Feature-based Pronunciation Modeling with Trainable Asynchrony Probabilities

Karen Livescu and James Glass

MIT Computer Science and Artificial Intelligence Laboratory
Cambridge, MA 02139, USA
{klivescu,glass}@csail.mit.edu

Abstract

We report on ongoing work on a pronunciation model based on explicit representation of the evolution of multiple linguistic feature streams. In this type of model, most pronunciation variation is viewed as the result of asynchrony between features and changes in feature values. We have implemented such a model using dynamic Bayesian networks. In this paper, we extend our previous work with a mechanism for learning feature asynchrony probabilities from data. We present experimental results on a word classification task using phonetic transcriptions of utterances from the Switchboard corpus.

1. Introduction

Pronunciation variation, especially the wide variation seen in casual speech, poses a significant challenge for automatic speech recognizers [13]. Pronunciation models using phonetic substitution, insertion, and deletion rules can account for many phenomena (e.g., [8, 14]), but their success in recognition experiments has been limited and some types of variation remain difficult to represent.

Approaches to speech recognition using multiple streams of linguistic features, rather than a single stream of phones, have been proposed as one way of better handling pronunciation variation. A great deal of effort has been focused on the problem of classifying feature values from the acoustic signal (e.g., [3, 9]). There has been much less work on the relationship between linguistic features and words (i.e. the pronunciation model), with most feature-based approaches still using an essentially phone-based representation of words. By constraining the features to match phone-like units, some of the power of the feature representation may be lost. In particular, the tendency of feature streams to desynchronize in casual speech is ignored. Exceptions to this, such as [6, 10], have attempted to model more explicitly the semi-independent evolution of features.

Our work on feature-based pronunciation modeling is in the same spirit as that of [6, 10]. We aim to develop a model that is general enough to take advantage of known (or assumed) inter-feature independencies, while avoiding overly strong independence assumptions. We have been developing such a model using dynamic Bayesian networks [4], which provide flexible control over the constraints and independence assumptions in the model. The main contribution of this paper is the incorporation of a mechanism for inter-feature asynchrony modeling in which the asynchrony probabilities can be learned from data. In the following sections, we review the main features of our approach, introduce the trainable asynchrony mechanism, and present experiments on an isolated-word task using phonetic transcriptions of utterances from the Switchboard conversational speech corpus [7].

2. Approach

A feature-based pronunciation model is one that explicitly represents the evolution of multiple linguistic feature streams to generate the allowed realizations of a word and their probabilities. For concreteness, we define a "realization of a word" as a sequence of feature value vectors, one per time frame, corresponding to the surface (i.e. actual) feature values produced by a speaker.

2.1. From baseforms to surface realizations

Our model begins with the usual assumption that each word has one or more target phonemic pronunciations, or baseforms. Each baseform can be represented as a table of underlying feature values, as shown in Table 1 for the word *everybody*. Baseforms may include "unspecified" feature values ('*' in the table). More generally, each table entry is a distribution over the range of feature values.

index	0	1	2	3	...
phoneme	eh	v	r	iy	...
LIP-OPEN	wide	critical	wide	wide	...
TT-LOC	alv.	*	ret.	alv.	...
...	...	...	...	...	...

Table 1: *Part of a target pronunciation for* everybody. *In this feature set,* **LIP-OPEN** *is the lip opening degree;* **TT-LOC** *is the location along the palate to which the tongue tip is closest (*alv. = *alveolar;* ret. = *retroflex).*

Starting with a given baseform, the sequence of surface frames is generated as follows. In the first frame, all

of the features begin in their first state, corresponding to index 0 in the table (in our example, **LIP-OPEN** = 'wide', **TT-LOC** = 'alv.', and so on). In each subsequent frame, each feature can either stay in the same state or transition to the next one with some probability. If the features do not all transition at once, a situation can arise where different features are in states corresponding to different indices. This situation is what we refer to as "asynchrony". This can account for phenomena such as vowel nasalization before a nasal consonant (where the nasality feature is "ahead" of the other features). More "synchronous" configurations may be preferred, and there may be an upper bound on the allowed degree of asynchrony. The model takes this into account by assigning a cost to each configuration of feature indices according to the degree of asynchrony in that configuration (see Section 2.2).

