Computational Linguistics and Chinese Language Processing
Vol. 15, No. 1, March 2010, pp. 61-76

Cross-Validation and Minimum Generation Error based Decision Tree Pruning for HMM-based Speech Synthesis

Heng Lu*, Zhen-Hua Ling*, Li-Rong Dai*, and Ren-Hua Wang*

Abstract

This paper presents a decision tree pruning method for the model clustering of HMM-based parametric speech synthesis by cross-validation (CV) under the minimum generation error (MGE) criterion. Decision-tree-based model clustering is an important component in the training process of an HMM based speech synthesis system. Conventionally, the maximum likelihood (ML) criterion is employed to choose the optimal contextual question from the question set for each tree node split and the minimum description length (MDL) principle is introduced as the stopping criterion to prevent building overly large tree models. Nevertheless, the MDL criterion is derived based on an asymptotic assumption and is problematic in theory when the size of the training data set is not large enough. Besides, inconsistency exists between the MDL criterion and the aim of speech synthesis. Therefore, a minimum cross generation error (MCGE) based decision tree pruning method for HMM-based speech synthesis is proposed in this paper. The initial decision tree is trained by MDL clustering with a factor estimated using the MCGE criterion by cross-validation. Then the decision tree size is tuned by backing-off or splitting each leaf node iteratively to minimize a cross generation error, which is defined to present the sum of generation errors calculated for all training sentences using cross-validation. Objective and subjective evaluation results show that the proposed method outperforms the conventional MDL-based model clustering method significantly.

Keywords: Speech Synthesis, Hidden Markov Model, Decision Tree Pruning, Cross-validation, Minimum Generation Error.

* University of Science and Technology of China, No. 96, Jinzhai Road, Hefei, Anhui, China
Tel: (+86) 13721053256; fax: (+86) 551 5331801.
E-mail: luhenglh@mail.ustc.edu.cn; zhling@ustc.edu; lrdai@ustc.edu.cn; rhw@ustc.edu.cn
The author for correspondence is Heng Lu.

1. Introduction

Currently, there are two main speech synthesis methods. One is unit-selection speech synthesis (Hunt & Black, 1996) (Ling & Wang, 2007) and the other is the hidden Markov model (HMM) based parametric speech synthesis (Black, Zen, & Tokuda, 2007). The unit-selection approach concatenates the natural speech segments selected from a recorded database to produce synthetic speech. It can generate highly natural speech often, but its performance may degrade severely when the contexts for synthesis are not included in the database. In HMM-based parametric speech synthesis, speech waveforms are parameterized and modeled by HMMs in model training (Yoshimura, Tokuda, Masuko, Kobayashi, & Kitamura, 1999). During synthesis, speech parameters are generated from the trained models (Tokuda, Yoshimura, Masuko, Kobayashi, & Kitamura, 2000) and sent to a parametric synthesizer to reconstruct speech waveforms. Although the quality of synthetic speech still needs improvement, HMM-based parametric synthesis has several important advantages, including high flexibility of the statistical models, a comparatively small database necessary for system construction and robust performance of the synthetic speech -- it never makes the serious errors that unit-selection speech synthesis may make sometimes.

In HMM-based parametric speech synthesis, binary decision tree based context-dependent model clustering is a necessary step in dealing with data-sparsity problems and predicting model parameters for the contextual features of synthetic speech that do not occur in the training set. In the conventional model clustering process, the maximum likelihood (ML) criterion is utilized to choose the optimal question from the question set for each tree node split and the minimum description length (MDL) criterion (Shinoda & Watanabe, 2000) is used as the stopping criterion to control the size of trained decision trees, which affects the performance of synthetic speech significantly, *e.g.*, a large decision tree may alleviate the over-smoothing effects in generated speech parameters but may also lead to over-fitting problems. Nevertheless, the MDL criterion is derived based on an asymptotic assumption and the assumption that fails when there is not enough training data (Rissanen, 1980). Therefore, it may not work successfully in HMM-based speech synthesis, where the amount of training data is much smaller than that in speech recognition.

Some research work has been done to improve the MDL criterion for the decision tree construction of HMM-based speech synthesis. A decision tree backing-off method was proposed in (Kataoka, Mizutani, Tokuda & Kitamura, 2004). In this method, a decision tree was first built using ML criterion without pruning. During synthesis, the tree nodes that generated the observations with maximum likelihood were chosen by a process of backing-off from the leaf node that was decided by the contextual information of each state for synthesis to the root node. Nevertheless, there still exist two issues in this method. One is the one-dimensional optimization algorithm adopted in (Kataoka, Mizutani, Tokuda, & Kitamura,

2004) to reduce the computational complexity, which means the decision tree backing-off is conducted simultaneously for all states instead of processing each state separately. The other is the inconsistency between the ML criterion and the aim of speech synthesis, which is to generate speech (acoustic parameters) as close to natural speech as possible. The minimum generation error (MGE) criterion has been proposed to solve the second issue. It optimized the model parameters by minimizing the distortion between the generated speech parameters and the natural ones for the sentences in the training set. The MGE criterion has been applied not only to the clustered model training (Wu & Wang, 2006b) but also to the decision tree based model clustering of context-dependent models (Wu, Guo & Wang, 2006) and positive results have been achieved in improving the naturalness of synthetic speech. In (Wu, Guo & Wang, 2006), MGE was adopted to replace the ML criterion to select the optimal question at each tree node split. Since increasing the size of the decision tree always leads to the reduction of the generation error on the training set, MGE cannot be used directly as a stopping criterion in decision tree building. Thus, the size of the decision tree trained in (Wu, Guo & Wang, 2006) was tuned manually to compare the results with the MDL clustering that had almost equivalent numbers of leaf nodes.

