Application of the Trended Hidden Markov Model to Speech Synthesis

John Dines, Sridha Sridharan and Miles Moody

Speech Research Laboratory, RCSAVT
School of Electrical and Electronic Systems Engineering
Queensland University of Technology
GPO Box 2434, Brisbane QLD 4001, Australia
j.dines@qut.edu.au s.sridharan@qut.edu.au m.moody@qut.edu.au

Abstract

This paper presents our work on a speech synthesis system that utilises the trended Hidden Markov Model to represent the basic synthesis unit. We draw upon both speech recognition and speech synthesis research to develop a system that is able to synthesise intelligible and natural sounding speech. Acoustic units are clustered using the decision tree technique and speech data corresponding to these clusters is used for the training of trended Hidden Markov Model synthesis units. The overall system has been implemented in a PSOLA synthesiser and the resultant speech has been compared with that produced by a conventional diphone synthesiser to yield very encouraging results.

1. Introduction

Speech synthesis using the Hidden Markov Model (HMM) has been a topic of interest to researchers for some time and recent research has shown that relatively high quality synthesis may be achieved with the Hidden Markov Model framework [1]. The Hidden Markov Model framework provides a means of synthesising speech directly from the same statistical models used to perform speech recognition tasks. This makes the HMM inherently suited to the trainable speech synthesis problem which requires us to generate a transcription of a speech database then construct an acoustic inventory from the transcribed database. By using the Hidden Markov Model for speech synthesis we are able to bundle these two tasks into a single training procedure.

Much of the HMM speech synthesis research to date has concentrated on smoothing discontinuities at state boundaries by using the HMM’s dynamic features, but the dynamics of the parameters within the center of the state remain relatively stationary. By using the trended Hidden Markov Model framework, not only are we able to control the dynamics of the speech at the state boundaries, but also, the dynamic behavior within the states may be reconstructed using the data generative features of the trend function that govern the underlying time-varying behaviour of the model. This is especially important for longer sonorant segments of speech and all plosives.

The trended Hidden Markov Model is a generalisation of the standard Hidden Markov Model whereby each state is represented by a time varying function plus a stationary residual (and the standard HMM is the degenerate case in which the trend function is of 0th order). Early work on the trended Hidden Markov Model [2, 3] concentrated on formulation and algorithm development for parameter estimation and Viterbi decoding. This work demonstrated the superiority of the trended HMM over the stationary HMM in both recognition and feature reconstruction. We have used this research as a basis for our own research on the use of the trended HMM as a fundamental unit for speech synthesis.

In this paper we expand upon our earlier work [4] on the use of trended Hidden Markov Models in speech synthesis. We demonstrated that the trended HMM is better able to produce intelligible speech than the stationary HMM by comparing the results from Modified Rhyme Tests (MRT). Our new research has been focused on improving synthesis quality by refining the training procedures and clustering techniques that were used earlier.

This paper is organised as follows: The overall system design is detailed in Section 2. Particular detail is given to describing the variable rate coding and clustering schemes in Sections 2.1 and 2.2 respectively. Section 3 discusses the mathematical formulation of the trended Hidden Markov Model and the algorithms for parameter estimation and synthesis. Sections 4 and 5 give some results and brief conclusions.

2. System Design

There are several stages to the creation of a voice using trainable speech synthesis techniques. First of all a phonetically transcribed single speaker database is required. The US KED TIMIT single speaker database was used for the experiments in this paper. The database includes hand corrected phone level transcriptions and is freely available via the Festvox Project [5]. The database comprises 453 phonetically balanced sentences at 16 kHz which equates to approximately 22 minutes of speech.

Segmentation of the speech database was carried out using a Hidden Markov Model alignment tool based on HTK. The speech analysis comprised a 24th order mel-cepstral parameterisation [6] plus 0th order coefficient, delta and delta^2 terms. This was used to train 42 left-right, single-mixture, continuous density monophone HMMs including a silence model. All models comprised three states, except the bursts (which had one state with skip transition), closures (merged into voiced and unvoiced one state models), fricatives (which only had one state) and the affricates (which had two states). It was found that separate modelling of closures and stops was important as in some instances stops were unreleased or closures were not present. The silence model comprised seven states. The parameterisa-
tion of speech data for both the alignment and synthesis stages of processing was based on the variable rate coding scheme described in [7] and is briefly described in Section 2.1.