Given the sequences of feature indices, the underlying feature values in each frame are chosen as per the baseform table. The surface feature values may differ from the underlying values, for example because of reduction phenomena (e.g. every → [eh w r iy] due to reduction in **LIP-OPEN**). The surface value S corresponding to each underlying value U is generated according to a distribution $p(S|U)$. Finally, the resulting surface feature values constitute the realization of the word.

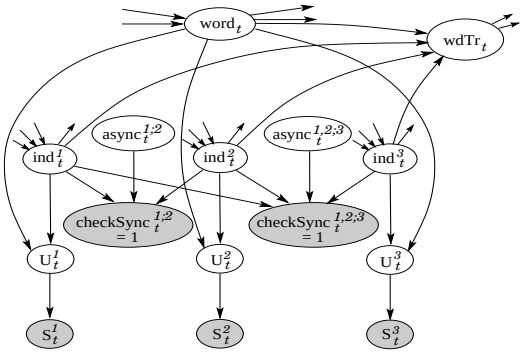

Figure 1: *One frame of a DBN for recognition with a feature-based pronunciation model. Nodes represent variables; shaded nodes are observed. Edges represent dependencies between variables. Edges without parents/children point from/to variables in adjacent frames.*

2.2. A dynamic Bayesian network implementation

A natural framework for such a model is provided by dynamic Bayesian networks (DBNs). Figure 1 shows one frame of the type of DBN used in our model (simplified somewhat for clarity of presentation). This example DBN assumes a feature set with three features, $\{1, 2, 3\}$. All of the variables are discrete, so all of the conditional distributions are probability mass functions. The variables U_t^j are the underlying feature values, and S_t^j are the surface values, at time frame t. ind_t^j are the indices of the features. $wdTr_t$ is the word transition variable: Its (binary) value indicates whether or not this is the last frame of the current word. For a more detailed description, see [12].

The variables $async_t^{A;B}$ and $checkSync_t^{A;B}$ are responsible for implementing the asynchrony constraints. We define the degree of asynchrony between two subsets A and B of the feature set as the absolute difference (rounded to the nearest integer) between the mean indices of the features in A and of the features in B. At time frame t, the degree of asynchrony between A and B is determined in the following way: A value for $async_t^{A;B}$ is drawn from an (unconditional) distribution over the integers, while $checkSync_t^{A;B}$ checks that the degree of asynchrony between A and B is in fact equal to $async_t^{A;B}$. To enforce this constraint, $checkSync_t^{A;B}$ is always observed with value 1 and its distribution is

$$\begin{aligned} P\left(checkSync_t^{A;B}{=}1 \,|\, async_t^{A;B}, ind_t^A, ind_t^B\right) &= 1 \\ \iff round\left(|mean\left(ind_t^A\right) - mean\left(ind_t^B\right)|\right) &= async_t^{A;B}, \end{aligned}$$

and 0 otherwise, where ind_t^A and ind_t^B are the sets of indices of the features in A and B, respectively. Therefore, by learning the distribution of $async_t^{A;B}$, we learn the probabilities of different degrees of feature asynchrony. The subsets A and B for each $async$ variable are, for the time being, selected manually.

The parameters of the model can be learned from data via maximum likelihood using the Expectation-Maximization (EM) algorithm [5], given observations for the word variable and surface feature variables. Surface feature observations can be obtained using collections of recorded speech with simultaneous articulatory measurements (e.g. [15]); from detailed phonetic transcriptions, which can be converted to feature transcriptions if they are sufficiently fine-grained; or perhaps by manually or semi-automatically generating feature transcriptions for a limited amount of recorded speech. Thus far, we have chosen to use the second option, in particular using the detailed phonetic transcriptions created at ICSI for a portion of the Switchboard corpus [7].

