CROSS-VALIDATION BASED DECISION TREE CLUSTERING FOR HMM-BASED TTS

Yu Zhang[1,2]*, Zhi-Jie Yan*[1] *and Frank K. Soong*[1]

[1]Microsoft Research Asia, Beijing, China
[2]Shanghai Jiao Tong University, Shanghai, China
sjtuzy@gmail.com,{zhijiey,frankkps}@microsoft.com

ABSTRACT

In HMM-based speech synthesis, we usually use complex, context dependent models to characterize prosodically and linguistically rich speech units. It is therefore difficult to prepare training data which can cover all combinatorial possibilities of contexts. A common approach to cope with this insufficient training data problem is to build a clustered tree via the MDL criterion. However, an MDL-based tree still tends to be inadequate in its power to predict unseen data. In this paper, we adopt the cross-validation principle to build such a decision tree to minimize the generation error of unseen contexts. An efficient training algorithm is implemented by exploiting the sufficient statistics. Experimental results show that the proposed method can achieve better speech synthesis results, both objectively and subjectively, than the baseline results of the MDL-based decision tree.

Index Terms— HMM-based speech synthesis, cross validation, context clustering, MDL

1. INTRODUCTION

HMM-based approach has been successfully developed and applied to speech synthesis in the past two decades [1]. In this approach, the spectrum, excitation, and duration features are modeled and generated in a unified HMM framework. In building such an HMM, a large number of contextual factors are used to represent the segmental and supra-segmental information of speech (e.g., phone identity, accent, stress, break) as separate models [2]. However, because the large number of combinatorial possibilities of all contextual factors, it is impossible to obtain enough training data to estimate reliably all full context models. Therefore, a decision tree based model clustering [2, 3] is usually adopted to deal with the data sparseness problem and to predict unseen context in synthesis. This method can successfully produce more robust parameter estimates and improve their generalization capabilities.

Conventional decision tree based clustering is a top-down, data driven training process, based on a greedy tree growing algorithm. The tree growth is based upon two factors, i.e., splitting criterion and stopping criterion. In HMM-based TTS, the splitting criterion is based on Maximum Likelihood (ML) principle. Since the likelihoods increase monotonically with increasing number of decision tree leaf nodes, a stopping criterion, e.g. likelihood thresholding or Minimum Description Length (MDL), needs to be used. Although the conventional method provides an effective and efficient way to build the decision tree for continuous density HMMs, it has several disadvantages: 1) the greedy search-based decision tree growing is sensitive to the training set due to interfering, irrelevant attributes or outlier data [4]. Affected by a small variation in the training set, the algorithm may choose a split which may not be the best one; 2) likelihood threshold is set empirically and it may be dependent upon different tasks or data sets. To alleviate this problem, the minimum description length (MDL) criterion [5] which consists of a model complexity penalty term, is introduced to balance the monotonically growing likelihood. However, the MDL criterion is based on asymptotic assumption and it is not very effective when the amount of training data is not asymptotically large.

In this paper, cross-validation (CV) is adopted for building a decision tree for HMM-based TTS. Cross-validation is a useful technique for many tasks encountered in machine learning, e.g. accuracy estimation, model selection or parameter tuning, etc. In previous studies, cross-validation method has been successfully applied to speech processing, including: Gaussian mixture optimization [6], automatic speech recognition [7], and tuning priors [8]. In this study, K-fold cross-validation is applied to decision tree based model clustering on Multi-space Probability Distribution (MSD) HMMs [9]. First, A cross-validation based splitting criterion is proposed to avoid the conventional greedy splitting criterion and we calculate the likelihood with different validation set with corresponding sufficient statistics. Then, because we calculate the likelihood of the unseen data with the current model parameters, tree-growing can be stopped automatically. Using the proposed splitting and stopping criteria, we are able to build a better decision tree and improve its generalization capability to synthesize unseen contexts.

The cross-validation based decision tree clustering algorithm was evaluated in our HMM-based TTS system. We compared several objective and subjective measures of the synthesized speech using conventional method and the cross-validation based method. The experimental results show that the CV decision tree yields better Log Spectral Distance (LSD), root mean square error of f0 and duration model objectively than the conventional decision tree. The speech quality improvement is also confirmed by the subjective preference test results.

