AN ITERATIVE APPROACH TO DECISION TREE TRAINING FOR CONTEXT DEPENDENT SPEECH SYNTHESIS

Xiayu Chen, Yang Zhang, Mark Hasegawa-Johnson

Department of Electrical and Computer Engineering,
University of Illinois at Urbana-Champaign, IL
ahcxy2010@gmail.com, yzhan143@illinois.edu , jhasegaw@illinois.edu

ABSTRACT

In speech synthesis with sparse training data, phonetic decision trees are frequently used for balance between model complexity and available data. The traditional training procedure is that decision trees are constructed after parameters for each phones optimized in the EM algorithm. This paper proposes an iterative re-optimization algorithm in which the decision tree is re-learned after every iteration of the EM algorithm. The performance of the new procedure is compared with the original procedure by training parameters for MFCC and F0 features using an EDHMM model with data from The Boston University Radio Speech corpus. A convergence proof is presented, and experimental tests demonstrate that iterative re-optimization generates statistically significant test corpus log-likelihood improvements.

Index Terms— speech synthesis, speech clustering, EM algorithm, decision tree

1. INTRODUCTION

The explicit duration hidden Markov model (EDHMM) [3] is widely used in parametric speech synthesis and speech recognition systems [4]. Essentially EDHMM assumes that the Markov state chain has an explicitly modeled duration, as opposed to the incorrect geometric duration assumption inherent in a traditional HMM.

As speech parameters (F0, MFCC, etc.) vary greatly in different contexts, context dependent modeling is widely applied. However there are usually not enough data to train a different model for every different context. Therefore, it is useful to cluster contexts where these speech parameters have similar probabilistic behavior. Odell [3] introduces the decision tree as a useful clustering technique. A decision tree is a binary tree whose leaf nodes are context-dependent clusters, with each observation directed to a leaf node cluster from the root node by answering the questions in non-terminal nodes all the way down. Decision trees are constructed essentially by maximizing the log likelihood of training data with some constraints preventing overfitting. Usually finding the global optimal tree is intractable and in practice, people normally use a greedy algorithm. The traditional tree construction algorithm that trains a decision tree for each hidden state, as in [2], relies on an important approximation assumption: state assignment probability is insensitive to the tree structure. In other words, though a change in tree structure changes the state-conditional likelihood of observation, this change is unlikely to impact state assignment. Therefore this algorithm can be divided into two phases. In phase 1, each state is assumed to have only one cluster, and state assignment probability is calculated using the EM algorithm; in phase 2, state assignment probability is fixed and decision tree for each node is constructed. We are therefore interested in the impact of this approximation, which, for convenience, called Out-EM procedure. Specifically, we would like to investigate if there's an efficient algorithm that removes this approximation, and evaluate what improvement can be achieved if this approximation is eliminated. This paper proposes an algorithm that incorporates tree construction in each EM iteration (the In-EM procedure). It turns out that, by adding a safeguard step, we can guarantee convergence of the new algorithm. Experiments in this paper show that the log-likelihood improvements provided by In-EM are statistically significant, and furthermore the improvements can be intuitively understood in terms of structural properties of the trees.

The following sections are arranged as follows: Section 2 derives the algorithm, and proves convergence. Experiments comparing the new algorithm with the traditional one are carried using The Boston University Radio Speech corpus and the results are presented in Section 3. Conclusions are reviewed in Section 4.

2. DERIVATION OF IN-EM ALGORITHM

In this section, the training procedure of In-EM algorithm will be derived. We will find parameter reestimation similar to that of the traditional EM algorithm, and that by adding one double checking step, In-EM training is ensured to reach convergence.

2.1. Background: model assumptions

Although the In-EM algorithm can be applied to tree-based clustering models for any speech parameters, we concretize our derivation by assuming the EDHMM model of MFCC and F0 for speech synthesis, as proposed in [3]. Specifically, we use a 3-state EDHMM for each phone. In EDHMM, the event of transition from state i to j at t is divided into two independent events: one event is that state i ends at t, the other is that state i transits to state j. Probability of the latter is determined by transition probability $a(j|i)$, while the former relies on the time of the previous transition and duration model of i.

