S3.9

Decision Trees for Phonological Rules in Continuous Speech

L.R. Bahl, P.V. de Souza, P.S. Gopalakrishnan, D. Nahamoo, M.A. Picheny
IBM Research, T.J. Watson Research Center
P.O. Box 704, Yorktown Heights, NY 10598

ABSTRACT

In a continuous speech recognition system it is important to model the context dependent variations in the pronunciations of words. Traditionally, this has been done manually by phoneticians, who construct sets of rules that explain such phonological variations. In this paper we present an automatic method for modeling phonological variation using decision trees. For each phone we construct a decision tree that specifies the acoustic realization of the phone as a function of the context in which it appears. Several thousand sentences from a natural language corpus spoken by several talkers are used to construct these decision trees. Experimental results on a 5000-word vocabulary natural language speech recognition task are presented.

1 INTRODUCTION

It is well known that the pronunciation of a word or subword unit such as a phone depends heavily on the context. This phenomenon has been studied extensively by phoneticians who have constructed sets of phonological rules that explain this context dependence [10, 16]. However, the use of such rules in recognition systems has not been extremely successful. Perhaps, a fundamental problem with this approach is that it relies on human perception rather than acoustic reality. Furthermore, this method only identifies gross changes, and the more subtle changes, which are generally unimportant to humans but may be of significant value in speech recognition by computers, are ignored. Possibly, rules constructed with the aid of spectrograms would be more useful, but this would be very tedious and difficult.

We propose an automatic method for modeling the context dependence of pronunciation. In particular, we expand on the use of decision trees for modeling allophonic variation, which we previously outlined in [4]. Other researchers have modeled all distinct sequences of tri-phones (three consecutive phones) in an effort to capture phonological variations [18, 12]. The method proposed in this paper has the advantage that it allows us to account for much longer contexts. In the experiments reported in this paper, we model the pronunciation of a phone as a function of the five preceding and five following phones. This method also has better powers of generalization, i.e. modeling contexts that do not occur in the training data.

We have previously used decision trees for statistical modeling of natural language [5], and for the generation of spelling-to-sound rules [6]. Use of decision trees for identifying allophones have been considered in [4, 9, 13, 17]. However, apart from [4], these methods have either not been used in a recognizer or have not provided significant improvements over existing modeling methods.

In the next section we describe the preparation of the data needed for the construction of the decision trees. In Section 3 we describe the algorithms used for constructing the decision trees, and we also present experimental results on a 5000-word natural language continuous speech recognition task. Concluding remarks are presented in Section 4.

2 INPUT DATA FOR TREE CONSTRUCTION

The decision trees are constructed using a database of 20,000 continuous speech natural language sentences spoken by 10 different speakers. For more details about this database, see [4]. Spectral feature vectors are extracted from the speech at a rate of 100 frames per second. These frames are labeled by a vector quantizer using a common alphabet for all the speakers. This data is used to train a set of phonetic Markov models for the words. Using the trained phonetic Markov model statistics and the Viterbi algorithm, the labeled speech is then aligned against the phonetic baseforms. This process results in an alignment of a sequence of phones (the phone sequence obtained by concatenating the phonetic baseforms of the words in the entire training script) with the label sequence produced by the vector quantizer. In our case the training data consists of about 320,000 words, 1,500,000 phones, and 10,000,000 frames. For each aligned phone we construct a data record which contains:

- The identity of the current phone, denoted as P_0.
- The context, i.e. the identities of the K previous phones and K following phones in the phone sequence, denoted as $P_{-K}, \dots P_{-1}, P_1, \dots P_K$.
- The label sequence aligned against the current phone, denoted as y.

We partition this collection of data on the basis of P_0. Thus we have collected, for each phone in the phone alphabet, several thousand instances of label sequences in various phonetic contexts. Based on this annotated data we construct a decision tree for each phone. This collection of trees is referred to as a *Decision Tree Wood* (DTW for short).

The quality of the decision trees depends on the amount of data used in their construction. We have collected a sizable amount of speech data, however one should never forget the immortal words (sic) of Robert LeRoy Mercer [14]:

Dere's no data like mo' data.