The second stage of the training process involves the clustering of context dependent phonetic units. This serves two purposes; it reduces the number of units to be stored and it also prevents over training of synthesis units, which has been observed to be significantly detrimental to synthesis quality. The monophones were cloned into triphones and the decision tree clustering algorithm was used to cluster units after embedded re-estimation of the context dependent models. This process is described in further detail in Section 2.2.

The last step is the parameter estimation stage, in which the trended HMMs are trained from the database segments and the state duration statistics are retained for re-synthesis purposes. Models were trained from both a 24th order mel-cepstral analysis and a 16th order Bark-scaled Line Spectral Frequency (LSF) analysis for comparative purposes. All models had three states except for stops, fricatives and closures which had only one state and affricates which had two states.

Speech synthesis is based upon a pitch excited LPC framework using the OGIresLPC PSOLA plug-in for the Festival Text-To-Speech Synthesis System [8, 9]. Synthesis of the spectral features is achieved by concatenating the sequence of trended HMMs and using the model statistics of the sentence HMM to reconstruct the feature vectors. The parameter generation algorithm [1] may be applied to trended HMMs, in which the dynamic features are used to constrain the temporal evolution of the spectral features generated from the concatenated models. This has been shown to significantly reduce spectral discontinuities at state boundaries and has been used in our work to great effect.

Phone durations are determined from the mean and standard deviation of state durations obtained during training according to Eq. (1) — a scaling factor of 0.1 was used for sentences and 0.5 for isolated words [7]. This method had been observed to give fairly natural sounding speech, reproduced in the style in which the speech database was originally recorded. The pitch is set at a constant value throughout synthesis at the average of the training speaker (which was approximately 110 Hz for the KED database).

\[
\text{synthesis duration} = \text{average duration} + \text{scaling factor} \times \text{std. dev. of duration} \quad (1)
\]

2.1. Variable Rate Coding

Parameterisation of speech data is carried out using frame-size and frame-shift determined by the class of the phoneme spanning the speech segment according to the following rules:

- \(25/60\) coding for vowels, glides and other voiced phones;
- \(6/4\) coding for fricatives, affricates and silence; and
- \(6/2\) coding for bursts and closures

where \(L/\Delta L\) refers to the frame-size and the frame-shift respectively.

By using a variable rate coding scheme, we are able to make a compromise between two opposing optimisation criteria: more accurate location of phone boundaries by smaller frame shift and more accurate estimation of feature parameters by using a longer analysis window. For shorter phones such as bursts, accurate estimation of the spectral feature parameters is not as essential (or even possible given their short duration), hence we assign these segments a smaller frame size and shift, while for voiced sounds, such as vowels, accurate estimation of the spectral feature parameters is important, hence we retain the \(25/6\) coding scheme.

2.2. Clustering of Acoustic Units

Clustering serves to reduce the number of synthesis models that need to be trained and stored and also prevents over-training of a triphone model for which there are insufficient training examples. We use the decision tree technique [10] which also has the advantage of being able to synthesise phones in contexts that were not seen during training, especially important under conditions of limited training data.

The decision tree clustering algorithm functions by pooling all of the available context dependent models for a single phone into a single root node and splitting the node using a series of questions pertaining to the context of the model. The split is performed for the question which gives the highest increase in log-likelihood. Typically, in speech recognition applications, the process is repeated on each leaf node until the increase in log-likelihood drops below a certain threshold or the number of training observations pooled at the leaf node drops below a preset level. Clustering and tying of models is usually carried out at the state level.

In our implementation of the clustering procedure, there are two major diversions from the standard implementation [10, 11]. First of all, we use a single stopping criterion which is a lower bound on the number of training segments that must be present at each leaf node of the tree (an occupation count of eight was set in our experiments). The minimum change in log-likelihood is set to zero. This has been demonstrated to be an effective technique, despite the possibility that over-training may be observed when synthesising unseen contexts [7].

The second major difference in the implementation used in our work is that the clustering was performed at the model level, rather than the state level. This is because the trended Hidden Markov Model framework does not yet enable embedded estimation of sub-word model parameters, which is a necessary condition for the training of state-tied Hidden Markov Models.