Given each state $I = i$, there are two separate decision trees, one for MFCC, and the other for F0. Leaf nodes of both are mixtures. Given each mixture component $K = k$ for MFCC, we assume a Gaussian distribution with diagonal covariance, namely

$$P_{X_{MFCC}|I,K}(x|i,k) = \mathcal{N}_2(x; M_{ik}, \Sigma_{ik}) \qquad (1)$$

Given mixture component $S = s$ for F0, we assume a multi-space distribution (MSD) model [1]. In an MSD model, the sample space Ω consists of G spaces, $\Omega = \bigcup_{g=1}^{G} \Omega_g$, where Ω_g is an n_g-dimensional real space indexed by g. Each space Ω_g has its prior w_g, i.e., $P(\Omega_g) = w_g$, where $\sum_{g=1}^{G} w_g = 1$. A space with $n_g > 0$

Thanks to the Illinois Innovation Initiative and 985 fund for funding.

has a probability density function $\mathcal{N}_g(X), X \in R^{n_g}$. For F0, $G = 2$, as data can be divided into voiced and unvoiced categories, unvoiced frames belonging to sample space Ω_1 where $n_1 = 0$, and voiced frames belonging to Ω_2 where $n_2 = 1$. Distribution for voiced frames is a Gaussian distribution $\mathcal{N}(\mu, \sigma)$. According to [1], it's easy to derive that given mixture component s, probability of observed F0 data X_{F0} is:

$$P_{X_{F0}|I,S}(x|i,s) = \begin{cases} w_{is1} & X_{F0} \in unvoiced \\ w_{is2}\mathcal{N}_2(x;\mu_{is},\sigma_{is}) & X_{F0} \in voiced \end{cases} \tag{2}$$

2.2. Derivation and convergence proof

In this paper, we use the minimum description length principle as the stopping rule of tree clustering. The target to minimize is the description length :

$$l = -\log\left(P_X(x|\Theta)\right) + D(\Theta) \tag{3}$$

where X is all the observed features, i.e. MFCC and F0; Θ stands for parameters of the acoustic model, including both the structure of the decision trees and parameters for each node; and $D(\Theta)$ is the term to prevent overfitting, and is determined by the number of parameters in Θ only

$$D(\Theta) = \frac{\alpha_j}{2}\log N + \log J \tag{4}$$

α_j is the number of free parameters of the model j; N is the number of training frames; and J is the number of models. According to Jensen's inequality,

$$\begin{aligned} \log\left(P_X\left(x|\Theta\right)\right) &= \log\left(\int_z P_{X,Z}\left(x,z|\Theta\right)\,dz\right) \\ &= \log\left(\int_z R\left(z\right) P_{X,Z}\left(x,z|\Theta\right) R(z)\,dz\right) \\ &\geq \int_z R(z)\log\frac{P_{X,Z}\left(x,z|\Theta\right)}{R\left(z\right)}\,dz \end{aligned} \tag{5}$$

where Z denotes hidden variables, including state I, MFCC mixture component K and F0 mixture component S. Therefore the upper bound of (3) could be represented as

$$-\int_z R(z)\log\frac{P_{X,Z}\left(x,z|\Theta\right)}{R\left(z\right)}\,dz + D(\Theta) \tag{6}$$

To minimize the target function ,we can minimize its upper bound. Minimization of (6) can be carried out by minimizing over $R(z)$ and Θ alternatively, and thus EM algorithm is involved.

First, in the E step, minimization over $R(z)$ is achieved when

$$R(z) = P_{Z|X}(z|x,\Theta) \tag{7}$$

and therefore,

$$\begin{aligned} &\int_z R(z)\log\frac{P_{X,Z}(x,z|\Theta)}{R(z)}\,dz + D(\Theta) \\ &= -\log\left(P_X\left(x|\Theta\right)\right) + D(\Theta) \end{aligned} \tag{8}$$

which means that the bound is tight at the end of E step.

Then in M step, minimizing is carried out with Θ while $R(z)$ stays unchanged, thus the target description length l is rewritten as

$$\begin{aligned} l = &-\int_z P_{Z|X}(z|x,\Theta_{old})\log\left(\frac{P_{X,Z}(x,z|\Theta)}{P_{X,Z}\left(x,z|\Theta_{old}\right)}\right) \\ &+ D(\Theta) \end{aligned} \tag{9}$$

As is mentioned before, the Θ includes the structure of the decision tree and parameters for each node, thus optimization is performed in two steps.