On the other hand, cross-validation (CV) is a well-known technique to deal with the over-training and under-training problems without requiring extra development data. It estimates the accuracy of performance of a predictive model by partitioning the data set into complementary subsets and uses different subsets for training and validation (Bishop. 2006). In (Hashimoto, Zen, Nankaku, Masuko & Tokuda, 2009), a CV based method of setting hyper-parameters for HMM-based speech synthesis under the Bayesian criterion was proposed and positive results were reported.

In this paper, we integrate the minimum “cross” generation error criterion to optimize the size of the model clustering decision tree automatically for HMM-based speech synthesis. Different from (Wu, Guo & Wang, 2006), the ML criterion is still adopted to select the optimal question at each tree node split. A “cross” generation error is defined to calculate the sum of generation errors for all training sentences by cross-validation using the models clustered with a given decision tree. The size of the decision tree is optimized to minimize the cross generation error in two steps. First, an initial decision tree is obtained through model clustering with the MDL factor tuned with MCGE criterion. Then, the decision tree is finely modified by backing-off or splitting each leaf node iteratively to minimize the cross generation error. Objective and subjective evaluation results show that this proposed method outperforms the conventional MDL based HMM model clustering method significantly.

This paper is organized as follows: Section 2 describes the HMM-based speech synthesis method with conventional MDL clustering. In Section 3, the proposed MCGE based decision tree pruning method is introduced. Objective and subjective experimental results are discussed

in Section 4. Finally, conclusions are given in Section 5.

2. HMM-based Parametric Speech Synthesis

2.1 The Framework of HMM-based Speech Synthesis

As shown in Figure 1, a typical HMM-based parametric speech synthesis system consists of two parts: the model training part and the speech synthesis part. In the model training part, spectrum, F0 and state duration are modeled simultaneously in a unified HMM framework. For each HMM state, the spectral features are modeled by a continuous probability distribution and F0 features are modeled using a multi-space probability distribution (MSD) (Tokuda, Masuko, Miyazaki & Kobayashi, 1999). In the synthesis step, speech parameters are generated from the trained models using maximum likelihood parameter generation (MLPG) algorithm (Tokuda, Yoshimura, Masuko, Kobayashi & Kitamura, 2000) and a parametric synthesizer is employed to reconstruct speech waveforms from the generated parameters.

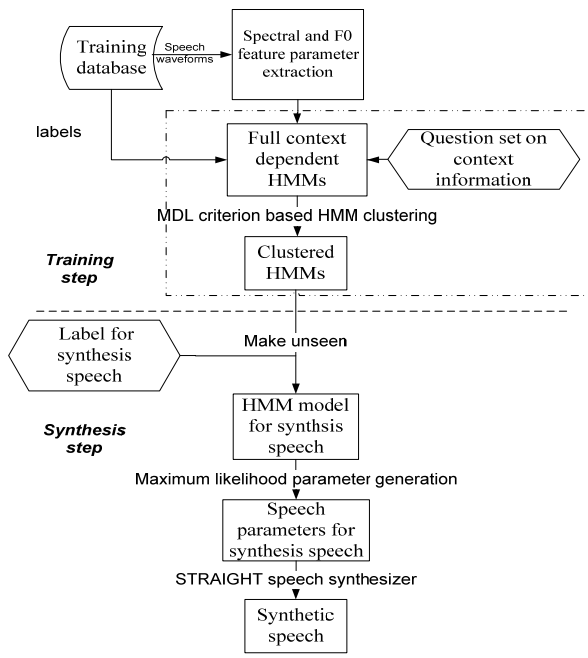

Figure 1. Flowchart of a conventional HMM-based parametric speech synthesis system.

2.2 MDL-based Model Clustering

In the training stage, decision-tree-based model clustering is conducted after training for full context-dependent HMMs to avoid data-sparsity problems and to predict model parameters for the context features that do not occur in the training set. A question set containing

language-dependent contextual questions is used. In the top-down decision tree building process, the ML criterion is commonly adopted to choose the optimal question and leaf node for splitting that lead to the greatest likelihood of growth. Further, the MDL principle is employed as a stopping criterion for decision tree pruning (Shinoda & Watanabe, 2000). The description length (DL) is defined as

$$I(\lambda) \equiv -\log P(\boldsymbol{o} \mid \lambda) + \frac{1}{2} D(\lambda) \log N + C \tag{1}$$

where λ denotes the clustered models; $\boldsymbol{o} = [\boldsymbol{o}_1^{\mathrm{T}}, \boldsymbol{o}_2^{\mathrm{T}}, ..., \boldsymbol{o}_N^{\mathrm{T}}]^{\mathrm{T}}$ is the training feature sequence, $(\cdot)^{\mathrm{T}}$ means the matrix transpose and N is the total frames of training data; $\log P(\boldsymbol{o} \mid \lambda)$ is the log likelihood function of λ on the training set; $D(\lambda)$ is the dimensionality of the model parameters; and C is a constant. The decision tree stops growth if the optimal leaf node splitting determined by the ML criterion can no longer reduce the DL.