3. Trended Hidden Markov Models

3.1. Model Formulation

The trended Hidden Markov Model provides a means of overcoming the inherent disadvantage of the standard Hidden Markov Model, which relies on the assumption that the speech segments characterised by each state are Independently and Identically Distributed (IID). The formulation of the trended HMM represents each state by a time varying trend function plus a stationary residual, Eq. (2), thus relaxing the assumption of IID.

\[
O_t = \sum_{m=0}^{M} B_t(m) f_m \left( \frac{t - \tau}{T_t} \right) + R_t(\Sigma_t) \quad (2)
\]

where the left-hand term is the state-dependent polynomial regression function of order \(M\), with \(B_t(m)\) as the polynomial.
coefficients for state $i$, and $f_{m}(\frac{t-n}{T})$ the $m$th order polynomial trend function. The right-hand term is the residual, $R_k$, with covariance, $\Sigma_t$, $\eta_i$ and $T_i$ are auxiliary parameters which normalise the duration over which the polynomial is evaluated to $[0, \ldots, 1]$ where $T_i$ is the unnormalised sojourn time of state $i$.

3.2. Parameter Estimation

We have used the Legendre family of orthogonal polynomials to estimate the polynomial coefficients $B(m)$. Model parameters are estimated via a two-stage process involving a segmentation step followed by a maximisation step, similar to the K-means algorithm used to initialise standard HMMs. The segmentation step involves a modified version of the Viterbi algorithm and the maximisation step uses multi-variate linear regression to find the best fit between the trend functions and training data (for an in depth analysis see [2, 3]).

A problem present in the formulation of the trended HMM maximisation step in [3] became apparent during early synthesis experiments. When attempting to fit a multiple state model to multiple training examples, it was found that in some cases each state may tend to behave as an independent mixture. This may be because there were too many states allocated to model that particular speech segment and as a result a form of overtraining was occurring.

It was difficult to predict which models would behave in this fashion a priori so that we could reduce the number of states for that model, so a free shift parameter, $S$, was introduced into the linear regression, Eq. (3). This parameter enables the trend functions to be fitted to the shape of the speech features that they are attempting to model, irrespective of the variations in the mean of the spectral values between training examples. As it stands, the solution to the system of equations in Eq. (3) is infinite, hence, we impose an additional constraint that $\sum_{k=1}^{K} S_k^d = 0$, $d = 1, 2, \ldots, D$. This is equivalent to normalising the mean of each of the training segments to the global mean of all the segments, Eq. (4).

$$X'X \left[ B^{(1)} | B^{(2)} \ldots | B^{(D)} \right] =
X' \left[ O^{(1)} - S^{(1)} | O^{(2)} - S^{(2)} | \ldots | O^{(D)} - S^{(D)} \right]$$

where

$$O^{(d)} = [O_{1,\ldots,\ldots,T_d} | O_{2,\ldots,\ldots,T_d} | \ldots | O_{K,\ldots,\ldots,T_d}]$$

$$S^{(d)} = [S_1^{(d)} | S_2^{(d)} | \ldots | S_K^{(d)}]$$

$$S_k = \frac{\sum_{t=1}^{T_k} O_{k,t}^{(d)}}{T_k} - \frac{\sum_{t=1}^{T_k} \sum_{k=1}^{K} O_{k,t}^{(d)}}{K}$$

$$X = [X_1 | X_2 | \ldots | X_K]$$

and where $B^{(d)}$, $d = 1, 2, \ldots, D$ are the polynomial coefficients, up to $M$th order, for only the $d$th component in the multivariate analysis. The $d$th components of the observations from the training segments, $k = 1, 2, \ldots, K$, at time $t = 1, 2, \ldots, T_k$ are $O_{k,t}^{(d)}$. The free shift parameter for the $d$th component of the $k$th training segment (and is a vector of length $T_k$) is $S_k^{(d)}$. The $m$th order Legendre polynomials valued at time $t$ are $f_m(t)$.

