# ROBUST SPLICING COSTS AND EFFICIENT SEARCH WITH BMM MODELS FOR CONCATENATIVE SPEECH SYNTHESIS

*Ivan Bulyko, Mari Ostendorf and Jeff Bilmes*

University of Washington
Department of Electrical Engineering
Seattle, WA 98195. USA
{bulyko,mo,bilmes}@ssli.ee.washington.edu

**ABSTRACT**

With the growing popularity of corpus-based methods for concatenative speech synthesis, a large amount of interest has been placed on borrowing techniques from the ASR community. This paper explores the applications of Buried Markov Models (BMM) to speech synthesis. We show that BMMs are more efficient than HMMs as a synthesis model, and focus on using BMM dependencies for computing splicing costs. We also show how the computational complexity of the dynamic search can be significantly reduced by constraining the splicing points with a negligible loss in synthesis quality.

## 1. INTRODUCTION

Recently, a growing amount of attention in speech synthesis research has been drawn toward unit selection methods, which use dynamic programming to search for speech segments in a database that minimize some cost function [1, 2, 3]. The cost function is designed to quantify distortion introduced when selected units are modified and concatenated. Typically there are two components to the unit selection cost function: the *target cost*, which is an estimate of distance between the database unit and the target, and the *concatenation cost*, which is an estimate of the distortion associated with concatenating units that were not originally spoken in sequence. Target and concatenation costs have mostly focused on segmental distortion, and have included linguistically motivated distances based on phonetic categories [4] and/or spectral distances [3].

Unrestricted TTS demands the use of sub-word units. Smaller units give more flexibility, but result in a larger unit inventory that requires more computation to be searched. One approach to reduce the computational complexity is to prune the unit database [5]. Alternatively, since computing concatenation costs is the slowest operation, one can precompute and cache concatenation costs between the most frequently used pairs of units [6], or vector-quantize the space of units and store a complete distance table between groups of units [3, 7]. In our earlier work [7], we introduced *splicing costs* as a measure of the potential discontinuity that a given unit may incur when a splice is made at its boundary irrespective of the adjoining unit. This leads to further computation reduction, since the search tree can be pruned based on the splicing costs prior to evaluating all possible concatenations. Perceptual experiments show that splicing costs also help achieve smoother concatenations.

The recent focus on corpus-based methods in speech synthesis has encouraged researchers to adapt techniques, such as decision tree clustering [8] and Hidden Markov Models (HMM) [9, 10, 11], that are commonly used in speech recognition. In this paper we explore the benefits of using Buried Markov Models (BMM) [12] in speech synthesis. In particular, we propose a new method for computing splicing costs that takes the predictability of successive speech frames into account. We also investigate the potential for reducing the cost of the unit selection search by restricting the set of boundaries where a splice is allowed.

The rest of the paper is organized as follows. In Section 2 we provide some details about the modeling assumptions made by the BMM and how the structure of the dependencies differs with the type of application (i.e. synthesis vs. recognition). Section 3 explains how BMMs can be applied to concatenative synthesis. Experiments are described in Section 4, and we conclude with a summary of the key results in Section 5.

## 2. MODELING SPECTRAL DYNAMICS IN SYNTHESIS WITH BMMS

Buried Markov models [12], a form of graphical model [13], augment the dependency structure relative to that of an HMM. In a BMM, each element of a feature vector may include direct dependencies on elements of feature vectors in addition to dependencies already included in an HMM (namely, the hidden state variable and possibly other elements of that same vector). These dependences may switch depending on the current hidden state value. Specifically, if

$X_t^i$ is the $i^{th}$ element of the $t^{th}$ feature vector, then a BMM uses the distribution $p(X_t^i|Q_t = q, Z_t^i(q))$, where $Q_t$ is the hidden state at time $t$ and $Z_t^i(q)$ is a $q$-dependent subset of feature vectors either before, at, or after time $t$. In this paper, all BMM dependencies are linear and distributions are unimodal Gaussian, in which case

$$p(X_t|Q_t = q, Z_t(q)) \sim N(X_t; B_q Z_t(q) + \mu_q, \Sigma_q),$$

where $B_q$ is a sparse matrix.

Other than issues of training and dependency representation, one of the main challenges in producing a BMM system is in choosing the structure of the dependencies for each state $q$. Choosing all dependencies leads to an enormous free parameter increase, could lead to over-training, and is probably unnecessary. The goal for structure learning is to choose that minimal set of dependencies which are most appropriate for the task at hand [12]. For automatic speech recognition (ASR), dependencies should be chosen discriminatively, as ASR is inherently a problem of pattern classification. For speech synthesis, however, dependencies should be chosen not so much for their discriminative but rather for their predictive ability — if a BMM can be "predictively structured" so that they predict $X_t$ well given past acoustic vectors, both synthesis quality and quality assessment could improve. A good measure of the predictive ability between two random variables is standard mutual information [14] which we investigate in this paper.