If a single-Gaussian distribution with diagonal covariance matrix is used as the output probability distribution function (PDF) of each HMM state, Eq. (1) can be calculated as Equation (2) in (Shinoda & Watanabe, 2000)

$$I(\lambda) = \sum_{m=1}^{M} \frac{1}{2} \Gamma_m (E + E \log(2\pi) + \log |\boldsymbol{\Sigma}_m|) + EM \log N + C \tag{2}$$

where M is the leaf node number of the model clustering decision tree; Γ_m is the sum of state occupation probabilities for all frames in the training set belonging to the states that share the PDF of node m; E is the dimensionality of feature vectors; $\boldsymbol{\Sigma}_m$ is the covariance matrix of the Gaussian distribution function at node m.

Assume leaf node S with a contextual question is chosen among the M leaf nodes by ML criterion and further split into two child nodes SY and SN. Thus, the DL of the updated model λ' becomes

$$\begin{aligned} I(\lambda') = & \sum_{m=1, m \neq S}^{M} \frac{1}{2} \Gamma_m (E + E \log(2\pi) + \log |\boldsymbol{\Sigma}_m|) \\ & + \frac{1}{2} \Gamma_{SY} (E + E \log(2\pi) + \log |\boldsymbol{\Sigma}_{SY}|) \\ & + \frac{1}{2} \Gamma_{SN} (E + E \log(2\pi) + \log |\boldsymbol{\Sigma}_{SN}|) + E(M+1) \log N + C. \end{aligned} \tag{3}$$

The change of DL after the tree node splitting is

$$\Delta I = I(\lambda') - I(\lambda) = \frac{1}{2} \Gamma_{SY} \log |\boldsymbol{\Sigma}_{SY}| + \frac{1}{2} \Gamma_{SN} \log |\boldsymbol{\Sigma}_{SN}| - \frac{1}{2} \Gamma_S \log |\boldsymbol{\Sigma}_S| + E \log N. \tag{4}$$

The tree growth stops if $\Delta I > 0$. Thus, the stop condition of MDL-based decision tree building is

$$\frac{1}{2} \Gamma_S \log |\boldsymbol{\Sigma}_S| - \frac{1}{2} \Gamma_{SY} \log |\boldsymbol{\Sigma}_{SY}| - \frac{1}{2} \Gamma_{SN} \log |\boldsymbol{\Sigma}_{SN}| < E \log N. \tag{5}$$

The left side of Equation (5) presents the increase of log likelihood after the splitting. Therefore, the MDL criterion can be explained as introducing a threshold $E \log N$ into the ML-based decision tree construction. In practical system construction, an MDL factor $\alpha > 0$ is used to tune the threshold and control the size of the trained decision tree. Thus, Equation (5) can be rewritten as

$$\frac{1}{2}\Gamma_S \log|\boldsymbol{\Sigma}_S| - \frac{1}{2}\Gamma_{SY} \log|\boldsymbol{\Sigma}_{SY}| - \frac{1}{2}\Gamma_{SN} \log|\boldsymbol{\Sigma}_{SN}| < \alpha E \log N. \tag{6}$$

Small α would lead to a large decision tree.

Besides MDL, the node size is also used as a complementary stop condition in practical system construction. It requires each leaf node to contain at least β samples otherwise the tree growth stops. Therefore, the pruning of the ML-trained model clustering decision tree is determined by a pair of parameters $\{\alpha, \beta\}$ with a default value of $\{1.0,15\}$ in our baseline system.

3. Minimum Cross Generation Error based Decision Tree Pruning

3.1 Cross Generation Error

In order to introduce MGE criterion into the pruning of model clustering decision tree, Cross Generation Error (CGE) is calculated on the training set by cross-validation. Assume the training database is composed of L sentences. To do cross-validation, we first divide the database into K subsets, $\{\boldsymbol{S}_1, \boldsymbol{S}_2, ..., \boldsymbol{S}_K\}$ and

$$\boldsymbol{S}_k = \{\boldsymbol{C}_{k,1}, \boldsymbol{C}_{k,2}, ..., \boldsymbol{C}_{k,L_k}\}, k = 1, 2, ..., K \tag{7}$$