The rest of this paper is organized as follows: In Section 2, the splitting and stopping criteria in conventional MDL-based decision tree are presented. In Section 3, the cross-validation based decision tree in TTS is introduced. In Section 4, we present the experimental results. In Section 5, we draw our conclusion.

2. MDL-BASED DECISION TREE CLUSTERING

Traditionally, the ML criterion is used as node splitting criterion for tree growing. The ML criterion for splitting tree nodes is consistent with that used in training HMMs parameters. Let $\mathcal{L}(S)$ denote the log likelihood of generating observation frames at node S. Fig.1

The work was done during the first author's internship in Microsoft Research Asia.

shows the tree growing procedure. Suppose that node S_m is split

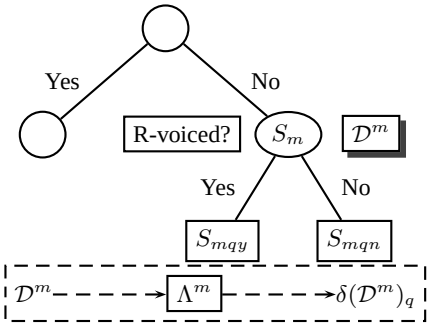

Fig. 1. Node splitting of MDL-based decision tree

into two successor nodes, S_{mqy} and S_{mqn} by a binary (yes or no) question q. The increase of log likelihood by splitting S_m through a question q is [2]:

$$\delta(\mathcal{D}^m)_q^{ML} \quad = \quad \mathcal{L}(S_{mqy}) + \mathcal{L}(S_{mqn}) - \mathcal{L}(S_m)$$

Log likelihood increases monotonically with increasing number of terminal leafs. As a result, a threshold of likelihood improvement (change) is therefore necessary to terminate the node splitting. On the other hand, the MDL criterion evaluates the splitting performance according to the description length, which consists of a likelihood term and a penalty term associated with the model complexity. We can calculate the splitting cost by the following equations [5]:

$$\delta(\mathcal{D}^m)_q^{MDL} \quad = \quad \delta(\mathcal{D}^m)_q^{ML} - \alpha L \log G \tag{1}$$

where G is the total number of data samples, L the increase of model parameters when splitting one node, α the scaling factor which is used to balance the likelihood and model complexity, respectively.

The physical meaning of MDL aims at building a tree model which can balance data likelihood and model complexity. But there are two drawbacks of this method: 1) **Splitting criterion** which may be sensitive to the training set due to some irrelevant attributes or outlier data. 2) **Stopping criterion** of MDL is based on asymptotic assumption and it is equivalent to a likelihood threshold. In most applications, we often need to tune the penalty factor to determine an appropriate tree.

3. CROSS-VALIDATION BASED DECISION TREE CLUSTERING

In order to overcome the above mentioned problems in the traditional MDL-based decision tree, it is desirable to build a decision tree that can explicitly minimize the generalization error and select the model topology (complexity) automatically. In this study we use cross validation for node splitting and tree growing stopping criteria.

3.1. Decision Tree based on Cross Validation

In cross validation, we divide the training data $\mathcal{D}^m$ into K subsets $\mathcal{D}_i^m, i = 1, \ldots, K$ at node S_m. Among the K subsets, a single subset $\mathcal{D}_k^m$ is reserved as validation data, i.e., to test the model, and the remaining $K-1$ subsets, $T_k = \mathcal{D}^m \setminus \mathcal{D}_k^m$[1] are used as training data. The cross-validation process is then repeated K times (the folds), with each of the K subsets used exactly once as the validation data.

[1] $B \setminus A$ is the set of all elements which are members of B, but not members of A.

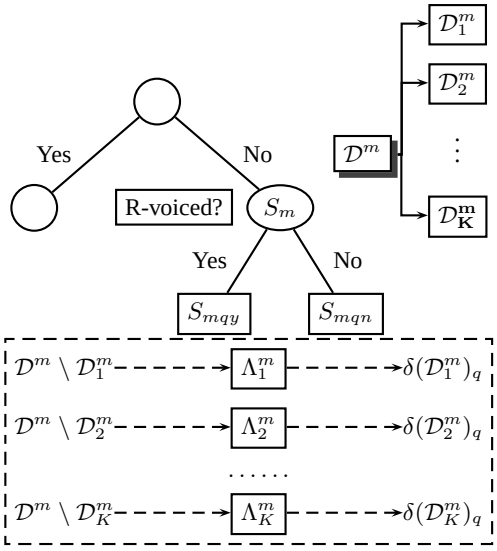

Fig. 2. Node splitting of cross validation based decision tree

Based on this procedure, we can select the question which gives the highest scores on all validation data. It is not limited to, but in this study we use the log likelihood improvement as the score function.

