum, Pitch and Duration in HMM-Based Speech Synthesis* (Proceedings of Eurospeech, Budapest, Hungary, 1999), pp. 2347–2350
28. T Yoshimura, K Tokuda, T Masuko, T Kobayashi, T Kitamura, *Duration Modeling in HMM-Based Speech Synthesis System* (Proceedings of ICSLP, Sydney. Australia, 1998), pp. 29–32
29. H Zen, T Nose, J Yamagishi, S Sako, T Masuko, A Black, T Keiichi, *The HMM-Based Speech Synthesis System (HTS) Version 2.0* (6th ISCA Workshop on Speech Synthesis (SSW), Bonn, Germany, 2007), pp. 294–299
30. K Tokuda, H Zen, AW Black, *An HMM-Based Speech Synthesis System Applied to English* (IEEE Workshop on Speech Synthesis, Scotland, 2002), pp. 227–230
31. H Zen, T Toda, M Nakamura, K Tokuda, Details of the Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005. IEICE Trans Inf Syst **90**(1), 325–333 (2007)
32. H Zen, T Toda, K Tokuda, The Nitech-NAIST HMM-based speech synthesis system for the Blizzard Challenge 2006. IEICE Trans Inf Syst **91**(6), 1764–1773 (2008)
33. K Tokuda, Y Nankaku, T Toda, H Zen, J Yamagishi, K Oura, Speech synthesis based on hidden Markov models. Proc IEEE **101**(5), 1234–1252 (2013)
34. JJ Odell, *The use of Context in Large Vocabulary Speech Recognition* (PhD dissertation, Cambridge University, 1995)
35. SJ Young, JJ Odell, PC Woodland, *Tree-Based State Tying for High Accuracy Acoustic Modeling* (Proceedings of the Workshop on Human Language Technology, Association for Computational Linguistics, Stroudsburg, PA, USA, 1994), pp. 307–312
36. K Tokuda, T Masuko, N Miyazaki, T Kobayashi, Multi-space probability distribution HMM. IEICE Trans Inf Syst **85**(3), 455–464 (2002)
37. H Zen, T Keiichi, T Masuko, T Kobayashi, T Kitamura, A hidden semi-Markov model-based speech synthesis system. IEICE Transactions on Information and Systems, E series **D-90**(5), 825–834 (2007)
38. 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 2013* (IEEE, 2013), pp. 7962–7966
39. H Lu, S King, O Watts, *Combining a Vector Space Representation of Linguistic Context With a Deep Neural Network for Text-to-Speech Synthesis* (8th ISCA Speech Synthesis Workshop, Barcelona, Spain, 2013), pp. 261–265
40. ZH Ling, L Deng, D Yu, *Modeling Spectral Envelopes Using Restricted Boltzmann Machines for Statistical Parametric Speech Synthesis* (IEEE Acoustics, Speech and Signal Processing (ICASSP), Vancouver, British Columbia, Canada, 2013), pp. 7825–7829
41. S Kang, X Qian, H Meng, *Multi-Distribution Deep Belief Network for Speech Synthesis* (IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, British Columbia, Canada, 2013), pp. 8012–8016
42. T Koriyama, T Nose, T Kobayashi, Statistical parametric speech synthesis based on Gaussian process regression. IEEE Journal of Selected Topics in Signal Processing **99**, 1–11 (2013)
43. K Hashimoto, Y Nankaku, K Tokuda, *A Bayesian Approach to Hidden Semi Markov Model Based Speech Synthesis* (Proceedings of Interspeech, Brighton, United Kingdom, 2009), pp. 1751–1754
44. S Khorram, H Sameti, F Bahmaninezhad, S King, T Drugman, Context-Dependent Acoustic Modeling Based on Hidden Maximum Entropy Model for Statistical Parametric Speech Synthesis. EURASIP Journal on Audio, Speech, and Music Processing **12**, 1–21 (2014)
45. Y Nankaku, K Nakamura, H Zen, K Tokuda, *Acoustic Modeling With Contextual Additive Structure for HMM-Based Speech Recognition* (IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Las Vegas, Nevada, USA, 2008), pp. 4469–4472
46. S Takaki, Y Nankaku, K Tokuda, *Spectral Modeling With Contextual Additive Structure for HMM-Based Speech Synthesis* (Proceedings of 7th ISCA Speech Synthesis Workshop, Kyoto, Japan, 2010), pp. 100–105