2.2.1. *Reestimation of parameters for each node*

In the first step, given the tree structure, parameters for each cluster node are updated, which is very similar to the traditional EM algorithm. For example, update equations for F0 parameters [1] are

$$\begin{aligned} w_{isg} &= \frac{\sum_{t\in T(X,g)}\gamma_t(i,s,g)}{\sum_{h=1}^{2}\sum_{t\in T(X,h)}\gamma_t(i,s,h)} \\ \mu_{is} &= \frac{\sum_{t\in T(X,2)}\gamma_t(i,s,2)X_t}{\sum_{t\in T(X,2)}\gamma_t(i,s,2)} \\ \sigma_{is} &= \frac{\sum_{t\in T(X,2)}\gamma_t(i,s,2)(X_t-\mu_{is})(X_t-\mu_{is})'}{\sum_{t\in T(X,2)}\gamma_t(i,s,2)} \end{aligned} \tag{10}$$

where g is a space index from index set $\{1, 2\}$; w_{isg} is weight of space g in cluster s, state i; $T(X, g)$ is a set of time t indices whose observation vectors X_t include space index g, and $\gamma_t(i, s, g)$ is the posterior probability of being in space g of cluster s, state i at time t, calculated as in [4].

2.2.2. *Reshaping of tree structure*

In the second step, to reach the optimal tree structure, we use a greedy algorithm to construct a decision tree. We will use the tree for F0 as illustration. Given hidden state $I = i$, according to (10), it can be proved using the trace identity that (9) can be rewritten as

$$\begin{aligned} l = &\sum_{s\in S}\sum_{g=1}^{G}\frac{1}{2}\left(n_g\left(\log\left(2\pi\right)+1\right) + log|\sigma_{isg}| - 2\log w_{isg}\right) \\ &\sum_{t\in T(X,g)}\gamma_t(i,s,g)_{old} + \\ &\left(\sum_{s\in S}\sum_{g=1}^{G}\frac{1}{2}\left(2n_g+1\right)\right)\left(\log\sum_{s\in S}\sum_{g=1}^{G}\sum_{t\in T(X,g)}\gamma_t\left(i,s,g\right)_{old}\right) \end{aligned} \tag{11}$$

We will start from a single root node, and split the nodes into two descendants subsequently. At the point of splitting cluster S_i into two descendent nodes S_{il} and S_{ir}, the question is selected from the question pool such that it minimizes δl ($\delta l = l_{new} - l$), where

$$\begin{aligned} \delta l = &\sum_{s\in S_{il\setminus ir}}\sum_{g=1}^{G}\frac{1}{2}\left(log\left(|\sigma_{isg}|\right) - 2logw_{isg}\right)\sum_{t\in T(X,g)}\gamma_t(i,s,g)_{old} \\ &-\sum_{s\in S_i}\sum_{g=1}^{G}\frac{1}{2}\left(\log\left(|\sigma_{isg}|\right) - 2logw_{isg}\right)\sum_{t\in T(X,g)}\gamma_t(i,s,g)_{old} \\ &+\left(\sum_{g=1}^{G}\frac{1}{2}\left(2n_g+1\right)\right)\left(log\sum_{s\in S}\sum_{g=1}^{G}\sum_{t\in T(X,g)}\gamma_t(i,s,g)_{old}\right) \end{aligned} \tag{12}$$

If no question decreases description length, the split stops.

Since the greedy algorithm generally does not lead to the optimum solution, it is possible that a newly constructed tree may not increase the upper bound. To ensure convergence, we add a safeguard step where every time the E step is finished, if the value of target function increases instead of decreasing, meaning the greedy algorithm finds a poorer tree structure, the new tree will be abandoned. Instead, only the distribution given each cluster is updated.

2.3. Steps of In-EM training

As a recap, the proposed In-EM procedure iterates the following steps until description length converges:

(i) Calculate the state assignment probability, $\gamma_t(i, s, g)$;

(ii) Given each state, the two decision trees are initialized to contain only one node whose parameters are randomly chosen, and the initial value of description length is set as negative infinity;

(iii) Construct each tree using a greedy algorithm, until description length can no longer decrease;

(iv) Update the parameters of each cluster node.