3 CONSTRUCTING THE DECISION TREE

If we had an unlimited supply of annotated data, we could solve the context dependence problem exhaustively by constructing a different model for each phone in each possible context. Of course, we do not have enough data to do this,

CH2977-7/91/0000-0185 $1.00 © 1991 IEEE

but even if we could carry out the exhaustive solution, it would take a gigantic amount of storage to store all the different models. Thus, because of limited data, and a need for parsimony, we combine the contexts into equivalence classes, and make a model for each class. Obviously, each equivalence class should consist of contexts that result in similar label strings. One effective way of constructing such equivalence classes is by the use of binary decision trees. Readers interested in this topic are urged to read *Classification and Regression Trees* by Breiman, Friedman, Olshen and Stone [8] which is the *bhagavada gita*[1] of decision arborists.

To construct a binary decision tree we begin with a collection of data, which in our case consists of all the annotated samples for a particular phone. We split this into two subsets, and then split each of these two subsets into two smaller subsets, and so on. The splitting is done on the basis of binary questions about the context P_i, for $i = \pm 1, \dots \pm K$. In order to construct the tree, we need to have a goodness-of-split evaluation function. We base the goodness-of-split evaluation function on a probabilistic measure that is related to the homogeniety of a set of label strings. Finally, we need some stopping criteria. We terminate splitting when the number of samples at a node falls below a threshold, or if the goodness of the best split falls below a threshold. The result is a binary tree in which each terminal node represents one equivalence class of contexts. Using the label strings associated with a terminal node we can construct a fenonic Markov model for that node by the method described in [1, 2]. During recognition, given a phone and its context, we use the decision tree of that phone to determine which model should be used. By answering the questions about the context at the nodes of the tree, we trace a path to a terminal node of the tree, which specifies the model to be used.

Let Q denote a set of binary questions about the context. Let n denote a node in the tree, and $m(q,n)$ the goodness of the split induced by question $q \in Q$ at node n. We will need to distinguish between tested and untested nodes. A tested node is one on which we have evaluated $m(q,n)$ for all questions $q \in Q$ and either split the node or designated it as a terminal node. It is well-known that the construction of an optimal binary decision tree is an NP-hard problem. We use a sub-optimal greedy algorithm to construct the tree, selecting the best question from the set Q at each node. In outline, the decision tree construction algorithm works as follows:

1 Start with all samples at the root node.

2 While there are untested nodes do

- **2.1** Select some untested node n.
- **2.2** Evaluate $m(q,n)$ for all possible questions $q \in Q$ at this node.
- **2.3** If a stopping criterion is met, declare this node as terminal.
  else
- **2.4** Associate the question with the highest value of $m(n,q)$ with this node. Make two new successor nodes. All samples that answer positively to the question are transferred to the left successor and all other samples are transferred to the right successor.

[1] A reviewer objected to the use of *bible*.

The most important aspects of this algorithm are the set of questions Q, the goodness-of-split evaluation function $m(q,n)$, and the stopping criteria. We discuss each of these below.

3.1 The Question Set

Let P denote the alphabet of phones, and N_P the size of this alphabet. In our case $N_P = 55$. The question set Q consists of questions of the form [Is $P_i \in S$] where $S \subset P$. We start with singleton subsets of P, e.g. $S = \{p\}$, $S = \{t\}$, etc. In addition, we use subsets corresponding to phonologically meaningful classes of phones commonly used in the analysis of speech [11], e.g., $S = \{p,t,k\}$ (all unvoiced stops), $S = \{p,t,k,b,d,g\}$ (all stops), etc. Each question is applied to each element P_i for $i = \pm 1, \dots \pm K$, of the context. If there are N_S subsets in all, the number of questions N_Q is given by $N_Q = 2KN_S$. Thus there will be N_Q splits to be evaluated at each node of the tree. In our experiments $K = 5$ and $N_S = 130$, leading to a total of 1300 questions.

Note that, in general, there are 2^{N_P} different subsets of P, and, in principle, we could consider all $2K2^{N_P}$ questions. Since this would be too expensive, we have chosen what we consider to be a meaningful subset of all possible questions and consider only these *fixed* set of questions during tree construction. It is possible to generalize the tree construction procedure to use *variable* questions which are constructed algorithmically as part of the tree construction process, as in [5, 15].

