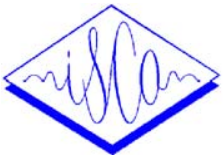

5th International Conference on Spoken Language Processing (ICSLP 98)
Sydney, Australia
November 30 - December 4, 1998

# Optimized Stopping Criteria for Tree-Based Unit Selection in Concatenative Synthesis

*Andrew Cronk and Michael Macon*

Center for Spoken Language Understanding
Oregon Graduate Institute, Portland, Oregon U.S.A.
http://cslu.cse.ogi.edu/tts

## ABSTRACT

The lack of naturalness hampers the widespread application of speech synthesis. Increasing the size of the unit database in a concatenative speech synthesizer has been proposed as a method to increase the variety of units—thereby improving naturalness. However, expanding the unit database increases the computational cost of selecting the most appropriate unit and compounds the risk that a perceptually suboptimal unit is chosen. Clustering the unit database prior to synthesis is an effective method for reducing this cost and risk.

In this study, a unit selection method based on tree-structured clustering of data is implemented and evaluated. This approach to tree construction differs from similar approaches used in both synthesis and recognition in that a "right-sized" tree is found automatically rather than using hand-tuned stopping criteria. The tree is grown to its maximum size, and its leaves are systematically recombined in order to determine the most suitable subtree.

Trees are grown using the automatic stopping method and compared with those grown using thresholds. Cross validation shows that trees grown to their maximum size and systematically recombined produce fuller clusters with lower objective distortion measures than trees whose growth is arrested by a threshold. The study concludes with a discussion of how these results may affect the perceptual quality of a speech synthesizer.

10.21437/ICSLP.1998-23

## 1. BACKGROUND

Natural speech is characterized by segmental variations influenced by many factors. A synthesizer is judged by its ability to simulate this variation. Increasing the size and variety of the unit database in a concatenative speech synthesizer has been proposed as a method to improve the naturalness of synthetic speech. However, increasing the size of the speech database mandates that more sophisticated techniques be employed to select the most appropriate unit.

As discussed in [2], one approach to concatentative speech synthesis employs a large database of prerecorded, subphonetic units. In order to synthesize any target unit sequence, a candidate pool of units is selected from the speech database for each target. Once a sequence of candidate pools or clusters has been selected, a dynamic programming (Viterbi) search is performed to find the sequence of database units that minimize the concatenation cost of the overall sequence. This process is depicted in Figure 1.

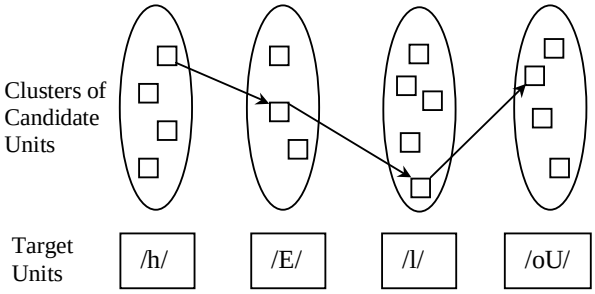


**Figure 1:** When multiple instances of a unit type exist in the database, the selection process returns clusters of candidate units to match the target sequence[1]. A dynamic programming search finds the sequence of units through the clusters that minimizes the concatenation cost.

In [1], the concepts of target and concatenation cost are presented. Target cost reflects how well the linguistic and contextual features a given database unit resemble the ideal target. On the other hand, the concatentation cost reflects how a unit will join with the previously selected unit. A tree structure can be used to efficiently and prudently cluster the database units prior to synthesis. Within this approach to synthesis, the clusters represent units with equivalent target cost. The tree offers a means to find the optimal cluster at runtime, and the dynamic programming search determines the concatenation cost.