In order to better understand the relationship between discriminatively vs. predictively structured BMMs, we performed two informal listening experiments using the synthesis algorithm described in [10]. The first experiment compared speech synthesized from MFCCs that had been randomly sampled from: 1) an HMM, 2) a discriminatively structured BMM (DBMM), and 3) a predictively structured BMM (PBMM). It has been shown in the past that a DBMM can lead to improved ASR results. We predicted that a DBMM would not outperform an HMM for synthesizing speech, but that a PBMM would outperform both. Our hypothesis is based on the fact that the task of ASR is to extract the word sequence for a computer (i.e. intelligibility is the only concern), which is different from synthesis which involves presenting speech to a human listener for whom naturalness is also important. A model optimized for one task could be ill-suited for the other. The results of informal listening experiments supported this hypothesis; samples based on Radio news data are available at [15].

A second informal listening experiment compared the quality of HMM-synthesized speech when MFCCs included delta coefficients with that of PBMMs that did not include deltas. The synthesis algorithm utilized delta-coefficients to better smooth the final speech signal [10]. The results of the experiment showed that there was little if any difference in synthesis quality, but the PBMMs used 25 percent fewer parameters. Even though the overall quality of speech output was not good enough for general user acceptance, these listening tests led us to believe that BMM models are at least as good in capturing spectral dynamics as HMMs, and hence it may be useful in concatenative speech synthesis.

## 3. BMMS IN CONCATENATIVE SYNTHESIS

In applying BMMs to the process of unit selection in speech synthesis, we look specifically at concatenation points. In particular, we introduce a new method for computing splicing costs and assess the potential for reducing search complexity by constraining the set of possible splicing points.

### 3.1. Splicing Costs

As mentioned earlier, splicing costs measure the potential discontinuity that a given unit may incur when a splice is made at its boundary. In [7] we proposed a splicing cost that is inversely related to the spectral change at a given boundary. In other words, unit boundaries where the spectral characteristics in the original recording are changing rapidly are potentially good splicing points. A simple measure of rapid change was investigated: the Mahalanobis distance between successive vectors of spectral features at the splice point, using the grand covariance in the distance. However, there is evidence of context dependency in the perceptual significance of rate of spectral change at a unit boundary [16], where a context-dependent covariance (and added mean term) leads to an improved concatenation cost. The cost proposed in [16] is effectively the probability of the prediction residual using the simple predictor $\hat{X}_{t+1} = X_t + \mu_q$, where $q$ represents the context (which could be described by an HMM state index). Here, we extend the notion using the BMM, which involves a more general linear predictor of the form $\hat{X}_{t+1} = B_q Z_t(q) + \mu_q$, where $Z_t(q)$ may include $X_t$ as well as elements from other time vectors. Our hypothesis is that if, given the phonetic context and the spectrum on one side of the boundary, we are able to accurately predict the spectrum on the other side, then the concatenation cost should be low. Interpreting this measure for splicing costs: when the frames are very predictable, then effects of coarticulation are strong and the boundary is a poor choice for making a splice.

Our unit database contains half-phone segments. We used the Festival TTS system to cluster the units according to the decision tree clustering procedure described in [8]. Motivated by work described in [17], we used line spectral frequencies (LSF) for the parametric representation of units. Two BMM models were trained for each cluster: one with dependencies on the preceding frames (i.e. left-to-right feature dependency links), and another with the dependencies on the following frames (right-to-left). For each feature we

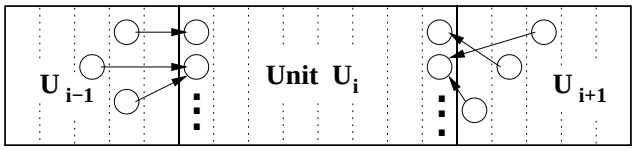


**Fig. 1**. Feature dependencies used for computing splicing costs for unit $U_i$. Dotted lines show frame boundaries of LSF vectors. Circles are individual features.

selected three links that correspond to pairs having the highest conditional mutual information (conditioned on the unit cluster identity). We searched up to ten 10 ms frames in the past (or future, depending on the type of model) to select these dependencies. The mean distance was 1.7 frames, but features as far back as 10 frames were occasionally chosen.