An end-to-end recognizer could be built by adding acoustic observation variables as children of the S_t^j, which would be unobserved. We have done this in the past using a much simpler pronunciation model [11]. To facilitate quick experimentation and to isolate the performance of the pronunciation model, we are testing how well we can do when given observed surface feature values. In addition, for the time being, we have assumed that all features synchronize at word boundaries. This assumption could be dropped, for example by having multiple word variables, one for each feature.

3. Experiments

In our experiments to date with this model, we have been using the following feature set, based on the vocal tract variables of articulatory phonology [2]: degree of lip opening (**LIP-OPEN**); tongue tip location and opening degree (**TT-LOC, TT-OPEN**); tongue body location and opening degree (**TB-LOC, TB-OPEN**); velum state

(**VEL**); and glottal (voicing) state (**GLOT**). We impose the following (hard) synchrony constraints: **(1)** The tongue features are completely synchronized, i.e. $ind_t^{TT-LOC} = ind_t^{TT-OPEN} = ind_t^{TB-LOC} = ind_t^{TB-OPEN}$; **(2)** The lips can desynchronize from the tongue by up to one index value, i.e. $ind_t^{LIP-OPEN} - ind_t^{TT-LOC} \leq 1$ (and equivalently for the other tongue features); and **(3)** The glottis and velum are synchronized ($ind_t^{GLOT} = ind_t^{VEL}$), and their index must be within 2 of the mean index of the tongue and lips.

We used the Graphical Models Toolkit [1] to implement the model. The distributions $p(S_t^j|U_t^j)$ were initialized with some hard constraints based on linguistic considerations, e.g. that more "constricted" underlying feature values may become less constricted on the surface, but not vice versa. The $p(U_t^j|word_t, ind_t^j)$ were derived from manually-constructed phoneme-to-feature-probability mappings. The $p(async_t^{A;B})$ were initialized with zero probabilities where dictated by constraints (1)-(3) above. The hard constraints imposed on $p(S_t^j|U_t^j)$ and $p(async_t^{A;B})$ reduce both the computational needs and the number of parameters to learn. For the non-zero values in the distributions, we compared two initializations, a "good" initialization based on linguistic considerations ($p(async_t^{A;B})$ monotonically decreases for increasing $async_t^{A:B}$, $p(S_t^j|U_t^j)$ monotonically decreases as S_t^j moves farther from U_t^j) and a "bad" initialization with uniform values for the non-zero probabilities.

We tested the performance of the model on an isolated word recognition task: Given a set of observed surface feature sequences $S_{1:T}^{\{F\}}$ (where T is the number of frames and $\{F\}$ is the feature set) corresponding to a word, the task was to determine the identity of the word from a ˜3300-word vocabulary. We segmented a portion of the ICSI transcriptions into words, and for each word, converted its phonetic transcription to a sequence of feature vectors, one vector per 10 ms frame. For this purpose, we divided diphthongs and stops into pairs of feature configurations. Given the input feature sequences, we used GMTK to compute $p(word, S_{1:T}^{\{F\}})$ for each word in a ˜3300-word vocabulary by "observing" the value of $word$ and computing the joint probability of all of the observations. The output of the recognizer is then the word that maximizes this probability. Assuming that all words in the vocabulary are equally likely, this is equivalent to maximizing the conditional probability $p(word|S_{1:T}^{\{F\}})$.

We divided a portion of the ICSI transcriptions into a ˜2900-word training set, a 165-word development set, and a 236-word test set [1]. Starting from either of the initializations described above, EM parameter learning took 6-8 iterations to converge with a 0.2% difference in the training set log probability. The development set was used mainly to tune the hard constraints on $p(S_t^j|U_t^j)$ and $p(async_t^{A;B})$. In addition, we checked the development set for word segmentation errors and corrected any that we found (such errors can occur because the phonetic and word transcriptions are not perfectly aligned). Therefore, the performance on the development set gives us an idea of how well we can do given perfect feature transcriptions and knowledge of the phenomena that occur in the data.