### 1.1. Classification and Regression Trees

The work of Breiman *et al.* on classification and regression trees (CART) [3] provides the theoretical framework for developing phonetic decision trees. The basic classification tree is defined by four elements:

1. A set of binary splitting questions
2. A goodness-of-split criterion
3. A stop-splitting rule
4. A rule for assigning every terminal node to a class

The set of questions represents possible partitions of the unit space. There are two types of splitting questions: categorical and numerical. A question is categorical if it takes values in a finite set not having a natural ordering. A question is numerical if its values are real numbers. A sample numerical question might be "Is the average fundamental frequency (F0) of the unit between 123.4 and 135.3 Hz?" Whereas, a sample categorical question might be, "Is the unit a fricative?"

1 All phonemic transcriptions herein use the Worldbet phonebet.

The goodness-of-split criterion is used to determine which of the available splitting questions best divides the subspace. The best split maximizes the decrease in data impurity. To elaborate, the data within child nodes after the split should have lower within-cluster variance than the data in the parent node before the split.

The stopping criterion halts the propagation of a tree branch. Various thresholds have been proposed, such as the minimum improvement of some impurity measure and the minimum number of units per cluster [2, 3]. Stopping criteria involving thresholds may lead to trees larger than the data warrants. Such trees may not be able to generalize well to non-training data. In [3], it is proposed that growing the tree to the maximum size and then systematically pruning leaves produces a more reliable classifier.

The fourth CART specification assigns terminal nodes to a classification. This specification does not apply to the synthesis problem, as the classifications are not known *a priori*. In CART, multiple leaves can be tied to a single classification, but in the clustering problem each leaf must be assumed to be a new category. The unsupervised nature of clustering distinguishes the approach from classification.

## 1.2. Constructing a Phonetic Decision Tree

Classification and regression trees are built by posing a series of questions to a set of units. Each question must be in a form that results in a binary split of the data. This process results in a recursive partitioning of the unit space. At each node, the splitting question is found that minimizes a measure of impurity. An example of a simple CART is illustrated in Figure 2.

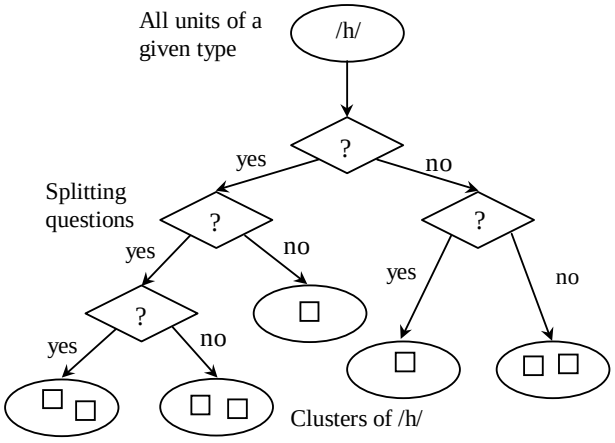


**Figure 2:** A classification tree is built by posing questions at each node that result in binary splits. Units within a leaf have equivalent target costs.

As its name suggests, CART is a classifier, not a clustering algorithm. While it does provide the theoretical foundation for phonetic decision tree construction, many of the CART techniques must be adapted when applied to clustering.

The primarily difference between Breiman's CART and clustering is seen when evaluating the goodness of a tree. Cross validation is a technique in which classified data is held out to estimate the misclassification rate of the tree. V-fold cross validation repeats the cross validation process *V* times, reshuffling the data with iteration, in order to provide a more accurate estimate. Obviously, this technique requires a set of test data whose classifications are known in advance. In the clustering problem, the classifications are not known in advance—indeed the whole purpose of clustering is to determine these categories.

# 2. CLUSTERING PARAMETERS

In this experiment, the speech database was constructed from 450 TIMIT sentences spoken by a male, native speaker of American English. The phones were labeled using HMM-based forced alignment and grouped by type. Each unit is denoted by a vector of linguistic and contextual features, as well as an acoustic representation based upon calculating 13 cepstral coefficients over 30 ms windows every 5 ms.