where $\boldsymbol{C}_{k,l} = [\boldsymbol{c}_{k,l,1}^{\mathrm{T}}, \boldsymbol{c}_{k,l,2}^{\mathrm{T}}, ..., \boldsymbol{c}_{k,l,T}^{\mathrm{T}}]^{\mathrm{T}}$ denotes the speech parameter sequence of the *l*-th sentence in the *k*-th subset, $\boldsymbol{c}_{k,l,t}$ is feature vector of the *t*-th frame in $\boldsymbol{C}_{k,l}$ and T is the frame number of $\boldsymbol{C}_{k,l}$; L_k is the number of sentences in subset *k* and $\sum_{k=1}^{K} L_k = L$. The phonetic balance needs to be considered when partitioning the database and the subsets should be divided as evenly as possible. When a model clustering decision tree *TR* is given, the "cross" generation error is calculated as

$$\mathcal{D}(TR) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{L_k}\sum_{l=1}^{L_k}\sum_{t=1}^{T_{k,l}} d(\boldsymbol{c}_{k,l,t}, \boldsymbol{c}'_{k,l,t}(\lambda_k(TR))) \tag{8}$$

where $\lambda_k(TR)$ represents the model estimated using the decision tree *TR* and the training subsets $\overline{\boldsymbol{S}}_k = \{\boldsymbol{S}_j\}_{j=1,...,K, j\neq k}$; $\boldsymbol{c}'_{k,l,t}(\lambda)$ denotes the generated parameter vector of frame *t* for the *l*-th sentence in subset *k* using model λ; $d(\boldsymbol{c}, \boldsymbol{c}')$ is an objective distortion function to calculate the generation error between the natural and generated speech parameters and a Euclidean distance measure is adopted here. The calculation process of the cross generation error is illustrated in Fig. 2.

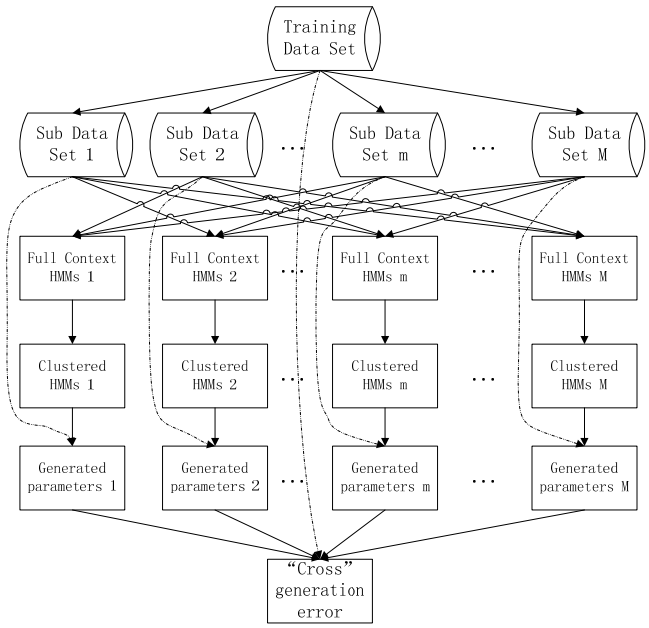

Figure 2. The calculation process of cross generation error.

3.2 Decision Tree Initialization

The pruning of the decision tree by CV and MGE is carried out in two steps. First we tune the MDL factor in Eq. (6) and the threshold in the node size stop condition discussed in Section 2.2 to generate an initial decision tree with a minimum cross generation error. Then the effect of each single tree leaf node on the cross generation error is inspected separately for further decision tree leaf backing-off or splitting. The decision tree initialization process is introduced in this section.

As shown in Equation (6), a small α would decrease the threshold in the stop condition of the MDL criterion and lead to a large decision tree. On the other hand, reducing the threshold β in the stop condition of the node size would also increase the size of the decision tree. A set of threshold parameter pairs $\{\alpha,\beta\}$ is designed in accordance with our speech synthesis system construction experience. For each pair of $\{\alpha,\beta\}$, a decision tree is trained via the method discussed in Section 2.2 and the cross generation error is calculated. We tune α first and keep β equal to its default value. When reducing α can no longer increase the size of the decision tree, we keep α constant and reduce β further. By such tuning, we are able to find a pair of $\{\alpha,\beta\}$ that leads to the smallest cross generation error. When the optimum pair of $\{\alpha,\beta\}$ is obtained, they are applied to conduct the model clustering using all of the training data and to generate the initial decision tree TR_0 for further optimization.

3.3 Cross Generation Error based Tree Pruning

Given an initial decision tree TR_0 by Section 3.2, the effect of every single leaf node on the cross generation error is inspected for further tree node back-off or splitting. Here, we define the cross generation error of tree node *m* as

$$\mathcal{D}_m(TR) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{L_k}\sum_{l=1}^{L_k}\sum_{t=1}^{T_{k,l}}\gamma_m(t)d(\boldsymbol{c}_{k,l,t},\boldsymbol{c}'_{k,l,t}(\lambda_k(TR))) \tag{9}$$

where $\gamma_m(t)$ denotes the state occupancy probability of frame *t* in the *l*-th sentence of subset *k* belonging to the node *m*. By comparing the sum of the cross generation error of each tree leaf node and its brother node with the cross generation error of their father node, it can decide whether we should back-off the leaf nodes to reduce the cross generation error or not. In the same way, we can decide whether the decision tree leaf should be split further. Backing-off or splitting continues for each decision tree leaf until no tree leaf can be backed-off or split. The optimization process for the decision tree backing-off and splitting is conducted iteratively and is described in detail as follows.