(v) Compare the target function with that in the previous iteration; if the target function decreases, go to step (i); else go to step (vi)

(vi) Maintain the decision tree structure in the last iteration while updating parameters in each node.

Since step (vi) is guaranteed to increase the upper bound, this double check can ensure decrease of target function after each iteration. For this reason, the new procedure is ensured to reach convergence. However, as parameters for the tree structure are discrete, this training algorithm is likely to be trapped at a local optimum. To mitigate this problem, we used the method of random restart, which is explained in detail in section 3.

3. EXPERIMENT AND ANALYSIS

3.1. Experiment configuration

In this section, a set of experiments comparing In-EM and Out-EM algorithms are described and analyzed. These experiments are conducted on the Boston Radio Speech Corpus [6], an American English Corpus with comprehensive annotations on phone, pitch, prosodic boundary etc. To facilitate preliminary experiments, silent phone (including pause, silence and closure of plosives) and phones with too few data are discarded. The dataset thus includes 37 phones with up to 33511 segments. One segment is a complete record of the phonation a phone, consisting of several frames.

The experiments are divided into two parts. The first part, where all data are incorporated in training set, will demonstrate that In-EM is able to better optimize the target function, and evaluate approximation error of Out-EM. The second part, where frames are split into equally-sized training set and test set, will demonstrate that the superiority of In-EM is generalizable to data outside the training set. These two parts will be discussed in detail in subsequent subsections respectively.

3.2. Convergence analysis

In this part, decision trees are trained with MDL as the stopping rule. As is derived in section 2, description length is essentially the target function for training, and therefore is used as the evaluation metric in this part, i.e. an algorithm that achieves greater description length is considered superior. As also mentioned, In-EM algorithm is easily trapped in local optima. For this reason, In-EM algorithm is run for 20 times with different random initializations. In this way, we expect to see that the In-EM algorithm has better performance than Out-EM statistically.

To study the performance on MFCC and F0 separately, trees of F0 are controlled when trees of MFCC are trained with the two distinct algorithms, and vice versa. Table 1 shows the percentage of instances where In-EM achieves better description length than Out-EM for MFCC (with trees of F0 controlled) and F0 (with trees of MFCC controlled). To see the relationship of tree scale and performance, the number of leaf nodes for a decision tree calculated in Out-EM algorithm is also presented. The phonemes being displayed in the table are selected randomly.

Table 1: *Percentage of experiments (P) in which In-EM description length is better than Out-EM for MFCC and F0, u stands for unvoiced, v for voiced phone, LN is the number of leaf nodes of decision trees of the 3 state*

Phone	u/v	P_{MFCC}	LN of MFCC	P_{F0}	LN of F0
$/p/$	u	90%	(8,3,5)	80%	(1,1,4)
$/k/$	u	100%	(8,13,10)	30%	(1,1,11)
$/s/$	u	95%	(6,8,14)	55%	(12,1,12)
$/l/$	v	100%	(10,6,11)	100%	(12,4,8)
$/ng/$	v	95%	(2,2,5)	100%	(2,2,9)
$/ih/$	v	100%	(10,13,13)	65%	(13,3,20)
$/ao/$	v	100%	(12,7,5)	100%	(5,5,4)
$/oy/$	v	70%	(2,1,2)	15%	(1,1,2)
Total	-	90.6%	-	70.045%	-

Table 1 shows that in most cases, the In-EM algorithm can better minimize the target function than Out-EM. Furthermore, it can also be observed that this advantage of In-EM algorithm is more evident when the decision tree is large. For instance, phone $/oy/$, the worst case being displayed for MFCC, has the fewest number of nodes; while phones where In-EM for MFCC has 100% outperformance have on average 10 nodes for each state. This is because when leaf nodes of decision tree are too sparse, the trees constructed In or Out EM are simple, with limited difference in their structure. Conversely for a properly sized tree, not only the question selected make a difference, but the position each question took to shape the tree also matters. In this case, the difference in structure of decision trees trained with In-EM from trees trained with Out-EM leads to obvious outperformance. It is also presented that the advantage of In-EM for F0 is more distinct with voiced phones. This can similarly be explained by the scale of decision tree. For unvoiced phoneme, data are sparse to the extent that decision trees for some of the states only contain one node. In this case, In-EM is less likely to find a better structured decision tree.