## 2.1. Splitting Questions

Twenty-two categorical or numerical variables comprise the feature vectors used in this study. From these variables, 192 splitting questions are generated. Numerical questions are formed by dividing the variable range into 10 equal subranges.

## 2.2. Goodness-of-split

Two types of distortion measures are needed to construct the decision tree. One to measure the between-unit distortion and another to determine the within-cluster distortion (*i.e. variance)*. With these measures, the goodness-of-split measure can be defined.

The between-unit distortion measure $d(U,V)$ is given as

$$d(U, V) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\left[W_j (U_{ij} - V_{ij})^2\right]}{N}, \quad (1)$$

where variables $U$ and $V$ denote the acoustic representations of units. The number of frames in the longer unit is $N$, and $M$ is the number of coefficients in any frame. The shorter unit is linearly interpolated to the length of the longer unit. The notation, $U_{ij}$, refers to the *jth* cepstral coefficient in the *ith* frame of unit $U$. The weight, W, reflects the Mahalanobis distance, in which each coefficient is weighted by the inverse of the variance. Mahalanobis distance is theoretically appealing as the ranges of the coefficients may vary widely. By using the inverse of the variance as the weight, each coefficient is given a chance to contribute. In essence, Equation 1 calculates the mean squared error at the frame level.

The impurity function $D(C)$ determines the within-cluster distortion:

$$D(C) = \frac{2 * \left( \sum_{i=1}^{|C|}\sum_{j=i}^{|C|}\left[d\left(U_i, U_j\right)\right] \right)}{|C|^2 - |C|}. \quad (2)$$

The expression, $|C|$ , represents the number of units in cluster $C$. This equation (2) reflects the average pair-wise, between-unit distortion within the cluster. Note that the within-cluster distortion of a leaf containing only one unit is defined to be zero.

The selection of a goodness-of-split function influences the overall shape of the tree. A poorly chosen goodness-of-split measure can lead to an abundance of end-splits which cause the tree to degenerate. An end-split occurs when the best split results in a cluster containing a few units or a single unit and another cluster containing many. Degenerate binary trees have two drawbacks. First, degenerate trees tend to be right-biased, which means the growth occurs in the "no" direction. In general, a "no" split is less discerning than a "yes" split—and therefore less preferable. Any path to a leaf in the right-biased degenerate case contains at most a single "yes" split, whereas in the more complete tree, a path may contain many. Second, degenerate trees have far fewer subtrees than their complete counterparts. This means that far fewer partitions of the unit space will be considered in the search for the most suitable partition.

The goodness-of-split function $G(C_1,C_2)$ is given by

$$G(C_1, C_2) = \frac{D(C_1)T(C_1) + D(C_2)T(C_2)}{T(C_1) + T(C_2)}, \quad (3)$$

where $C_1$ and $C_2$ are clusters. The term, $T(C)$, reflects the weight such that $T(C) = .5(|C|^2-|C|)$. By weighting the within-cluster distortion $D(C)$ by a function of the number of units in cluster $C$, balanced trees are encouraged.

## 2.3. Stopping Criteria

Critical to the producing the "right" tree is the ability to stop its growth at the appropriate time. In several other published works, the predominant method of halting tree growth is by setting thresholds [2, 3, 4, 6, 7]. The exact threshold varies according to the task, but minimum node occupancy (of units or frames) or minimum improvement of some measurement (such as impurity) are common.

Finding the ideal stopping threshold is difficult, often requiring much trial-and-error. Also, the threshold for a given data set may not be appropriate for another. Means for automatically determining the most suitable tree are preferable. An alternative is to grow the tree to purity and then selectively recombine leaves in order to produce a more reliable clustering.