Each unit has left and right splicing costs that indicate suitability of a splice at its left or right boundary respectively. The *left* splicing cost for unit $U_i$ is computed by finding the inverse of the Mahalanobis distance between the first frame of LSFs in the unit and a frame predicted with the "left-to-right" model for the cluster that $U_i$ belongs to, using the data frames that precede unit $U_i$ in the original recording. This is illustrated in Fig. 1 at the left boundary of unit $U_i$. Symmetrically, we used the "right-to-left" model and the data frames following $U_i$ to predict the last frame in the unit. Then the inverse of the Mahalanobis distance between the true and the predicted last frames gives us the *right* splicing cost.

Through informal listening we established that we achieve as good or better synthesis quality compared to using the inverse of the Mahalanobis distance between two successive frames at the boundary, but differences did not appear to be significant.

### 3.2. Redefining Cutting Points

In [7] we suggested that the dynamic search at run-time can be made more efficient if the search tree is pruned based on the splicing costs prior to evaluating all possible concatenations. Another approach is to disallow a splice for entire classes of boundaries. Conditional mutual information that we used for selecting the BMM dependencies can provide us with an estimate of the degree of coarticulation at unit boundaries. Our hypothesis is that if the amount of information carried across a boundary is high then the boundary is a bad place to make a splice.

Note that even when two units are sharing the same boundary (i.e. are adjacent in the original recording) they may have different sensitivity to having a splice made at this boundary. We take this into account by treating the left and the right boundaries independently. For instance, when splicing is not allowed from the left side of the boundary

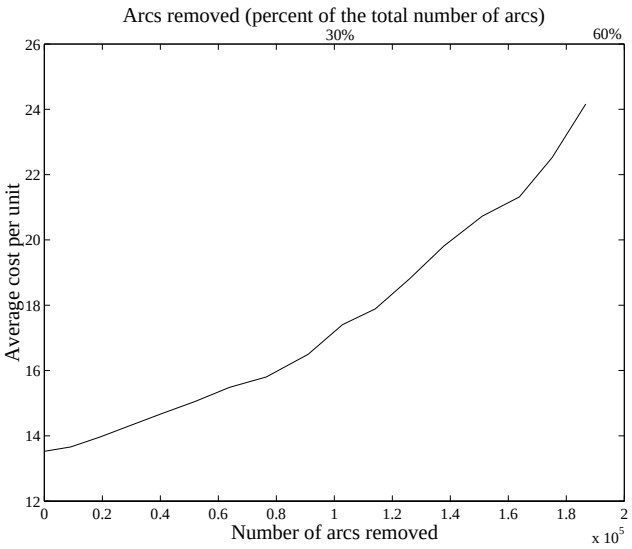


**Fig. 2**. Average cost per unit in the best path, as a function of the number of arcs (lower scale) and the percent of the total number of arcs (upper scale) removed from the unit database network.

it may still be possible to have another unit spliced on the right. For the left boundaries we compute the conditional mutual information carried across from the previous unit. Conversely, for the right boundaries we collect information that comes from the following unit. Different costs are in fact learned for the two conditions.

## 4. EXPERIMENTS

The unit database used in our experiments was extracted from the the synthesis component of a travel planning system developed at the University of Colorado [18]. The corpus contained approximately 2 hours of speech ($\approx 150,000$ half phone units) and was automatically segmented. F0 and energy were estimated automatically from the signal by means of Entropic tools. We used the weighted finite state transducer (WFST) architecture described in [7] and AT&T FSM tools to perform the unit selection. There was no spectral smoothing or prosodic modification applied to the signal. The target sentences were taken from the same domain, i.e. travel planning, but were produced by a different text generator, so while there was some overlap in the vocabularies, many of the target words were not present in the database recordings. Alternative pronunciations were also included in the target, thus making the search more flexible but more costly.

We altered the unit database WFST by gradually removing arcs that correspond to the boundaries with the largest amount of conditional mutual information carried across the

boundary. Entire classes of boundaries were eliminated at once. We kept, however, 5% of units in each cluster with the smallest splicing costs still connected to the VQ concatenation network to ensure that a path can always be found. Fig. 2 shows how the total cost of the best path (taken as an average over thirty target utterances and normalized by the number of units) changes as we remove more arcs from the database. The degradation in speech quality is graceful at first, but it rapidly becomes very noticeable after we remove about 130,000 links which is approximately 40% of all links that carry splicing costs in the unit database WFST. For examples of synthesis refer to [15].