Examples of decision trees trained In/Out EM are presented in Fig. 1. It shows that structure of decision tree trained In and Out EM are different, yet the main questions involved for clustering for a certain state of a certain phoneme are relatively stable.

Typical convergence curves are presented in Fig. 2. The convergence patterns of In and Out EM are very different. Since parameters for decision tree structure are discrete, In-EM description length generally has a small but evident decrease once every few iterations. Those decreases indicate that the structure of the decision tree changes. In contrast, the curve of Out-EM training decreases gradually in the first phase and then undergoes a large decrease, because that is when decision trees are constructed. After that, the value of DL for out-EM training stays stable. In Fig. 2, it is obvious that In-EM outperforms Out-EM in most cases. Poor equilibrium of In-EM is usually accompanied by slow convergence.

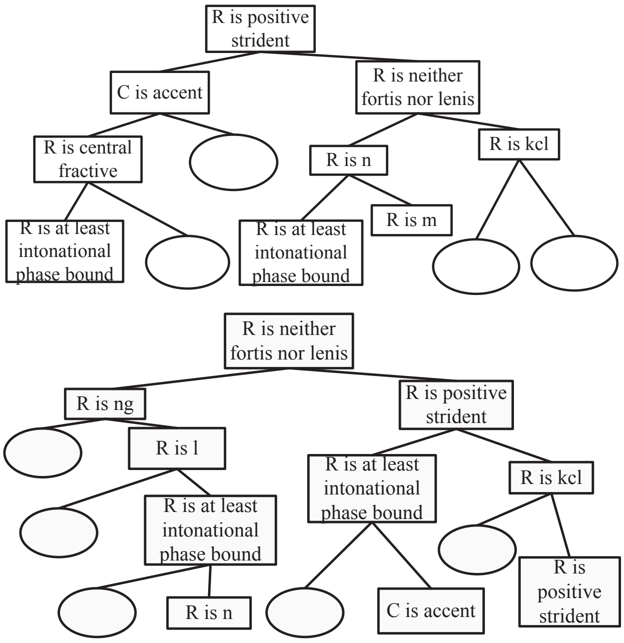
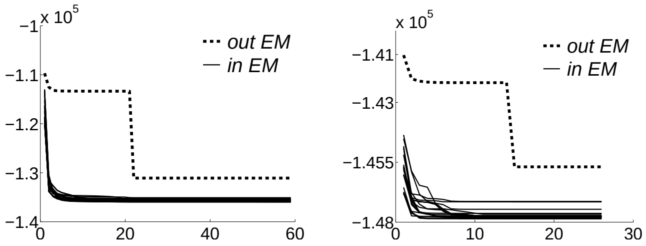

R is positive strident
C is accent
R is neither fortis nor lenis
R is central fractive
R is n
R is kcl
R is m
R is at least intonational phase bound
R is at least intonational phase bound
R is neither fortis nor lenis
R is ng
R is positive strident
R is l
R is at least intonational phase bound
R is kcl
R is at least intonational phase bound
R is positive strident
R is n
C is accent

Fig. 1: *Example decision trees. Top panel: MFCC tree for phoneme $/ae/$ state 1 trained Out-EM. Bottom panel: MFCC tree for phoneme $/ae/$ state 1 trained In-EM. R stands for right, L stands for left and C stands for current. Expression-like questions are on prosodic break level. Nodes with depth greater than 3 are omitted.*



(a) *MFCC* (b) *F0*

Fig. 2: *Convergence curve of phone $/ao/$, for MFCC and F0 training. Solid lines for 20 randomly initialized In-EM training , dotted line for Out-EM training. The curve begins after the first iteration.*

Table 2: *Percentage of experiments whose In-EM test likelihood is better than Out-EM for MFCC and F0, P stands for $P(N,T|H_0)$*

phon.	T	$\frac{N_{MFCC}}{T}$	P_{MFCC}	$\frac{N_{F0}}{T}$	P_{F0}
$/k/$	534	57.39%	$3*10^{-4}$	61.05%	$< 10^{-10}$
$/s/$	1456	76.58%	$< 10^{-10}$	59.55%	$< 10^{-10}$
$/th/$	143	69.23%	$< 10^{-10}$	53.85%	0.1786
$/m/$	608	63.65%	$< 10^{-10}$	66.61%	$< 10^{-10}$
$/eh/$	664	51.20%	0.2681	64.31%	$< 10^{-10}$
$/oy/$	23	26.09%	0.9947	78.26%	0.005
$/er/$	129	64.34%	$5*10^{-4}$	77.52%	$< 10^{-10}$
Total	18072	60.69%	$< 10^{-10}$	60.14%	$< 10^{-10}$