In the *minimum occupancy stopping criterion*, the best split that allows each child to contain at least the minimum required units is selected. If no such question exists, that branch of the tree stops growing. Minimum occupancy thresholds between 2 and 20 units per cluster are considered. Similarly, the *minimum improvement stopping criterion* considers the percent change in impurity (within-cluster distortion) between the parent and child nodes. If the impurity in either child node, divided by the impurity in parent node, is not greater than the threshold, no further splits along the branch are allowed. Thresholds from .05 to .95 are considered in increments of .05.

The *recombination stopping criterion*, as previously described, grows the tree to purity. A greedy algorithm considers every subtree of size $N$-1, selects the best subtree according to the average distance evaluation criterion described in the next section, and then repeats the process until the root is reached. The best subtree is determined through cross validation with the units in the development set.

## 2.4. Tree Evaluation

Determining the "rightness" of a tree implies that some evaluation criterion exists. This criterion must be a fair and appropriate measure of the tree's ability to find the best candidate units during synthesis.

The measure of a tree's performance should reflect requirements of the synthesis process. During synthesis a target feature vector is derived from textual analysis and presented to the appropriate tree. These features determine the path to the leaf (cluster) which the tree deems most likely to contain suitable matches for synthesis. Since any of the units contained in this leaf may be selected by the dynamic programming search, the evaluation of the tree should take into consideration not only how well the most appropriate unit matches the target, but also how poorly the worst unit matches.

In [5], Fukunaga provides insight into how a good evaluation criterion should behave. In essence, an ideal measure of classification goodness should decrease as the number of categories (in this case leaves) increases. If this measure reaches a minimum or becomes flat at some point L, then this point can be used as the proper number of leaves. So, ideally, the graph of evaluation criterion scores would show improvement as the tree size increases until the optimal size is reached, at which point the tree would begin to be overfit to the training data, resulting in a worst score.

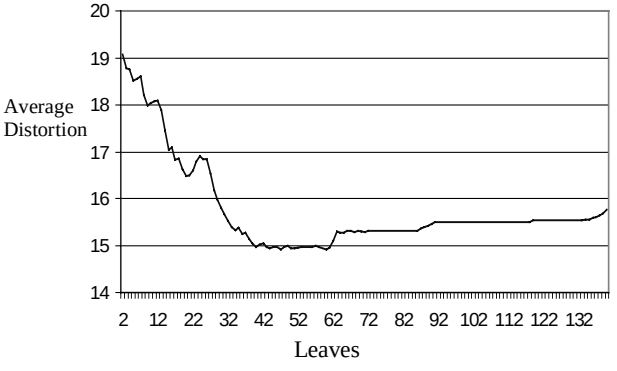


**Figure 3.** Evaluation of a typical fold of an experimental run on the /ei/ data set. The y-axis shows the average distortion at the frame level between the units in development set and units in the corresponding clusters.

To evaluate a tree, data must be reserved for cross validation. When a cross validation target unit is presented to the tree, it returns the cluster that contains the most suitable matches. An evaluation criterion involving the calculation of the average distance between all the units in the selected cluster and the cross validation target demonstrated the desired characteristics.

This approach is theoretically attractive as it considers both the best possible target and the influence of the worst. The graph in Figure 3 shows a typical fold on all instances of /ei/ in the data set using this evaluation criterion. The best tree in Figure 3 has approximately 40 leaves.

## 3. EXPERIMENT

In the experiment, the minimum occupancy and minimum change in impurity thresholds are compared to the recombination method using 5-fold cross validation on three data sets. The results generalized in other informal experiments.

### 3.1. Data Sets

A decision tree must be grown for each unit type. Once feature and acoustic vectors are derived, the corpus is separated by phoneme. To grow any particular tree, the data are divided into three sets: the training set, the development set, and the test set. In these experiments, the training set is comprised of approximately 80% of the data. The development and test sets contain about 10% each. A tree is grown using the training set and cross validated using the development set. Using the development set to cross validate the growth, the best threshold or the best subtree is found. The test set is used to determine how well the best tree generalizes to unseen data. The experiments are performed on the three data sets of various size and phoneme type.