We experimented with two variants of the model. The "frame-based" variant corresponds to Figure 1; the "segment-based" one has the additional constraint that S^j can only change value when ind^j changes (implemented by adding parents ind_t^j, ind_{t-1}^j, S_{t-1}^j to S_t^j), and is motivated by the observation that the original frame-based model tends to allow a large number of spurious pronunciations. For the segment-based model, we used only the "good" parameter initialization; for the frame-based model, we compared both initializations to get a sense of the sensitivity to initial parameters. As a reference, we also measured the performance of a baseform-only model and one that expands the baseforms with a large set of phonological rules (the "full rule set" described in [8]), neither of which was trained. Table 2 shows the performance of these models using the following measures of performance: **(1)** *Error rate (ER)*, or the percentage of incorrectly classified words; **(2)** *failure rate (FR)*, the percentage of input transcriptions for which the correct word receives 0 probability, i.e. for which the input is not an allowed realization of the word; and **(3)** *cohort size (CS)*, the number of words in the vocabulary for which $p(word|S_{1:T}^{\{F\}}) > 0$, averaged over all input transcriptions. The cohort should be as small as possible, while still containing the correct word.

The main difference between the segment-based and frame-based variants is in the cohort size, which is about 60% higher for the frame-based model. As the last two lines in the table show, the untrained models have very different error rates, depending on the choice of initialization. After training, the error rates are quite similar for both initial conditions. When using the "good" initialization, EM training does not have a significant effect on the error rate. As expected, EM has almost no effect on the failure rate and cohort size, which are determined purely by zeros in the DBN's conditional probability tables.

model	dev set			test set		
	ER	FR	CS	ER	FR	CS
baseforms only	60.0	57.6	0.5	64.8	62.3	0.5
phonological rules	57.0	54.5	0.5	63.1	58.5	0.6
seg.-based, "good" init	29.7	19.4	29.4	45.3	30.1	31.5
+ EM	29.7	19.4	29.0	44.1	30.1	31.2
fr.-based, "good" init	28.5	16.4	47.1	40.7	24.6	50.0
+ EM	27.9	16.4	47.1	40.7	24.6	50.0
fr.-based, "bad" init	73.3	16.4	47.1	76.3	24.6	50.0
+ EM	27.9	16.4	47.1	37.3	24.6	50.0

Table 2: *Development and test set performance. See the text for descriptions of the performance measures.*

[1] We used only those words whose phonemic pronunciations have at least 4 phonemes, so as to limit context effects from adjacent words.

However, these measures do not give the full picture. In an end-to-end recognizer, the model would be combined with language and acoustic scores. Therefore, it is important that, if the correct word is not top-ranked, its rank is as high as possible, and that the correct word scores as well as possible relative to competing words. Figure 2 (top) shows the cumulative distributions of the correct word's rank for the test set, using the frame-based model with both initializations. Figure 2 (bottom) shows the cumulative distributions of the *score margin*, the difference in per-frame log probability between the correct word and the highest-scoring incorrect word, in the same conditions. In all cases, training qualitatively improves the distributions. Similar distributions are obtained for the development set and for the segment-based model.

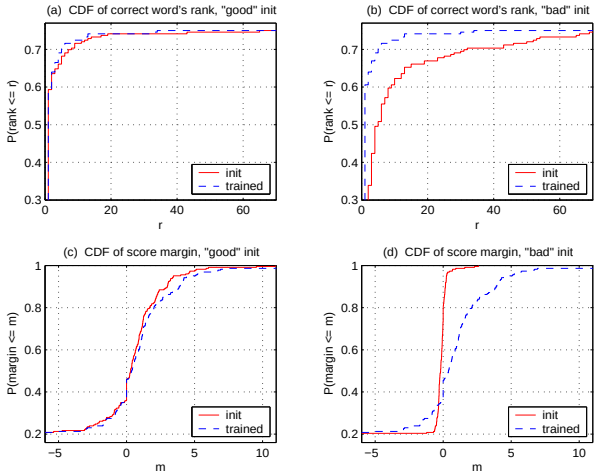

Figure 2: *Empirical cumulative distribution functions of the correct word's rank (top) and the score margin (bottom) for the test set, before and after training, using the frame-based model with different initializations. Failures are assumed to have a rank equal to the vocabulary size.*

4. Discussion

The experiments we have presented give an idea of the effect of EM training on the performance of the feature-based pronunciation model, as well as of the effect of different initializations. We have seen that EM does not suffer from a poor initialization (at least within the hard constraints we have imposed), and that, given a good initialization, the main effect of training is typically not to reduce the error rate but to improve the rank and score distributions. This may be particularly important when the model is incorporated into a complete recognizer.