As mentioned earlier, the goal of removing arcs from the unit database is to reduce computational cost. We observed a near linear speedup in synthesis as we removed arcs. In the case when the concatenation costs are computed at run time for each pair of candidate units, one can expect a quadratic reduction in computational complexity associated with disallowing a splice to be made at specific boundaries. While the use of multiple pronunciations in the target slowed the synthesis down to less than real time performance, we were able to double the speed and make it faster than real time by removing approximately 100,000 arcs.

## 5. DISCUSSION

We have demonstrated how context and the degree of coarticulation can be taken into account when computing splicing costs. Our approach uses prediction accuracy of BMM models as an indicator of how suitable a given boundary is for making a splice. The prediction accuracy is measured at the boundary frames, but one could also assess the prediction over a range of frames within the unit, which may give a more accurate estimate of the effects of coarticulation. In addition, one could use the BMM residual likelihood in a concatenation cost and to determine the specific join points of two units, but this would increase the computation of the concatenation cost substantially.

We also showed how the computational complexity of the dynamic search can be significantly reduced by constraining the set of boundaries where a splice is allowed. The boundary elimination procedure that we described can be used to merge certain types of units, thus creating new clusters of variable length units. For instance, starting with a unit database consisting of half phones, one can obtain a heterogeneous database of units ranging from half phones to phones, diphones or longer units, where the unit boundaries are automatically chosen according to their suitability for making a splice.

### Acknowledgments

We wish to thank Bryan Pellom and the Center for Spoken Language Research at the University of Colorado for providing the speech corpus. This material is based upon work supported by the National Science Foundation under Grant No. (IIS-9528990). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## 6. REFERENCES

[1] Hunt, A., and Black, A., "Unit selection in a concatenative speech synthesis system using a large speech database", *Proc. ICASSP*, 1:373–376, 1996.

[2] Hon, H., Acero, A., Huang, X., Liu, J., and Plumpe, M., "Automatic generation of synthesis units for trainable text-to-speech systems", *Proc. ICASSP*, 293–296, 1998.

[3] Beutnagel, M., Conkie, A., Schroeter, J., Stylianou, Y., and Syrdal, A., "The AT&T Next-Gen TTS system", *Joint Meeting of ASA, EAA, and DAGA*, 18–24, 1998.

[4] Yi, J., and Glass, J., "Natural-sounding speech synthesis using variable-length units", *Proc. ICSLP*, 1167–1170, 1998.

[5] Donovan, R., "Segment preselection in decision-tree based speech synthesis systems", *Proc. ICASSP*, 2:937–940, 2000.

[6] Beutnagel, M., Mohri, M., and Riley, M., "Rapid unit selection from a large speech corpus for concatenative speech synthesis", *Proc. Eurospeech*, 2:607–610, 1999.

[7] Bulyko, I. and Ostendorf, M., "Unit selection for speech synthesis using splicing costs with weighted finite state transducers", *Proc. Eurospeech*, 2:987-990, 2001.

[8] Black, A., and Taylor, P., "Automatically clustering similar units for unit selection in speech synthesis", *Proc. Eurospeech*, 2:601–604, 1997.

[9] Donovan, R. E. and Woodland, P. C., "Automatic speech synthesizer parameter estimation using HMMs", *Proc. ICASSP*, 640–643, 1995.

[10] Tokuda, K., et. al., "An algorithm for speech parameter generation from continuous mixture HMMs with dynamic features", *Proc. Eurospeech*, 757–760, 1995.

[11] Plumpe, M., Acero, A., Hon, H., Huang, X., "HMM-based smoothing for concatenative speech synthesis", *Proc. ICSLP*, 2751–2754, 1998.

[12] Bilmes, J., "Buried Markov Models for speech recognition", *Proc. ICASSP*, 2:713–716, 1999.

[13] Lauritzen, S. L., *Graphical Models*. Clarendon Press, Oxford. 1996.

[14] Cover, T. M. and Thomas, J. A., *Elements of Information Theory*, John Wiley & Sons, Inc., 1991.

[15] http://ssli.ee.washington.edu/projects/synth/icassp02.html

[16] Donovan, R., "A new distance measure for costing spectral discontinuities in concatenative speech synthesizers", *Proc. ESCA*, 2001.

[17] Wouters, J., and Macon, M., "Control of spectral dynamics in concatenative speech synthesis", *IEEE Trans. Speech and Audio Processing*, 9(1):30–38, 2001.

[18] Pellom, B., Ward, W., and Pradhan, S., "The CU Communicator: an architecture for dialogue systems", *Proc. ICSLP*, 2:723–726, 2000.