### 3.2. V-fold Cross Validation

Reshuffling the data sets and repeating the experiment amounts to V-fold cross validation. When using thresholds as the stopping criteria, V-fold cross validation on the development set provides an accurate estimate of the best threshold.

When measuring trees grown using threshold-based stopping criteria, the development set indicates which threshold performed the best. When measuring trees using the recombination method, the development set is used to guide the recombination process. The best trees of each approach are then evaluated using the test set.

## 4. Results and Discussion

The results in Table 1 show that the recombination method consistently scores quantitatively better than the threshold-based stopping methods. The thresholds seem to favor limits that resulted in larger trees, resulting in a lower average units-per-node score. The recombination method was able to find trees automatically usually with more units per cluster with less distortion than either threshold.

A clustering module and a unit selection module have been implemented within the Festival Speech Synthesis System developed at the University of Edinburgh. While the initial clustering experiments were performed one phoneme data sets, the unit selection module is based on demiphones. Informal listening tests on short utterances indicate that synthesized speech is positively influenced by the selection of stopping criterion. Further tests are required to determine the extent of the clustering on perception. Online demos of speech synthesis research at the Oregon Graduate Institute can be found at < *http://cslu.cse.ogi.edu/tts/*>.

| Data Set | Recombination Method | | Minimum Occupancy | | Minimum Improvement | |
|---|---|---|---|---|---|---|
| | Ave. Units | Ave. Score | Ave. Units | Ave. Score | Ave. Units | Ave. Score |
| /tS/ | 2.9 | 18.5 | 3.5 | 20.1 | 1.5 | 19.8 |
| /ei/ | 3.1 | 15.6 | 2.3 | 18.6 | 1.6 | 18.7 |
| /i:/ | 3.8 | 19.0 | 2.3 | 22.2 | 1.5 | 22.7 |

**Table 1:** Summary of the experiments on 3 data sets using 5-fold cross validation. The average score indicates the average distortion between the units in the test set and the units in the clusters selected from the best trees grown by each method. A lower average score indicates less variance among units within the clusters.

Future work involves incorporating advanced techniques into the clustering modules such as minimal cost complexity pruning (instead of the current greedy algorithm), look ahead, and iterative feature selection. A more sophisticated concatenation function needs to be incorporated into the unit selection module, and further, more formal perceptual tests performed.

## 5. ACKNOWLEDGEMENTS

We gratefully acknowledge the support of Texas Instruments and the members of the CSLU Industrial Consortium who generously support research in speech synthesis.

## 6. REFERENCES

1. A. W. Black and N. Campbell. Optimising selection of units from speech databases for concatenative synthesis. *Eurospeech95*, vol. 1, pp. 581-584, Madrid, Spain, 1995.

2. A. W. Black and P. Taylor. Automatically clustering similar units for unit selection in speech synthesis. *Eurospeech97*, vol. 2, pp. 601-604, Rhodes, Greece, 1997

3. L. Breiman *et al. Classification and Regression Trees*. Wadsworth & Brooks, Monterey, California, 1984.

4. R. E. Donovan. *Trainable Speech Synthesis*, Ph.D Thesis. Cambridge University, 1996

5. K. Fukunaga. *Introduction to Statistical Pattern Recogition*. Academic Press, secondedition , 1990.

6. H. J. Nock, M. J. F. Gales, and S. J. Young. A comparative study of methods for phonetic decision-tree state clustering. *Eurospeech 97*, vol. 1, pp. 111-114.

7. S. Nakajima. Automatic synthesis unit generation for English speech synthesis based on multi-layered context oriented clustering. *Speech Communication*, vol. 14, pp. 313-324, 1994.

8. W. J. Wang *et al*. Tree-based unit selection for English speech synthesis. *ICASSP-93*, vol. 2, pp. 191-194, Minneapolis, 1993.