A decision tree-based clustering approach to state definition in an excitation modeling framework for HMM-based speech synthesis

Ranniry Maia$^1$, Tomoki Toda$^{†‡}$, Keiichi Tokuda$^{††}$, Shinsuke Sakai$^1$, Satoshi Nakamura$^†$

$^†$National Institute of Information and Communications Technology, Japan
$^‡$Nara Institute of Science and Technology, Japan
$^{††}$Nagoya Institute of Technology, Japan

{ranniry.maia,shinsuke.sakai,satoshi.nakamura}@nict.go.jp
tomoki@is.naist.jp, tokuda@nitech.ac.jp

Abstract

This paper presents a decision tree-based algorithm to cluster residual segments assuming an excitation model based on state-dependent filtering of pulse train and white noise. The decision tree construction principle is the same as the one applied to speech recognition. Here parent nodes are split using the residual maximum likelihood criterion. Once these excitation decision trees are constructed for residual signals segmented by full context models, using questions related to the full context of the training sentences, they can be utilized for excitation modeling in speech synthesis based on hidden Markov models (HMM). Experimental results have shown that the algorithm in question is very effective in terms of clustering residual signals given segmentation, pitch marks and full context questions, resulting in filters with good residual modeling properties.

Index Terms: speech synthesis, HMM-based speech synthesis, decision tree-based clustering, residual modeling.

1. Introduction

Efforts have been made recently to enhance HMM-based speech synthesizers with the goal of making them able to produce close to natural speech while keeping some interesting properties, such as use of voice transformation techniques, utilization of small corpora and small footprint demand. By focusing on the synthesis engine part, some approaches have been reported to improve HMM-based TTS systems through the design of better synthesis engine parts, some approaches have been reported to improve HMM-based TTS systems through the design of better synthesis engine parts. For instance, modeling of the so-called bandpass aperiodicity parameters and eventual use of the excitation scheme reported in [5] at run-time is part component of the system described in [6]. In [2] a sinusoidal modeling approach is proposed, in [3] the Liljencrants-Fant (LF) model is applied, and utilization of glottal inverse filtering is reported in [4].

In [1] a residual modeling approach for HMM-based speech synthesis is described. The method is based on the principle of analysis-by-synthesis speech coders and consists in the optimization of state-dependent filters coefficients through the minimization of the difference between synthetic excitation and residual. Although good performance was achieved, state definition remained as an open issue. Initially, as in the experiments presented in [1], filter states were regarded as leaves of decision trees for mel-cepstral coefficients. Later, in [7], an state definition approach was reported, which consisted in merging initial clusters represented by leaves of decision trees for mel-cepstral coefficients. The merging criterion utilized in that case was maximum likelihood (ML) of residual and good results in terms of filter modeling were reported. However, since this approach still relies on the utilization of decision trees constructed for mel-cepstral coefficients, it is natural to wonder if further improvements can be achieved if the entire clustering process was based on residual likelihood. This paper presents a way to define filter states of the excitation model of [1] through an algorithm that performs tree-based clustering of residual signals.

In Section 2 the residual modeling framework of [1] is outlined. In Section 3 previous approaches for state definition are revisited, and in Section 4 the proposed algorithm for residual clustering is described. Section 5 shows some experiments and the conclusions are in Section 6.

2. The excitation modeling framework

Figure 1 depicts the excitation model described in [1], which is applied to HMM speech synthesis instead of the excitation scheme where pulse train and white noise are assigned to voiced and unvoiced segments, respectively. During the synthesis stage, shown in Figure 1(a), the voiced filter $H_v(n)$ processes the pulse train $t(n)$ and outputs the voiced excitation signal $v(n)$ which mimics the voiced portions of the residual database. The unvoiced filter $H_u(n)$, on the other hand, weights the noise sequence $w(n)$ to produce the unvoiced component $u(n)$ of the excitation signal $\ell(n)$. Voiced and unvoiced filters vary according to each HMM state position $(s_1, \ldots, s_8)$, with $S$ being the number of states covered by the sentence to be synthesized. During the training part, the residual signal $e(n)$ becomes the target of the analysis-by-synthesis block of Figure 1(b) and the filters