While the two model variants we have tested–frame-based and segment-based–do not differ greatly in terms of error rate, the cohort size is much larger for the frame-based than for the segment-based variant. A large cohort size could be problematic, both because it increases the computational load and because it may indicate increased inter-word confusability. The difference may be explained by the fact that we are currently using context-independent feature substitution probabilities $p(S_t^j|U_t^j)$, and some substitutions are clearly licensed only in certain contexts. We are currently working on the incorporation of context-dependent probabilities into the model.

We have not addressed the relationship between the features and the acoustics. A number of past and current research efforts have been focused on this issue, using various feature sets. The pronunciation model we have described could be combined with a feature classifier using the same articulatory feature set. However, there need not be an exact match between the feature set used by the pronunciation model and the one used by the acoustic model; for example, the feature set we have used can be translated to a more abstract manner/place representation and vice versa. Alternatively, a phonetic recognizer could be used to produce a detailed phone lattice, which could then be used as input in a similar way to our Switchboard experiments. We are currently investigating options for integration into a complete recognizer.

5. References

[1] J. Bilmes and G. Zweig, "The Graphical Models Toolkit: An open source software system for speech and time-series processing," *ICASSP*, Orlando, 2002.

[2] C. P. Browman and L. Goldstein, "Articulatory phonology: An overview," *Phonetica*, **49:**155–180, 1992.

[3] S. Chang, S. Greenberg, and M. Wester, "An elitist approach to articulatory-acoustic feature classification," *Eurospeech*, Aalborg, Denmark, 2001.

[4] T. Dean and K. Kanazawa, "A model for reasoning about persistence and causation," *Computational Intelligence*, **5**:142–150, 1989.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, **39**:1–38,1977.

[6] L. Deng, G. Ramsay, and D. Sun, "Production models as a structural basis for automatic speech recognition," *Speech Communication*, **33**:93–111, 1997.

[7] S. Greenberg, J. Hollenback, and D. Ellis, "Insights into spoken language gleaned from phonetic transcription of the Switchboard corpus," *ICSLP*, Philadelphia, 1996.

[8] T. J. Hazen, I. L. Hetherington, H. Shu, and K. Livescu, "Pronunciation modeling using a finite-state transducer representation," *ITRW PMLA*, Estes Park, CO, 2002.

[9] S. King, T. Stephenson, S. Isard, P. Taylor, and A. Strachan, "Speech recognition via phonetically featured syllables," *ICSLP*, Sydney, 1998.

[10] K. Kirchhoff, "Syllable-level desynchronisation of phonetic features for speech recognition," *ICSLP*, Philadelphia, 1996.

[11] K. Livescu, J. Glass, and J. Bilmes, "Hidden feature models for speech recognition using dynamic Bayesian networks," *Eurospeech*, Geneva, 2003.

[12] K. Livescu and J. Glass, "Feature-based pronunciation modeling for speech recognition," *HLT/NAACL*, Boston, 2004.

[13] D. McAllester, L. Gillick, F. Scattone, and M. Newman, "Fabricating conversational speech data with acoustic models: A program to examine model-data mismatch," *ICSLP*, Sydney, 1998.

[14] M. D. Riley and A. Ljolje, "Automatic generation of detailed pronunciation lexicons," in C.-H. Lee, F. K. Soong, and K. K. Paliwal (eds.), *Automatic Speech and Speaker Recognition*, Kluwer Academic Publishers, Boston, 1996.

[15] A. A. Wrench and K. Richmond, "Continuous speech recognition using articulatory data," *ICSLP*, Beijing, 2000.