- ***Step 0.*** Given the divided training subsets $\{\boldsymbol{S}_1, \boldsymbol{S}_2, ..., \boldsymbol{S}_K\}$ for cross-validation, the initial decision tree TR_0 is backed-off to get TR_1 to guarantee that each leaf node should contain at least one frame of sample from every $\overline{\boldsymbol{S}}_k$.
- ***Step 1.*** A group of clustered models $\{\lambda_k(TR_1)\}_{k=1,...,K}$ is estimated. Set $i=1$.
- ***Step 2.*** Back-off all the leaf nodes in TR_i to their father nodes by one level and attain TR_i'. Assume that leaf node *m* in TR_i' is the father node of node *ml* and *mr* in TR_i. If $\mathcal{D}_m(TR_i') < \mathcal{D}_{ml}(TR_i) + \mathcal{D}_{mr}(TR_i)$, we merge node *ml* and *mr* in TR_i into their father node. Otherwise, these two leaf nodes are reserved. This process is carried out for all leaf nodes in TR_i' and a new tree TR_{i+1} after necessary backing-off. Then set $i = i+1$. The flowchart of this backing-off process is shown in Fig. 3.
- ***Step 3***. Step 2 is repeated until the number of merged leaf nodes per one time back-off is smaller than a given threshold τ.
- ***Step 4.*** Splitting is conducted in a similar way after the backing-off process is finished.

Following these steps, decision tree TR_0 is finely tuned for every leaf, reducing the cross generation error on the training set.

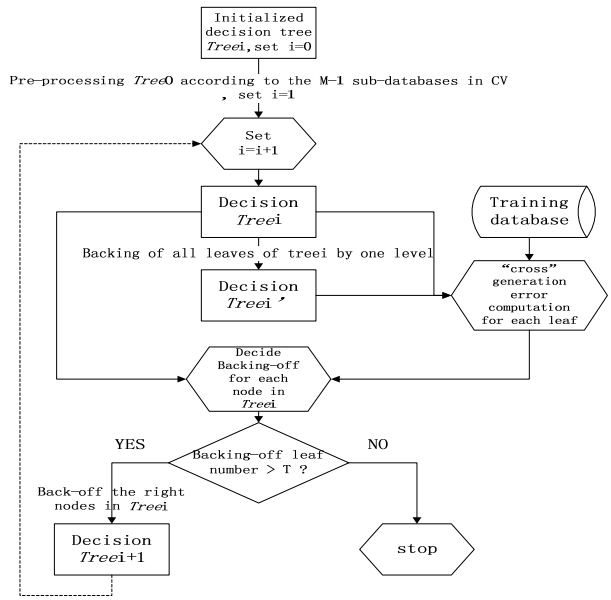

Figure 3. Flowchart for one decision tree back-off process.

4. Experiments

4.1 Experimental Conditions

In the experiment, we used a female phonetic balanced Mandarin database containing 1,000 sentences as the training database. The sample rate for the speech waves in the training database was 16kHz. 40 dimensional LSPs were extracted as the spectral features with 5ms frame shift. Five state context-dependent HMMs were used in the model training. Our experiments only focused on the decision-tree-based model clustering for spectral features. The context-dependent F0 and duration models were clustered in the conventional way.

A question set describing the contextual features for Mandarin Chinese was designed to conduct the decision tree splitting. The context features include:

- Left phone : phone before the current phone
- Current phone : the focused phone
- Right phone : phone after the current phone
- Left tone : tone of the syllable before the current syllable
- Current tone : the tone of the current syllable
- Right tone : tone of the syllable after the current syllable
- Part-of-speech : nature of the current word

- Relative positions of the current syllable, word, phrase, sentence, and sentence group
- Absolute positions from head and tail of the current syllable, word, phrase, sentence, and sentence group

4.2 Experiments on Decision Tree Initialization

4.2.1 Objective Evaluation

The training database was divided into ten subsets in our experiments. Following the method described in Section 3.2, a group of threshold parameter pairs $\{\alpha, \beta\}$ were designed as shown in Table 1. As the MDL factor α is the main factor that affects the size of the decision tree, we did not modify β until reducing α to where it could no longer enlarge the size of the decision tree. The System ID, the corresponding threshold parameter pairs $\{\alpha, \beta\}$, size of the decision tree, and the cross generation error calculated by LSP distortion introduced in Section 3.1 are shown Table 1.

Table 1. Scale of the decision tree and the objective LSF "cross" generation error for each system.