$$H_v(z) = \sum_{l=-M/2}^{M/2} h(l)z^{-l}, \quad (1)$$

$$H_u(z) = \frac{1}{1 - \sum_{l=1}^{K} g(l)z^{-l}}, \quad (2)$$

are iteratively optimized in the sense of minimizing the mean squared error of the system of Figure 1(b), given by

$$\varepsilon = E\{w^2(n)\} = E \left\{ \left[ g'(n) + e(n) - h(n) * t(n) \right]^2 \right\}, \quad (3)$$

where $g(l)$ and $h(l)$ are respectively the coefficients of $H_u(z)$ and $H_v(z)$, $K$ is the gain term of $H_u(z)$, and $g'(n)$ is the impulse response of $H_u(z)$. 
Figure 1: The assumed excitation modeling framework: (a) synthesis part; (b) training part.

3. State definition for the excitation model

From Figure 1 it can be noticed that filters $H_u(z)$ and $H_v(z)$ are associated with each HMM state position. The entire process spanning from state determination to excitation model training can be enumerated through the following steps:

1. create/define states for the excitation model;
2. quantize (classify) residual segments according to the defined states;
3. calculate filter for each cluster of residual segments using the procedure described in [1] to achieve the final excitation model.

This paper focuses on the first two steps enumerated above. In the next sections two existing methods to perform this task aside from the proposed algorithm are outlined.

3.1. The phonetic decision-trees method

In the phonetic decision trees method for state assignment, filter states for the excitation model, $\{s_1, \ldots, s_S\}$, with $S$ being the number of states, are regarded as terminal nodes of decision trees constructed for the spectrum stream of the HMM-based synthesizer in which the excitation model is applied to. The idea of utilizing trees for spectrum relies on the assumption that residual sequences are highly correlated with the spectral parameters from which they are derived by inverse filtering. Based on empirical approaches, the best trees are the ones constructed using solely phonetic questions. In addition, the minimum description length (MDL) factor, used to control the size of the trees, is set so as just gross phonetic information, such as voiced, unvoiced, fricative, stops, etc, can be classified by the trees.

3.2. The bottom-up clustering approach

The use of phonetic decision trees for state definition, as described in Section 3.1, presents some drawbacks. The first one consists in the fact that it is necessary to design specific phonetic questions to cluster mel-cepstral coefficient distributions, as well as some supervision to check whether the tree has achieved an appropriate size (MDL factor control). Besides, in a more general sense, the approach itself of making use of mel-cepstral coefficient likelihood to cluster residual segments is at best a rough approximation for filter state definition.

To alleviate the drawbacks of the phonetic trees method, an algorithm to merge terminal nodes of the usual decision trees for mel-cepstral coefficients using ML of residual segments has been proposed in [7]. This hybrid approach so-defined bottom-up clustering when compared with the utilization of phonetic decision-trees presents the advantages of: (1) using residual ML to obtain the final filter clusters; (2) no need to design specific question sets for clustering. In this case, likelihood increment or number of clusters can be used as stopping criterion.

4. State definition by top-down clustering

Both procedures for state definition described in Section 3 rely on the use of trees for mel-cepstral coefficients; one completely and the other partially. This section describes the proposed algorithm which is entirely based on residual ML.

4.1. Clustering criterion: residual ML

Assuming that the noise sequence $w(n)$ which is output by filter $G(z)$ in Figure 1(b) is a Gaussian process, the log likelihood of the signal $u(n)$ (also a Gaussian process) is given by:

$$\log P[u|H_u] = -\frac{N}{2} \log 2\pi + \frac{1}{2} \log |G^T G| - \frac{1}{2} u^T G^T G u,$$

where $N$ is the number of samples of the whole database and $u = [u(0) \ldots u(N-1)]^T$, $G = [g^{(0)} \ldots g^{(N-1)}]$, $g^{(m)} = \begin{bmatrix} 0 & \ldots & 0 & \frac{1}{K} & \frac{u(l)}{K} & \ldots & \frac{u(L)}{K} & 0 & \ldots & 0 \end{bmatrix}_T$. (7)