3.1.1. Node Splitting Criteria

Fig.2 shows the node splitting procedure. By assuming the alignments are fixed during the optimization process, we can evaluate the log likelihood on each validation data as follows

$$\mathcal{L}_k^{CV}(\mathcal{D}_k^m) = \sum_{\boldsymbol{x} \in \mathcal{D}_k^m} P(\boldsymbol{x}|\Lambda_k^m) \tag{2}$$

where Λ_k are the model parameter estimate from T_k. The increase of log likelihood by splitting S_m through the yes and no question q is given as

$$\delta^{CV}(\mathcal{D}_k^m)_q = L_k^{CV}(\mathcal{D}_k^{mqy}) + L_k^{CV}(\mathcal{D}_k^{mqn}) - L_k^{CV}(\mathcal{D}_k^m) \tag{3}$$

where $\mathcal{D}_k^{mqy} = \{x | x \in \mathcal{D}_k^m, \text{Question}(x) = yes\}$ and $\mathcal{D}_k^{mqn} = \{x | x \in \mathcal{D}_k^m, \text{Question}(x) = no\}$.

In this definition we select the best question for node splitting according to its likelihood increase on all the validation data

$$q_m = \arg\max_q \bigsqcup_k \delta^{CV}(\mathcal{D}_k^m)_q \tag{4}$$

Note that we can give $\bigsqcup$ different definitions, e.g. voting, maximizing or bagging. According to the given definitions, the best question has different physical interpretations. In this study, we define $\bigsqcup = \sum$. For this definition, the node splitting criterion is to reduce the bias.

3.1.2. Stopping Criteria

Because we calculate each $L_k^{CV}(\mathcal{D}_k^m)$ on the validation data sets, the tree splitting can stop automatically when

$$\bigsqcup_k \delta^{CV}(\mathcal{D}_k^m)_{q_m} < 0 \tag{5}$$

It's similar to the splitting criterion. We can also combine it with MDL as

$$\bigsqcup_k \delta^{CV}(\mathcal{D}_k^m)_{q_m} + \alpha L \log G < 0 \quad (6)$$

Eq.(6) can be used to generate different size decision tree.

To be consistent with the node splitting criterion, we define $\bigsqcup = \sum$. In our experiments, we found that this natural stopping gives good results.

4. EXPERIMENT AND RESULTS

A Chinese speech corpus of 1,000 recorded by a female speaker is used in our experiments. The recorded sentences were sampled at 16 kHz. 40^{th}-order LSP coefficients plus gain, as well as their first and second order dynamic features are extracted. They are used to train the ML-based, decision tree-tied baseline model. HMMs of 5-states, left-to-right, no-skip topology with diagonal covariance matrix are used to build all phone models. There are 25,761 different rich context phone models seen in the training corpus.

Separated development and test sets, each consisting of 50 sentences, respectively, are selected for our experiments. Parametric speech trajectories are synthesized by the conventional decision tree-tied models, and our new CV decision tree. Two synthesis systems based on LSP features are built for comparison: **Conventional MDL-based decision tree** and **Cross-Validation based decision tree**. We first train the model parameter by tuning the MDL parameter on the development set. Then we compare the two systems both objectively and subjectively.

4.1. Implementation Issues

In cross validation method, we need to access all data in each node. To reduce effort of revisiting the data and corresponding computations, we can access all the training data once in a preprocessing stage to collect all necessary sufficient statistics. The cross-validation likelihood can then be computed efficiently using the pre-computed sufficient statistics [6]. Because of space limitation, detail description of the procedure is omitted here.