System ID	*Sys-A*	*Sys-B*	*Sys-C*	*Sys-D*	*Sys-E*	*Sys-F*	*Sys-G*	*Sys-H*	*Sys-I*
$\{\alpha, \beta\}$	{0.01,1}	{0.01,5}	{0.01,10}	{0.01,15}	{0.1,15}	{0.5,15}	{1,15}	{2,15}	{10,15}
Number of all leaf nodes	52882	36706	21211	14683	14654	8909	3946	1886	470
LSF distortion	0.02576	0.02498	0.02442	0.02421	0.02421	0.02428	0.02470	0.02553	0.02869

From Table 1, we can see that parameter set {0.01,15} (*Sys-D*) and {0.1,15} (*Sys-E*) lead to the smallest cross generation error. The baseline system is *Sys-G* with $\{\alpha, \beta\}$ in default settings.

4.2.2 Subjective Evaluation

A subjective listening test was also conducted for the above systems. As the trained decision trees of *Sys-D* and *Sys-E* were very close, *Sys-E* was omitted in the following subjective evaluation. Sixteen out-of-training-set test sentences were synthesized by the remaining eight systems. Five native Mandarin Chinese speakers were asked to give a score from 1 (very unnatural) to 5 (very natural) on the 128 synthetic sentences. The mean opinion scores (MOS) of all systems are shown in Fig. 4. From these results, we can see that the subjective scores match the objective cross generation error very well, where a smaller cross generation error corresponds to a higher MOS. *Sys-D* is the best system in the subjective evaluation and outperforms the baseline system (Sys-G). This proves the effectiveness of the proposed decision tree initialization method and the minimum cross generation error criterion. From

Figure 4 and Table 1, we also find that the LSF distortion of *Sys-A* and *Sys-B* is larger than *Sys-G*, but with a higher MOS score. This is reasonable because with a much smaller decision tree like in system *Sys-G*, the acoustic model would be too "average", making the synthesis speech "blurring". Nevertheless, large decision trees like *Sys-A* and *Sys-B* cause an over-training problem, where voice quality is not impacted much, but synthesized speech may not be stable.

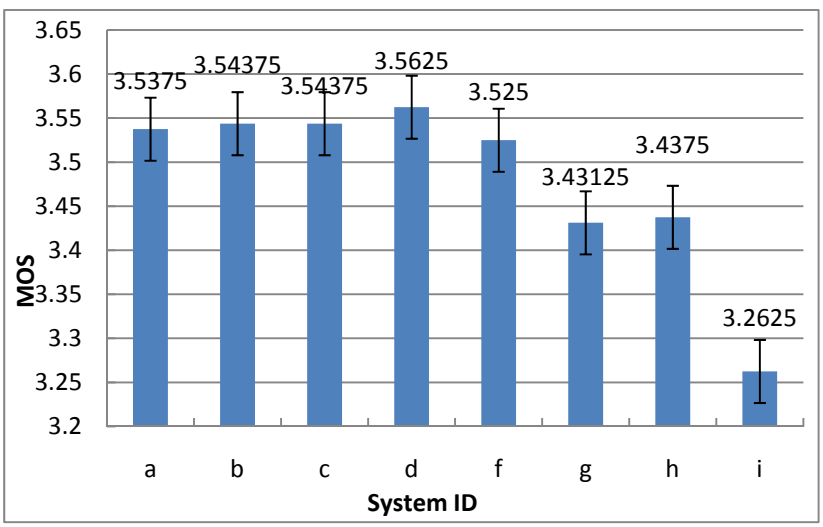

Figure 4. MOS of different systems for decision tree initialization.

4.3 Experiments on Decision Tree Pruning

4.3.1 Objective Evaluation

Using the threshold parameter pair {0.01,15} of *Sys-D*, the initial decision tree TR_0 was built by conducting MDL-based HMM clustering using this parameter set on the whole training database. Then further tree node backing-off and splitting introduced in Section 3.2 were conducted iteratively on the basis of TR_0. Here in the calculation of cross generation error, the same decision tree TR_0, other than the optimal $\{\alpha, \beta\}$, is utilized to conduct the model estimation of $\lambda_k(TR)$. The Euclidean LSP distance measure was used to compute the distortion between the generation and natural parameters. Figure 5 and Figure 6 describe the change in the cross generation error and the total number of the decision tree nodes in the iterative backing-off or splitting process. We can see that the cross generation error in Fig. 5 decreases consistently. Figure 6 shows that the backing-off was conducted for 9 iterations until no tree leaf could be backed-off and that node splitting was conducted for 2 iterations.

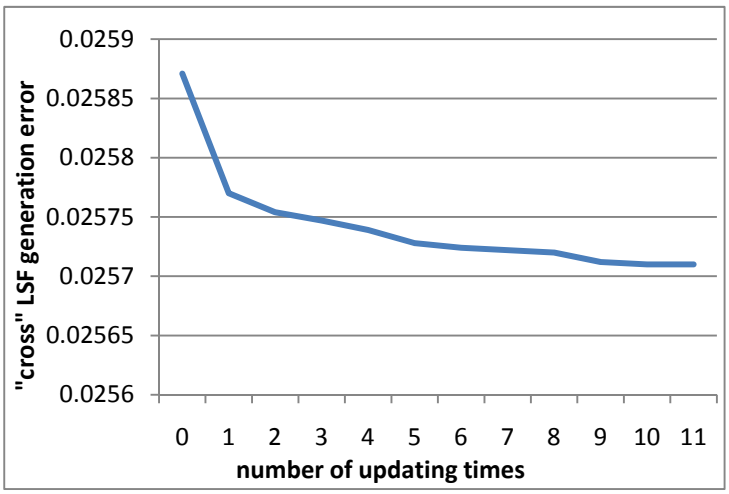

Figure 5. The "cross" generation error curve using Euclidean LSP distortion according to the decision tree pruning times. Decision tree backing-off is conducted 9 times until no leave can be combined. Then splitting for tree leaves is conducted for 2 times.