The second term in the right side of (4) can be written as:

$$\frac{1}{2} \log |G^T G| = \frac{1}{2} \sum_{n=0}^{N-1} \log \left[ 1 - \sum_{l=1}^{K} g(l)e^{jlw(n)} \right]^2 = -N \log K,$$

and because $G(z)$ is minimum-phase, the first term in the right side of (8) is zero [8]. Further, if $\tilde{w}(n) = \frac{1}{K} w(n)$ is a white noise sequence with variance one and mean zero, and $N \gg L$, the third term in the right side of (4) can be approximated as:

$$\frac{1}{2} u^T G^T G u = -\frac{1}{2} K^2 (N + L) E\{\tilde{w}^2(n)\} \approx -\frac{1}{2} K^2 N.$$

Therefore, the likelihood of $e(n)$ given the excitation model is simply a function of the unvoiced filter gain component $K$:

$$\log P[e|H_u, H_v, t] = -\frac{N}{2} \log 2\pi - N \left( \log K + \frac{K^2}{2} \right).$$

4.2. Clustering procedure

By taking into account the state-dependency of the filter coefficients, (10) can be re-written as:

$$\log P[e|H_u, H_v, t] = -\frac{N}{2} \log 2\pi + \sum_{j=1}^{S} \mathcal{L}_j,$$

Throughout this paper bold upper and lower case letters represent matrices and vectors, respectively, and $[\cdot]^T$ means transposition.

Note that $P[u(n)|H_u(z)] \leftrightarrow P[e(n)|H_v(z), H_u(z), t(n)]$. 

1784
where

\[ L_j = -N_j \left( \log K_j + \frac{K_j^2}{2} \right), \]  

(12)
is the likelihood of \( e(n) \) under state \( s_j \), \( N_j \) is its corresponding number of samples, \( K_j \) is the corresponding unvoiced filter gain, and \( S \) is the number of states (or clusters, assuming that we are dealing with tied states). From (12) one can see that the smaller the gain factor \( K_j \) is, the larger will be the contribution of cluster \( s_j \) to the overall likelihood, weighted by the number of samples of the cluster. In fact, considering voiced regions, a small \( K_j \) means that the power of the unvoiced excitation \( u(n) = e(n) - v(n) \) of segments belonging to cluster \( s_j \) is small, which implies that the \( H_e(z) \) outputs a signal \( v(n) \) which is close to the target \( e(n) \) in Figure 1(b).

To visualize the way to calculate \( L_j \) it is necessary to consider the block diagram of Figure 1(b). Initially voiced filter coefficients are computed, followed by the determination of the unvoiced excitation component \( u(n) \), finally leading to the gain component \( K_j \). The process of splitting one cluster into two can thus be sketched as follows:

1. split \( s_j \) into \( s_{j1} \) and \( s_{j2} \) given a candidate question;
2. calculate voiced filter coefficients, \( h_{j1} \) and \( h_{j2} \), for the new clusters \( s_{j1} \) and \( s_{j2} \), respectively;
3. compute unvoiced filter coefficients with corresponding gain components, \( g_{j1} , K_{j1} , g_{j2} , K_{j2} \), respectively for \( s_{j1} \) and \( s_{j2} \).

After calculating \( L_{j1} \) and \( L_{j2} \) from \( K_{j1} \) and \( K_{j2} \), respectively, using (12), likelihood increment due to the split can be given by

\[ L_{inc} = L_{after} - L_{before} = L_{j1} + L_{j2} - L_{j}. \] (13)

4.3. Approximations to decrease computational complexity

The determination of voiced filters and unvoiced gain components for \( s_{jx} | x = 1, 2 \) implies optimization of filter coefficients and pulse trains for the new clusters, according to the algorithm described in [1]. In order to decrease computational complexity this iterative optimization is replaced by single calculation of voiced filters followed by linear prediction analysis of the unvoiced excitation signal \( u(n) \) under segments belonging to \( s_{jx} \) to derive the gain component \( K_{jx} \).