3.3. Parameter Generation

Synthesis of the spectral features is achieved by concatenating the trended HMMs for the phonetic sequence to be synthesised. We use the algorithm proposed in [1] which uses the dynamic features of the model statistics to constrain the temporal evolution of the spectral features generated from the concatenated models. Under the constraint of Eq. (5) the set of feature vectors is determined by the set of linear equations:

$$\partial \log P(O|\lambda) / \partial c_t = 0$$

In our implementation of the algorithm, we only use the delta features and only one iteration of the algorithm is performed, using the trend functions from the concatenated synthesis models, Eq (6), as our initial estimate of both feature and delta-feature vectors. A sliding window implementation (window size equals 30 frames) is also undertaken in order to facilitate more rapid computation of spectral features with little degradation to the generated parameters.

$$\Delta c_t = \sum_{\alpha=-L}^{L} \omega(\alpha) c_{t+\alpha}$$

$$c_t' = \sum_{m=0}^{M} B_{i}(m) f_m \left( \frac{t-n}{T} \right)$$

where $\Delta c_t$ is the delta-feature calculated from features $q_{i\alpha}$ at time $t$. $2L$ is the window size over which the value is calculated and $\omega(\alpha)$ is the windowing function. The vector $c_t'$ is our estimate of the reconstructed features from the trend function inherent to our model.

4. Results

An initial comparison of speech synthesised using the LSF and mel-cepstral parameterisation techniques revealed some interesting results. While the LSF models tended to yield better spectral peaks and valleys than the mel-cepstral technique, it was apparent that the modelling process was causing some significant warping of the formant locations of the LSF models, resulting in poor sounding vowel segments in some cases. In general, though, the speech synthesised using either parameterisation was much the same, hence it was decided to proceed with our system assessment for only the mel-cepstral trained models.

System assessment comprised Modified Rhyme Tests (MRT). The mel-cepstral trained system was compared with the KED diphone voice which is released as part of the Festival Text-To-Speech Synthesis System [8]. The diphone voice was synthesised using the same constraints as our trained voice (monotone, default duration statistics). The test comprised 25 words for the KED-HMM and KED-Diphone voices, using six untrained listeners. The results (word error rate) are shown in Table 1.
It is apparent from the results that the trended HMM speech synthesis technique still does not yet perform at the same level of intelligibility as a diphone system, but these results (20.7% error) are greatly improved over our earlier results (29.6% error) published in [4]. The variable rate coding scheme has addressed much of the problems encountered when synthesising shorter sounds, such as bursts.

Explanation for many of the errors that occurred may also be linked back to the speech database that was used for training. Lack of training data (only 22 minutes of speech — approximately an hour of speech was used for each voice in [7]) and the speaking style resulted in data sparsity problems, particularly for word initial and word final consonants (upon which MRT relies to assess synthesis quality). Further system development will require a larger database recorded in a more appropriate speaking style in order to obtain better MRT results.

Informal listening tests were also carried out for the synthesis of complete sentences. These sentences had a stylised pitch contour and the resultant speech sounded very natural and like that of the training speaker (in speaking style and sound quality). It was observed that the free-shift parameter introduced to the parameter estimation algorithm (see Section 3.2) has had a significant impact on improving the quality of synthesised voiced sounds.

5. Conclusions

We have presented a trainable speech synthesis system which uses the trended Hidden Markov Model as its fundamental synthesis unit. The intelligibility of the system is approaching that of a diphone system and has the advantage that voices may be constructed from an arbitrary database of continuous speech in an entirely unsupervised and automated training process, while diphone voices must be carefully constructed from a database of nonsense speech segments, requiring an investment of much time and effort. Modification to the trended HMM parameter estimation algorithm and the adoption of a variable rate coding scheme have significantly improved the synthesiser quality compared with our earlier work.

It has been observed that a large proportion of recognition errors were due to the simple binary excitation scheme causing some burst sounds to sound similar. It is expected that the incorporation of mixed excitation and pitch into the synthesis unit modelling will greatly reduce these errors and will also improve overall naturalness.

In addition to incorporating the modelling of excitation parameters we plan to investigate the use of multiple mixtures and will also begin testing the system with more training speech which will be recorded using our own facilities. The final system will be incorporated into the Festival Text-To-Speech Synthesis System with the addition of our speech synthesis modules.

6. Acknowledgements

This work was partially supported by the CSIRO Department of Telecommunications and Industrial Physics.

Experiments were carried out using a TD-PSOLA synthesiser based upon the Festival Text-To-Speech Synthesis System [8] and OGiresLPC plug-in residual excited LPC synthesiser [9]. The speech database used was KED TIMIT which is publicly available via the Festvox Project [5].

7. References