Furthermore, the type of questions we use are called *simple* questions, since each question is applied to one element of the context at a time. It is possible to construct *complex* questions which deal with several context elements at once, as in [5]. Again, we did not use this more complicated technique in the experiments reported in this paper.

3.2 The Goodness-of-Split Evaluation Function

We now derive the goodness-of-split evaluation function based on a probabilistic model of collections of label strings. Let $\mathcal{M}$ denote a particular class of parametric models that assign probabilities to label strings. For any model $M \in \mathcal{M}$ let $Pr_M(y)$ denote the probability assigned to label string y. Let Y_n be the set of label strings associated with node n. $Pr_M(Y_n) = \prod_{y \in Y_n} Pr_M(y)$ is a measure of how well the model M fits the data at node n. Let $M_n \in \mathcal{M}$ be the best model for Y_n, i.e. $Pr_{M_n}(Y_n) \geq Pr_M(Y_n)$ for all M. $Pr_{M_n}(Y_n)$ is a measure of the purity of Y_n. If the label strings in Y_n are similar to each other, then $Pr_{M_n}(Y_n)$ will be large. A question q will split the data at node n into two subsets based on the outcome of question q. Our goal is to pick q so as to make the successor nodes as pure as possible. Let Y_l and Y_r denote the subsets of label strings at the left and right successor nodes, respectively. Obviously, $Y_l \cup Y_r = Y_n$. Let M_l and M_r be the corresponding best models for the two subsets. Then

$$m(q,n) = log\left(\left(Pr_{M_l}(Y_l)Pr_{M_r}(Y_r)\right)/Pr_{M_n}(Y_n)\right) \qquad (1)$$

is a measure of the improvement in purity as a result of the split. Since our goal is to divide the strings into subsets containing similar strings, this quantity serves us well as the goodness-of-split evaluation function.

Since, we will eventually use the strings at a terminal node to construct a Markov model, choosing $\mathcal{M}$ to be a class of Markov models would be the natural choice. Unfortunately, this choice of model is computationally very expensive. To find the best model M_n we would have to train the model, using the forward-backward algorithm using all the data at the node n. Thus for computational reasons, we have chosen a simpler class of models – Poisson models of the type used in [3] for the polling fast match.

Recall that y is a sequence of acoustic labels $a_1, a_2, \ldots a_l$. We make the simplifying assumption that the labels in the sequence are independent of each other. The extent to which this approximation is inaccurate depends on the length of the units being modeled. For strings corresponding to single phones, the inaccuracy introduced by this approximation is relatively small. However, it results in an evaluation function that is easy to compute and leads to the construction of very good decision trees in practice.

A result of this assumption is that the order in which the labels occur is of no consequence. Now, a string y can be fully characterized by its histogram, i.e. the number of times each label in the acoustic label alphabet occurs in that string. We represent the string y by its histogram $y_1 y_2 \ldots y_F$, a vector of length F where F is the size of the acoustic label alphabet and each y_i is the number of times label i occurs in string y. We model each component y_i of the histogram by an independent Poisson model with mean rate μ_i. Then, the probability assigned to y by M is

$$Pr_M(y) = \prod_{i=1}^{F} \frac{\mu_i^{y_i} e^{-\mu_i}}{y_i!} \tag{2}$$

The joint probability of all the strings in the set Y_n is then

$$Pr_M(Y_n) = \prod_{y \in Y_n} \prod_{i=1}^{F} \frac{\mu_i^{y_i} e^{-\mu_i}}{y_i!} \tag{3}$$

It can be easily shown that $Pr_M(Y_n)$ is maximized by choosing the mean rate to be the sample average, i.e., the best model for Y_n has as its mean rate

$$\mu_{ni} = \frac{1}{N_n} \sum_{y \in Y_n} y_i \quad \text{for } i = 1, 2 \ldots F \tag{4}$$

Let μ_{li} and μ_{ri} for $i = 1, 2 \ldots F$, denote the optimal mean rates for Y_l and Y_r respectively. Then $m(q, n)$ can be written as

$$\begin{aligned} m(q,n) \;=\; \sum_{i=1}^{F} \Bigg[& \sum_{y \in Y_l} (y_i \log \mu_{li} - \mu_{li}) - \sum_{y \in Y_l} \log(y_i!) \\ & + \sum_{y \in Y_r} (y_i \log \mu_{ri} - \mu_{ri}) - \sum_{y \in Y_r} \log(y_i!) \\ & - \sum_{y \in Y_n} (y_i \log \mu_{ni} - \mu_{ni}) + \sum_{y \in Y_n} \log(y_i!) \Bigg] \end{aligned} \tag{5}$$