3.3. Test set analysis

In this subsection, the data set of each phoneme is divided into two sets with almost equal number of frames. The first set is for training, while the second is for testing. In this experiment, training is performed on the training set, and after that, the likelihood of test data are calculated as the performance metric.

To formally compare the performance on test data, we perform a hypothesis test where the null hypothesis H_0 is that the joint likelihood of the model trained with In-EM is equal to that trained with Out-EM , namely

$$H_0 : P\left(P_X\left(x|\Theta_{In-EM}\right) > P_X\left(x|\Theta_{Out-EM}\right)\right) = 0.5 \quad (13)$$

where x is the observed feature vectors, i.e. MFCC and pitch, of segments in the testing data, $P_X\left(x|\Theta_{In-EM}\right)$ and $P_X\left(x|\Theta_{Out-EM}\right)$ are posterior probability of x calculated with parameters of Out-EM training Θ_{Out-EM} and In-EM training Θ_{In-EM}. If H_0 is true, then the probability that the likelihood of the model trained by In-EM is greater than that by Out-EM in at least N out of T segments follows the binomial tail distribution

$$P(N,T|H_0) = \sum_{n\geq N} \binom{T}{N} \left(\frac{1}{2}\right)^T \quad (14)$$

We perform the hypothesis test under 90% confidence level, which means if $P(N,T|H_0) < 0.1$, then the null hypothesis is rejected.

Similar to the previous subsection, the In-EM procedure for each phone is also carried out 20 times with random initialization. However, in this experiment, only the decision tree with least description length is selected to compare to the decision tree trained with Out-EM for the same phone. The number of segments whose joint likelihood of In-EM are better than that of Out-EM is counted for each phone and summed to calculate $P(N,T|H_0)$. Table 2 shows the result of some phones and the statistical result of all phones. The phones being displayed in the table are selected randomly.

The statistical result of all phones for both the training of MFCC and F0 show that, if null hypothesis H_0 is true, $P(N|T,H_0) \ll 0.1$, thus the null hypothesis is rejected, which shows that in general, In-EM does train a better model that is generalizable to all the data. The reason why some phones show large $P(N|T,H_0)$ is that when the number of training and testing data for a single phone is too small to represent all situations, type I error may happen.

4. CONCLUSION

In this paper, a new procedure of decision tree training in speech synthesis is proposed, and it is proven that the procedure converges. We find out that, though tree trained In or Out EM are disparately shaped, the questions involved for a given state of one phone are relatively stable. By the differently shaped decision trees, the In-EM decision tree can converge at superior description length of the training data in most cases. The occurrences of posterior probabilities of testing data calculated with In-EM tree better than those calculated with Out-EM tree are frequent enough to be statistically significant. Therefore the new procedure proposed in this paper is proved to be a better procedure for state clustering.

5. REFERENCES

[1] Takayoshi Yoshimura, “Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for hmm-based text-to-speech systems,” Ph.D. dissertation, Dept. Elec. Comp. Eng., Nagoya Inst. of Technology, Nagoya, Japan, 2002.

[2] Julian J. Odell, “The use of context in large vocabulary speech recognition,” Ph.D. dissertation, Queens’ College, Univ. of Cambridge, Cambridge, UK, 1995.

[3] J. D. Ferguson, Variable duration models for speech, in Proc. Symp. Application of Hidden Markov Models to Text and Speech, Princeton, NJ, 1980, pp. 141-179.

[4] Ken Chen, Mark Hasegawa-johnson , and Aaron Cohen , “Prosody dependent speech recognition on the radio news corpus of American English,”*IEEE Transaction On Speech and Audio Processing*, vol. 13, no. 6, Nov. 2005.

[5] Mari Ostendorf, Patti J. Price, Stefanie Shattuck-Hufnagel, “The Boston University radio news corpus,” *Linguistic Data Consortium*, 1995.