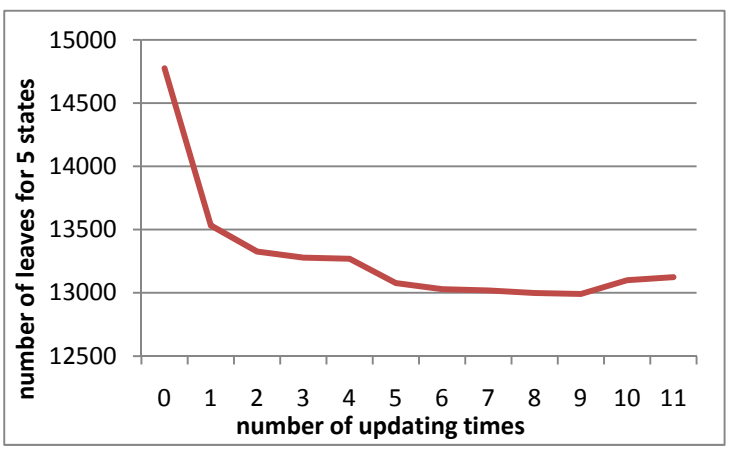

Figure 6. The scale of the decision tree according to the decision tree pruning times. Decision tree backing-off is conducted 9 times until no leave can be combined. Then splitting for tree leaves is conducted for 2 times.

Comparing Figure 5 and Table 1, one may find that the average "cross" generation error in the decision tree leaf backing-off and splitting process is larger than the average "cross" generation error in the MDL threshold parameter set optimizing process. This is normal because in the MDL threshold parameter optimization process, we employ the same MDL threshold parameter set for each $K-1$ sub-databases HMM clustering in the CV process. In

the backing-off and splitting process, however, the same decision tree except for the MDL parameters is employed for HMM clustering in the CV. A different decision tree for different divisions in CV leads to a smaller "cross" generation error.

4.3.2 Subjective Evaluation

A subjective listening test was conducted for the three systems: the baseline system (*Sys-G*), the system with tuned $\{\alpha, \beta\}$ (*Sys-D*), and the system with further backing-off and splitting based on *Sys-D*. Sixteen sentences were synthesized by each of the systems and five native speakers were asked to choose the best sentence from the randomly ordered three sentences by three systems. The results are listed in Fig. 7, where the preference ratios for the three systems are 21.6%, 36.7% and 41.7% respectively.

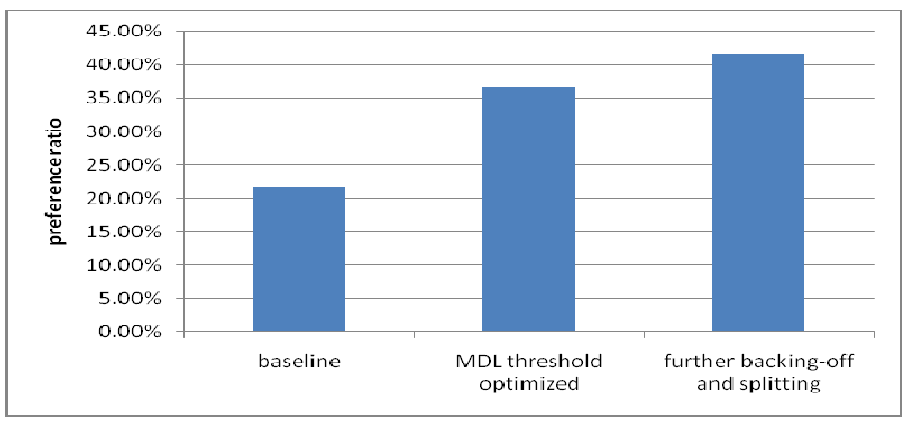

Figure 7. Preference ratio for the (1) baseline system, (2) MDL parameter optimized speech synthesis system and (3) further backing-off and splitting system.

From Figure 7, one can conclude that the MDL threshold parameter optimized speech synthesis system and further backing-off and splitting system both out-perform the baseline system. The proposed method for initialization of the decision tree and the further pruning method are both effective.