Assuming the diagram of Figure 1(b), voiced filter coefficients for cluster \( s_{jx} \) can be obtained by using the least squares formulation, i.e.,

\[ h_{jx} = \left( \sum_{i \in s_{jx}} A_i^T A_i \right)^{-1} \sum_{i \in s_{jx}} A_i^T e_i, \] (14)

where

\[ A_i = \begin{bmatrix} t_i^{(0)} & \cdots & t_i^{(M)} \end{bmatrix}, \] (15)

\[ \hat{t}_i^{(m)} = \begin{bmatrix} 0 \cdots 0 & t_i^{(N_i - 1)} & 0 \cdots 0 \end{bmatrix}^T, \] (16)

\[ e_i = \begin{bmatrix} 0 \cdots 0 & e_i^{(N_i - 1)} & 0 \cdots 0 \end{bmatrix}^T, \] (17)

with \( t_i(n) \) and \( e_i(n) \) being respectively pulse train and residual segments with \( N_i \) samples belonging to cluster \( s_{jx} \). Segments are obtained by alignment performed at the HMM state level.

After that, the gain \( K_{jx} \) is calculated from

\[ K_{jx} = r_{jx}(0) - \sum_{l=1}^{L} g_{jx}(l) r_{jx}(l), \] (18)

Table 1: Iterative algorithm to cluster one HMM state position for the excitation model shown in Figure 1.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>For each cluster ( s_j \in { s_1, \ldots, s_S } ) and each question ( q_i \in { q_1, \ldots, q_Q } ) (( Q ) is the number of questions)</td>
</tr>
<tr>
<td>1.1)</td>
<td>Split ( s_j ) into ( s_{j1} ) and ( s_{j2} ) according to question ( q_i )</td>
</tr>
<tr>
<td>1.2)</td>
<td>Calculate voiced filters ( h_{j1} ) and ( h_{j2} ) using (14)</td>
</tr>
<tr>
<td>1.3)</td>
<td>Calculate unvoiced filter gain components ( K_{j1} ) and ( K_{j2} ) from (18)</td>
</tr>
<tr>
<td>1.4)</td>
<td>Calculate ( L_{j1} ) and ( L_{j2} ) according to (12)</td>
</tr>
<tr>
<td>2)</td>
<td>Select the cluster ( s_j' ) and question ( q_i' ) that result in the largest likelihood increment given by (13)</td>
</tr>
<tr>
<td>3)</td>
<td>Make ( { s_1, \ldots, s_j', \ldots, s_S } \rightarrow { s_1, \ldots, s_{j1}', s_{j2}', \ldots, s_{S+1} }, { s_1', \ldots, s_{j1}' \rightarrow { s_1', \ldots, s_{j1}', s_{j2}', \ldots, q_{Q-1} }</td>
</tr>
<tr>
<td>4)</td>
<td>If stopping criterion is not fulfilled go to Step 1</td>
</tr>
<tr>
<td>5)</td>
<td>Stop</td>
</tr>
</tbody>
</table>

with

\[ r_{jx}(l) = \sum_{n=0}^{N_i-1} u_i(n) u_i(n-l), \quad l = 0, \ldots, L \] (19)

being the sum of autocorrelation sequences of all segments of \( u_i(n) = e_i(n) - h_{jx}(n) * r_{jx}(n) \), where \( i \in s_{jx} \). The unvoiced filter coefficients of cluster \( s_{jx} \), \( \{ g_{jx}(1), \ldots, g_{jx}(L) \} \), are determined from \( r_{jx}(l) \) by linear prediction using the Levinson-Durbin algorithm [8].

4.4. Algorithm for decision tree construction

Decision tree construction for one HMM state position starts by grouping all residual segments into a single cluster \( s_1 (S = 1) \). After that, split iterations are carried out as shown in Table 1.

4.5. Stopping criterion

Stopping criterion can be set to a minimum likelihood increment threshold. However, the same criterion usually employed in the clustering process during the training of HMM-based synthesizers, namely the MDL criterion, can also be applied here. The advantage in this case is that the size of the trees can be systematically controlled based on the trade-off between likelihood increment and model complexity [9].