4.2. Objective Test Results

4.2.1. Objective measures

In this paper, we use the following objective measures to estimate the distortion between the generated (gen) and reference (ref) parameters of spectrum, f0, duration, respectively. Here we use the extracted spectrum and manually checked f0 as the reference.
1) Log Spectral Distance

$$D_{\text{LSD}} = \frac{1}{T_{\text{voiced}}} \sum_{t=1}^{T_{\text{voiced}}} \sqrt{\frac{1}{N_{\text{FFT}}} \sum_{i=1}^{N_{\text{FFT}}} (l_{\text{ref}}(t,i) - l_{\text{gen}}(t,i))^2} \quad (7)$$

where the T_{voiced} is the number of voiced frames. N_{FFT} is the number of frequency points of each frame. l is the value of log magnitude spectrum (in dB).
2) Root mean square error of F0

$$D_{f_0} = \sqrt{\frac{1}{T_{\text{voiced}}} \sum_{t=1}^{T_{\text{voiced}}} (f_{\text{ref}}(t) - f_{\text{gen}}(t))^2} \quad (8)$$

where $f(t)$ is the fundamental frequency of frame t.
3) Root mean square error between force aligned reference and synthesis state durations

$$D_{dur} = \sqrt{\frac{1}{S} \sum_{s=1} (d_{\text{ref}}(s) - d_{\text{gen}}(s))^2} \quad (9)$$

where $d(s)$ is the duration in frames of state s.

4.2.2. Determining the number of cross-validation folds

In K-fold cross validation, we first need to determine the fold number K. We evaluate several K values, from 3 to 15, by using the development set. The results of log-spectral distortions are given in Table.1. We found that LSD is not sensitive to K values. Because of this result, we fix $K = 10$ for the rest of our experiments.

K	4	6	8	10	14
LSD (dB)	5.32	5.33	5.32	5.32	5.31

Table 1. The log spectral distortion for different K on the development set

4.2.3. Results

Using the MDL-based decision tree splitting, and with different penalty scaling factor α, we can plot the distortion curves of all objective measures on the test set, shown as the diamond curves in Figs.3-5. In practice, we also need to determine an "operating point" along these curves, which is usually done by tuning α on a development set, or simply set α to be 1.0. In our experiments, the optimal operating points determined on the development set for spectrum, f0 and duration models are: $\alpha_{\text{lsp}} = 0.5$, $\alpha_{\text{f0}} = 0.5$ and $\alpha_{\text{dur}} = 0.8$, respectively.

Then, the distortion curves of all objective measures on the test set are plotted, as the diamond curves in Figs.3-5. We also mark these "operating points" in their corresponding figures. From the results, we can see the α values tuned on the development set also yield reasonably good but still not the best performance on the test set.

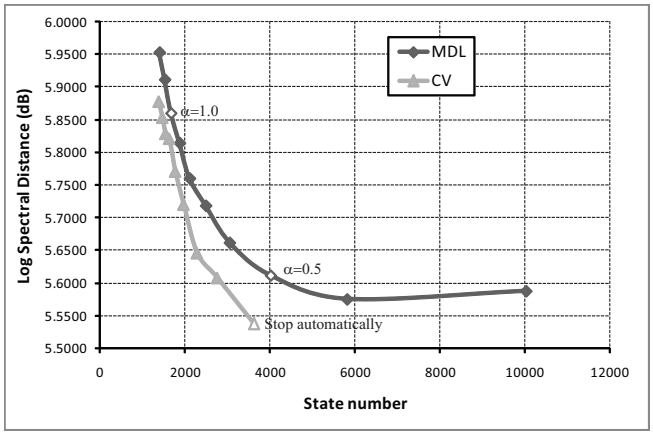

Fig. 3. Performance comparison of MDL criterion (MDL) vs. cross-validation (CV) on log spectral distance

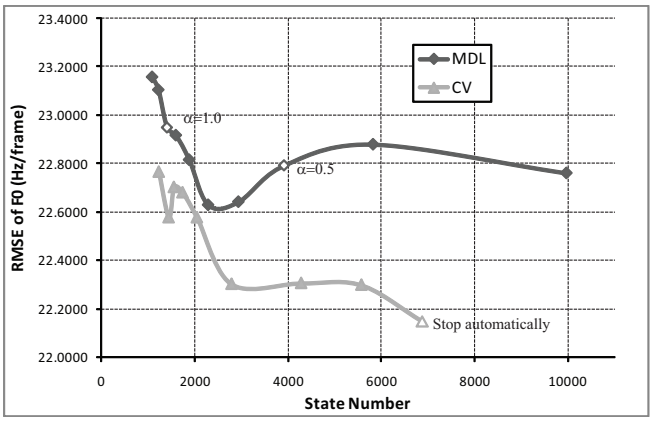

Fig. 4. Performance comparison of MDL criterion (MDL) vs. cross-validation (CV) on F0