Now, it is easily seen that for $i = 1, 2, \ldots F$

$$\sum_{y \in Y_l} \log(y_i!) + \sum_{y \in Y_r} \log(y_i!) = \sum_{y \in Y_n} \log(y_i!) \tag{6}$$

and

$$\sum_{y \in Y_l} \mu_{li} + \sum_{y \in Y_r} \mu_{ri} = \sum_{y \in Y_n} \mu_{ni} \tag{7}$$

Eliminating these terms and using equation (4) we have the evaluation function

$$\begin{aligned} m(q,n) \;=\; \sum_{i=1}^{F} \{ & N_l \mu_{li} \log \mu_{li} + N_r \mu_{ri} \log \mu_{ri} \\ & - N_n \mu_{ni} \log \mu_{ni} \} \end{aligned} \tag{8}$$

where N_l is the total number of strings at the left node and N_r is the total number of strings at the right node resulting from split q. At each node, we select the question q that maximizes the evaluation function (8).

The computation of the function $m(q, n)$ for each question q at node n requires $O(N_n F)$ operations. The number of operations to find the best question at a node is $O(N_Q N_n F)$ using a straight-forward computation. Some of the computations can be shared across several questions which reduces the number of computations to $O(K N_P N_n F + N_Q N_P F)$. The details are easy to work out and are omitted here. Thus, the computation required to construct the decision tree is not very large.

The evaluation function given in equation (8) is very general, and arises from several different model assumptions. For example, if we assume that the length of each string is given by a Poisson distribution, and the labels in a string are produced independently by a multinomial distribution, then the evaluation function of equation (8) results. There are also some interesting relationships between this function and a minimization of entropy formulation. Due to space limitations, the details are omitted here but may be found elsewhere [7].

3.3 The stopping criteria

We use two very simple stopping criteria. If the value $m(q, n)$ of the best split at a node n is less than a threshold T_m we designate it to be a terminal node. Also, if the number of samples at a node falls below a threshold T_s then we designate it to be a terminal node. The thresholds T_m and T_s are selected empirically.

3.4 Using the Decision Trees During Recognition

The terminal nodes of a tree for a phone correspond to the different allophones of the phone. We construct a fenonic Markov model for each terminal node from the label strings associated with the node. The details of this procedure are described in [1, 2].

During recognition, we construct Markov models for word sequences as follows. We construct a sequence of phones by concatenating the phonetic baseforms of the words. For each phone in this sequence, we use the appropriate decision tree and trace the path in the tree corresponding to the context provided by the phone sequence. This leads to a terminal node, and we use the fenonic Markov model associated with this node. By concatenating the fenonic Markov models for each phone we obtain a Markov model for the entire word sequence.

For the last few phones in the phone sequence, the right context is not fully known. For these phones, we make tentative models ignoring the unknown right context. When the sequence of words is extended, the right context for these phones will be available, and we can replace the tentative models by the correct models and recompute the acoustic match probabilities.

This procedure is quite simple and the details are omitted here.

We tested this method on a 5000-word, continuous speech, natural language task. Details of the task and the recognition system can be found in [4]. We constructed the decision trees using the data described in Section 2. The phone alphabet was of size 55, K was chosen to be 5, and on the average the number of allophones per phone was 45.

We tested the system with 10 talkers. Each talker provided roughly 2,000 sentences of training data for constructing the vector quantizer prototypes and for training the Markov model parameters. The test set consisted of 50 sentences (591 words) for each speaker. Tests were also done using context independent phonetic models. Table I shows the error rates for the phonetic (context independent) and allophonic (context dependent) models for the 10 talkers. On the average, the word error rate decreases from 9.2% to 5.3%.

4 CONCLUSIONS

Acoustic models used in continuous speech recognition systems should account for variations in pronunciation arising from contextual effects. This paper demonstrates that such effects can be discovered automatically, and represented very effectively using binary decision trees. We have presented a method for constructing and using decision trees for modeling the phonetic environment. The results prove the old maxim that trees are good for the environment.