4.4 Discussion

The subjective MOS test and the objective LSP distortion prove the effectiveness of our two step decision tree pruning method. Compared with generating decision tree from the top or backing-off from the bottom, our two-steps decision tree pruning method, pruning the decision tree from the middle of the decision tree avoids many sub-optimums. If we start to prune from a huge decision tree which is split without any constraint using the method described in Section 3.3, we cannot guarantee that once the cross generation error by the father node is larger than the current tree leaves, the cross generation error by the grandfather level is also

larger than the tree leaves. It could be smaller! Also pruning from the middle of the decision tree avoids a huge computational cost.

Theoretically, in order to get the decision tree that leads to the minimum cross generation error, one should use the minimum cross generation error criterion to choose the best question from the question set, and use the best question to conduct the splitting of every decision tree node. This means speech parameters for the synthesized speech should be generated and the cross generation error for the whole decision tree should be calculated for all the questions in the question set for each tree leaf. This will lead to an unacceptable computational cost. Another method of decision tree optimization is from the bottom to top. Using the ML criterion to conduct the decision tree generation with no stopping criterion, a huge decision tree is generated. In such a huge decision tree, there is almost only one sample for each tree leaf. Then the backing-off for each tree leaf to reduce the "cross" generation error is conducted. The problem, however, is that, backing-off the tree from the bottom does not always lead to the decision tree with the smallest "cross" generation error. It is quite possible that the backing-off process lead to some sub-optimal results. This is the case especially when there are only three tree leaves in the two level sub-tree. Nevertheless, informal experiments conducted by us revealed that, by conducting the decision tree leaf backing-off from the bottom of a huge decision tree as mentioned above, the out-of-training-set generation error of the optimized decision tree is even larger than the generation error by the decision tree initialized by only optimizing the MDL threshold parameters introduced in Section 3.2.

5. Conclusion

In this paper, we have proposed a minimum cross generation error criterion based decision tree pruning method for HMM-based parametric speech synthesis. Rather than generating the decision tree from the top or backing-off from the bottom, we optimize the decision tree from the middle. We first initialize the decision tree by tuning the MDL threshold parameter using the minimum "cross" generation error criterion over the whole decision tree. Then, by further backing-off or splitting tree leaves according to the cross generation error for every single leaf of the decision tree initialized in the first step, the optimal decision tree is obtained. In the decision tree pruning process, the cross generation error is calculated for every tree leaf using CV over the whole training database, and no extra development data set is needed.

In the experimental section, an objective cross generation error and subjective MOS score are both presented. The results show a smaller cross generation error leads to a higher MOS. Finally, subjective preference tests are conducted for the synthesized speech by comparing the baseline system, MDL threshold parameter optimized speech synthesis system and further backing-off and splitting system. The preference ratio indicates the effectiveness of our proposed method. The synthesized speech became more natural after the decision tree

pruning process.

Acknowledgement

This work was partially supported by Hi-Tech Research and Development Program of China (Grant No.: 2006AA01Z137,2006AA010104) and National Natural Science Foundation of China (Grand No.: 60475015). The authors also thank the research division of iFlytek Co. Ltd., Hefei, China, for their help in corpus annotation.

References

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer, New York, U.S.A.

Black, A. W., Zen, H., & Tokuda, K. (2007). Statistical parametric speech synthesis. in *Proc. of ICASSP*, 4, 1229-1232.

Hashimoto, K., Zen, H., Nankaku, Y., Masuko, T., &Tokuda, K. (2009). A Bayesian approach to HMM-based speech synthesis. in *Proc. of ICASSP*, 4029-4032.

Hunt, A. J., & Black, A. W. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. in *Proc. of ICASSP*, 373-376.

Kataoka, S., Mizutani, N., Tokuda, K., & Kitamura, T. (2004). Decision-tree backing-off in HMM-based speech synthesis. In *Proc. of Interspeech*, 1205-1208.

Kawahara, H., Masuda-Katsuse, I., & Cheveigne, A. (1999). 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 Commun*, 27 (3), 187-207.

Ling, Z. H., & Wang, R. (2007), HMM-based hierarchical unit selection combining Kullback-Leibler divergence with likelihood criterion. in *Proc. of ICASSP*, 1245-1248.

Rissanen, J. (1980). *Stochastic complexity in stochastic inquiry*.World Scientific Publishing Company.

Shinoda, K. & Watanabe, T. (2000). MDL-based context dependent subword modeling for speech recognition, *J. Acoust. Soc. Japan(E)*, 21(2), 79-86.

Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T. & Kitamura, T. (2000). Speech parameter generation algorithms for hmm-based speech synthesis. in *Proc. of ICASSP*, 3, 1315-1318.

Tokuda, K., Masuko, T., Miyazaki, N., & Kobayashi, T. (1999). Hidden Markov models based on multi-space probability distribution for pitch pattern modeling. in *Proc. of ICASSP*, 229-232.

Wu, Y.-J., & Wang, R. (2006b). Minimum generation error training for HMM based speech synthesis. in *Proc. of ICASSP*, 89-92.

Wu, Y.-J., Guo, W., & Wang, R. (2006). Minimum generation error criterion for tree-based clustering of context dependent HMMs. in *Proc. of Interspeech*. 2046-2049.

Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., & Kitamura, T. (1999). Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis. in *Proc. of Eurospeech*, 2347-2350.