47. S Takaki, Y Nankaku, K Tokuda, *Contextual Partial Additive Structure for HMM-Based Speech Synthesis* (IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, British Columbia, Canada, 2013), pp. 7878–7882
48. S Takaki, Y Nankaku, K Tokuda, Contextual additive structure for HMM-based speech synthesis. IEEE J Selected Topics in Signal Processing **8**(2), 229–238 (2014)
49. H Zen, N Braunschweiler, *Context-Dependent Additive log F0 Model for HMM-Based Speech Synthesis* (INTERSPEECH, Brighton, United Kingdom, 2009), pp. 2091–2094
50. YJ Wu, F Soong, *Modeling pitch trajectory by hierarchical HMM with minimum generation error training* (IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 2012), pp. 4017–4020
51. S Sakai, Additive modeling of English F0 contour for speech synthesis. Proceedings of ICASSP **1**, 277–280 (2008)
52. Y Qian, H Liang, FK Soong, *Generating Natural F0 Trajectory With Additive Trees* (INTERSPEECH, Brisbane, Australia, 2008). pp. 2126–2129
53. K Yu, F Mairesse, S Young, *Word-Level Emphasis Modeling in HMM-Based Speech Synthesis* (IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Dallas, Texas, USA, 2010), pp. 4238–4241
54. 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)
55. MJ Gales, Cluster adaptive training of hidden Markov models. IEEE Transactions on Speech and Audio Processing **8**(4), 417–428 (2000)
56. C Olaru, L Wehenkel, A complete fuzzy decision tree technique. Fuzzy Set Syst **138**(2), 221–254 (2003)
57. Y Yuan, MJ Shaw, Induction of fuzzy decision trees. Fuzzy Set Syst **69**(2), 125–139 (1995)
58. L Rabiner, BH Juang, An introduction to hidden Markov models. IEEE ASSP Mag **3**(1), 4–16 (1986)
59. K Yu, S Young, Continuous F0 modeling for HMM based statistical parametric speech synthesis. IEEE Transactions on Audio, Speech, and Language Processing **19**(5), 1071–1079 (2011)
60. TK Moon, The expectation-maximization algorithm. IEEE Signal Process Mag **13**(6), 47–60 (1996)
61. *HMM-based speech synthesis system (HTS)* http://hts.sp.nitech.ac.jp/
62. H Zen, *Implementing an HSMM-Based Speech Synthesis System Using an Efficient Forward-Backward Algorithm, Nagoya Institute of Technology, Technical Report TR-SP-0001*, 2007
63. JD Ferguson, *Variable Duration Models for Speech* (Proceedings of the Symposium on the Application Hidden Markov Models to Text and Speech, USA, 1980), pp. 143–179
64. SE Levinson, Continuously variable duration hidden Markov models for automatic speech recognition. Computer Speech and Language **1**(1), 29–45 (1986)
65. SZ Yu, H Kobayashi, An efficient forward-backward algorithm for an explicit-duration hidden Markov model. IEEE Signal Processing Letters **10**(1), 11–14 (2003)
66. K Shinoda, T Watanabe, *Speaker Adaptation With Autonomous Model Complexity Control by MDL Principle* (Proceedings of ICASSP, Atlanta, Georgia, USA, 1996), pp. 717–720
67. AL Berger, VJD Pietra, SAD Pietra, A maximum entropy approach to natural language processing. Computational Linguistics **22**(1), 39–71 (1996)
68. A Ratnaparkhi, *A Maximum Entropy Model for Part-of-Speech Tagging* (Proceedings of the Conference on Empirical Methods in Natural Language Processin, PA, USA, 1996), pp. 133–142
69. J Yamagishi, O Watts, *The CSTR/EMIME HTS System for Blizzard Challenge* (Proceedings of Blizzard Challenge 2010, Japan, 2010), pp. 1–6
70. J Yamagishi, *Average-Voice-Based Speech Synthesis* (PhD thesis, Tokyo Institute of Technology, 2006)

doi:10.1186/1687-6180-2015-2
Cite this article as: Khorram *et al.*: **Soft context clustering for F0 modeling in HMM-based speech synthesis.** *EURASIP Journal on Advances in Signal Processing* 2015 **2015**:2.