Speaker	Percentage Error	
	Phonetic Models	Allophonic Models
T1	9.7	7.6
T2	15.8	8.4
T3	12.5	6.3
T4	10.7	6.5
T5	2.6	1.7
T6	12.7	6.8
T7	7.0	3.0
T8	2.8	2.6
T9	5.8	3.5
T10	12.4	6.9
Average	9.2%	5.3%

Table 1. Recognition Error Rate

REFERENCES

[1] L.R. Bahl, P.F. Brown, P.V. de Souza, R.L. Mercer, M.A. Picheny, "Automatic Construction of Acoustic Markov Models for Words," Proc. International Symposium on Signal Processing and Its Applications, Brisbane, Australia, 1987, pp.565-569.

[2] L.R. Bahl, P.F. Brown, P.V. de Souza, R.L. Mercer, and M.A. Picheny, "Acoustic Markov Models Used in the Tangora Speech Recognition System," Proc. ICASSP-88, New York, NY, April 1988, pp. 497-500.

[3] L.R. Bahl, R. Bakis, P.V. de Souza, and R.L. Mercer, "Obtaining Candidate Words by Polling in a Large Vocabulary Speech Recognition System," Proc. ICASSP-88, New York, NY, April 1988, pp. 489-492.

[4] L.R. Bahl et. al.,"Large Vocabulary Natural Language Continuous Speech Recognition," Proc. ICASSP-89, Glasgow, Scotland, May 1989, pp.465-467

[5] L.R. Bahl, P.F. Brown, P.V. de Souza, R.L. Mercer, "A Tree-Based Language Model for Natural Language Speech Recognition," IEEE Transactions on ASSP, Vol. 37, No. 7, July 1989, pp.1001-1008.

[6] L.R. Bahl, S. Das, P.V. de Souza, M. Epstein, R.L. Mercer, B. Merialdo, D. Nahamoo, M. A. Picheny, J. Powell, "Automatic Phonetic Baseform Determination," elsewhere in these proceedings.

[7] L.R. Bahl, P.V. de Souza, P.S. Gopalakrishnan, A. Nadas, D. Nahamoo, M.A. Picheny, "Splitting Rules for Phonological Decision Trees," IBM Research Report, in preparation.

[8] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, *Classification and Regression Trees*, Wadsworth Statistics/Probability Series, Belmont, CA, 1984.

[9] F.R. Chen, J. Shrager, "Automatic Discovery of Contextual Factors Describing Phonological Variation", Proc. 1989 DARPA Workshop on Speech and Natural Language.

[10] P.S. Cohen and R.L. Mercer, "The Phonological Component of an Automatic Speech Recognition System," in *Speech Recognition,* D.R Reddy, editor, Academic Press, New York, 1975, pp.275–320.

[11] G. Fant, *Speech Sounds and Features*, MIT Press, Cambridge, MA, 1973.

[12] K.F. Lee, H.W. Hon, M.Y. Hwang, S. Mahajan, R. Reddy, "The Sphinx Speech Recognition System," Proc ICASSP-89, Glasgow, Scotland, May 1989, pp.445-448

[13] K.F. Lee, et. al., "Allophone Clustering for Continuous Speech Recognition", Proc. ICASSP-90, Albuquerque, NM, April 1990, pp.749-752.

[14] R.L. Mercer, "Language Modeling (invited paper)," IEEE Workshop on Speech Recognition, Arden House, Harriman, NY, May 1988.

[15] A. Nadas, D. Nahamoo, M.A. Picheny and J. Powell, "An Iterative Flip-Flop Approximation of the Most Informative Split in the Construction of Decision Trees," elsewhere in these proceedings.

[16] B.T. Oshika, V.W. Zue, R.V. Weeks, H. Nue and J. Auerbach, "The Role of Phonological Rules in Speech Understanding Research," IEEE Transactions on ASSP, Vol. ASSP–23, 1975, pp. 104–112.

[17] M.A. Randolph, "A Data-Driven Method for Discovering and Predicting Allophonic Variation", Proc. ICASSP-90, Albuquerque, NM, April 1990, pp.1177-1180.

[18] R. Schwartz, Y. Chow, O. Kimball, S. Roucos, M. Krasner, J. Makhoul, "Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech," Proc. ICASSP-85, April 1985