Let the description length of excitation model \( \lambda_i \) with cluster set \( \{ s_1, \ldots, s_S \} \), where \( S \) is the number of clusters, each one of them having a voiced and an unvoiced filter, be given by

\[ \ell_i = \frac{N_i}{2} \log 2 \pi - \sum_{j=1}^{S} L_j + \frac{(M + L + 2)S}{2} \log N. \] (20)

The first and second terms in the right side of (20) correspond to the likelihood of \( \lambda_i \), whereas the third term measures its complexity [9]. After each split step, the difference of description length between the model after the split, \( \lambda_{i+1} \), and the model before the split, \( \lambda_i \), is

\[ \Delta \ell = \ell_{i+1} - \ell_i = -L_{inc} + \frac{M + L + 2}{2} \log N. \] (21)

The clustering process is stopped if \( \Delta \ell > 0 \).

5. Experiments

The ATR503 database recorded by a female Japanese speaker was utilized to test the proposed clustering algorithm. An HMM-based synthesizer was trained for this database. Aside from \( F_0 \), mel-cepstral coefficients as described in [6] formed the multi-stream observation vectors.
Table 2: Number of terminal nodes at the end of the clustering process. Number of full context labels is 29395.

<table>
<thead>
<tr>
<th>HMM state</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. terminal nodes</td>
<td>14</td>
<td>17</td>
<td>37</td>
<td>16</td>
<td>13</td>
</tr>
</tbody>
</table>

Figure 2: $L_{inc}$ for each split iteration for all HMM states.

5.1. Applying the proposed algorithm to the speech data

5.1.1. Conditions

Residual signals were extracted from the speech database by inverse filtering using the same mel-cepstral coefficients employed to train the HMM synthesizer. Full context models were used to segment the residual signals at the HMM state level. Pulse trains were derived from pitch marks and eventually optimized for the residual segments through the procedure described in [1]. Residual segments were then clustered using the algorithm described in Table 1. The MDL criterion as described in Section 4.5 was used to stop tree growth.

5.1.2. Results

Figure 2 shows the evolution of $L_{inc}$ given by (13) for all HMM state positions along the split steps. Table 2 displays the number of terminal nodes at the end of the process. A total of $S = 97$ clusters were created. It can be noticed that the central states are the ones clustered with more details. This property was unexpected since the central states include more segments. Figure 3 depicts the decision tree constructed for the first HMM state. The interesting point to emphasize here is that although a question set related to full context features was used by the clustering algorithm, questions related to current phone and its left context were the ones mostly selected during the process. The same result was also observed for the other states. This perhaps might explain why the phonetic trees approach of Section 3.1 has been very effective for residual modeling. One can also see in Figure 3 that the voiced branch is clustered with more details. This property was more clear for the central states. This larger granularity of the decision trees for voiced sounds was also expected.

5.2. Excitation modeling

The state configuration created by the clustering process was utilized to train an excitation model for the HMM-based synthesizer. Voiced and unvoiced filters were determined for each terminal node of the constructed trees through the algorithm described in [1]. Figure 4 shows the impulse responses of the final voice filters. It can be noticed that convergence was achieved for most of the filters. This shows that the proposed algorithm was successful in clustering residual segments based on the context determined by the questions. On the other hand, a few visible samples of noisy impulse responses in Figure 4 infer that bad segmentation and/or pitch marking mistakes may have contributed to that. However, segmentation and pitch marking issues lie beyond the scope of this paper.

Figure 3: Decision tree generated for the first HMM state. The terms “C-” and “L-” mean respectively current and left context. Terminal nodes are represented by the yellow rectangles.

Figure 4: Impulse responses of filters $H_n(\mathbf{z})$ derived using the state configuration yielded by the proposed algorithm.

6. Conclusion

The proposed algorithm has shown to be effective for clustering residual signals under the assumed excitation model, eliminating the rough approximation that has been applied so far with the use of trees for mel-cepstral coefficients. Although filters are computed during the clustering process, the algorithm works just as a state definer. An unified method for state clustering and filter optimization is in future plans.

7. References