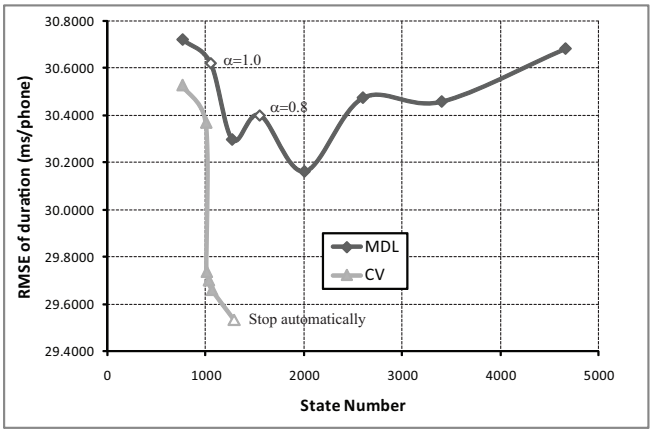

Fig. 5. Performance comparison of MDL criterion (MDL) vs. cross-validation (CV) on state duration

The distortion curve using the cross-validation-based criterion is plotted as the triangle line in Figs.3-5. To get similar model size, i.e., number of model parameters, a threshold is imposed as (Eq.(6)). As we can see from the figures, 1) The cross-validation method always give better performance when the two systems have similar number of model parameters. 2) The CV decision tree stops automatically. 3) Compared with spectrum and duration, the cross-validation decision tree for f0 has significantly larger number of terminal leaves than an MDL-based decision tree. This is due to the fact that splitting of the unvoiced space in MSD-HMM can always get a marginal likelihood increase. However, since this splitting does not effect the voiced/unvoiced decision in synthesis, it has no significant effects on the final result.

4.3. Subjective Test Results

In the subjective test we compare standard MDL based with the 10-fold cross-validation based decision trees. A separated test set of 50 sentences is selected in our experiments for an AB comparison preference test. Eight subjects are invited to listen to randomized pairs of sentences synthesized by the two methods, and to provide their preference. The results of the preference test are given in Fig.6 where shows our method achieves a better performance.

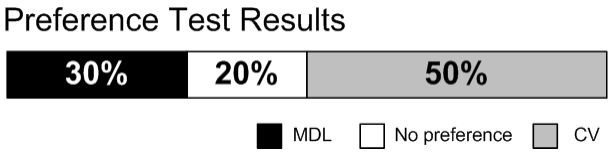

Fig. 6. The result of preference test for two system

5. CONCLUSIONS AND FEATURE WORK

We propose a training algorithm for building a decision tree, which can maximize its prediction capability via cross validation and stop the tree growing automatically for the given data. Experimental results show that in comparison with MDL training, a cross-validation based decision tree yields a better synthesis performance with a similar model size. It also can find an appropriate model size from the development set. The cross-validation based new decision tree construction facilitates a better (more robust) node splitting and an automatic stopping criterion for its growth. In the future we will use larger speech databases to verify that the concept of cross validation is also extendable to different sized databases and other languages.

6. REFERENCES

[1] K. Tokuda, T. Kobayashi, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for hmm-based speech synthesis," in *Proc. ICASSP*, 2000, pp. 1315–1318.

[2] J. J. Odell, "The use of context in large vocabulary speech recognition," 1995.

[3] S.J. Young, J.J. Odell, and P.C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," 1994.

[4] L. Rokach and O. Maimon, *Data Mining with Decision Trees: Theory and Applications*, World Scientific, 2008.

[5] K. Shinoda and T. Watanabe, "Acoustic modeling based on the mdl principle for speech recognition," in *Proc. EuroSpeech*, 1997, pp. 99–102.

[6] T. Shinozaki, S. Furui, and T. Kawahara, "Aggregated cross-validation and its efficient application to gaussian mixture optimization," in *Proc. Interspeech*, 2008, pp. 2382–2385.

[7] I. Rogina, "Automatic architecture design by likelihood-based context clustering with crossvalidation," in *Proc. Eurospeech 97*, 1997, pp. 1223–1226.

[8] K. Hashimoto, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "Bayesian context clustering using cross valid prior distribution for hmm-based speech recognition," in *Proc. Interspeech*, 2008, pp. 936–939.

[9] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Multi-space probability distribution hmm," in *IEICE Trans. Inf. & Syst.*, 2002, pp. 455–464.