IMPROVING DECISION TREES FOR ACOUSTIC MODELING

Ariane Lazaridès, Yves Normandin, Roland Kuhn
lazarids@crim.ca[1], normandin@locus.ca[2], kuhn@crim.ca[1]

[1] Centre de recherche informatique de Montréal (CRIM)
[2] Locus Speech Corporation
1801, McGill College, suite 800
Montréal, Québec, Canada
H3A 2N4

ABSTRACT

In the last few years, the power and simplicity of classification trees as acoustic modeling tools have gained them much popularity. In [1], we studied ''tree units'', which cluster parameters at the HMM level. Building on this earlier work, we examine some new variants of Young *et al*'s ''tree states'', which cluster parameters at the state level [2]. We have experimented with:

1. Making unitary models (which contain additional information about the context)
2. Pruning trees with various severity levels (idea introduced in [1])
3. Pooling some leaves (idea adapted from [2])
4. Refining the questions
5. Questions about the position of the phone within the word
6. Lookahead search
7. Making a single tree for each phone

1. MAKING TREE STATES

A tree state is a particular state of an HMM for a given phone (*e.g.*, state 1 of 'a') for which different phonetic contexts have been clustered by means of a decision tree [2]. In our earlier work on tree units, we employed the Gelfand-Ravishankar-Delp iterative growing and pruning algorithm [3] to build the trees. Fortunately, this efficient and easily implemented algorithm carries over almost unchanged to tree states. The algorithm calls for two sets of labelled training data, A and B. One first grows an overtrained tree on set A, prunes it by evaluating how well it predicts set B, then grows it again using set B, prunes it using set A, and so on until two successive pruned trees are identical. In practice, we have observed that two expansion-pruning cycles are usually sufficient.

1.1. Growing Tree States

Recall that to grow a decision tree on a set of labelled training data items, one must supply three elements:

- a set of possible yes-no questions;
- a rule for selecting the best question at a node;
- a method for pruning trees to prevent over-training.

The set of possible yes-no questions is application-dependent. In the case of acoustic models, the questions are usually about the identity of the phones surrounding the central phone, and can be grouped in classes (see section 1.4), but they can also be about anything judged relevant by the researchers.

The rule for selecting the best question at a node is based on the idea that the ''yes'' and ''no'' models generated when the training data in a node is split by the question should predict this training data as well as possible. If L(P) is the log likelihood that the parent node generated the training data, and L(Y) and L(N) are the corresponding ''yes'' and ''no'' log likelihoods, the question chosen is the question which maximizes

$$\Delta L = L(Y) + L(N) - L(P)$$

where

$$L(X) = -\frac{1}{2} c_x \left[n + \log\left((2\Pi)^n |\Sigma(X)| \right) \right]$$

with c_x the occupancy counter of the node, and n the number of parameters of the data [2].

Stopping criteria can be applied to prevent the trees from being overtrained. We use two stopping criteria: we impose a minimum value for c_x, and we split a node only if ΔL for the best question is greater than a specified threshold.

1.2. Pruning Tree States

The pruning step consists in evaluating how well the tree grown predicts new data, and pruning the nodes which do not perform well. This is done by a bottom-up procedure which compares, at each node X, how well X predicts the new data B and how well the subtree rooted at X predicts this same new data. The subtree rooted at X is kept if it predicts B better than X does. The prediction performance of node X on set B, if X was originally trained from set A, is calculated by

$$L_B(X) = \frac{-c_{AB}}{2} \sum_{i=1}^{N} \left(\log 2\Pi\sigma^2_{A,i} + \frac{\sigma^2_{AB,i}}{\sigma^2_{A,i}} \right)$$

where

$$c_{AB} = \sum_n \sum_t 1$$

$$\sigma^2_{AB,i} = \frac{d^2_{B,i}}{C_B} - 2\mu_{A,i}\mu_{B,i} + \mu^2_{A,i}$$

with n a training utterance and t a frame, i=1,..., N and

$$d^2_{B,i} = \sum_n \sum_t b^2_{nt,i}$$

$$\mu_{X,i} = \frac{1}{nt} \sum_n \sum_t x_{nt,i}$$

where $x_{nt,i}$ is the value of the i^{th} parameter for frame t of the n^{th} utterance of data set X.

2. VARIANTS ON TREE STATES

We have carried out experiments on both the ATIS corpus, with 16733 training utterances and 1001 test utterances, and the WSJ 284 corpus, with 36367 training utterances and 215 test utterances. Except where specified, the stopping criteria for the expansion of the trees are kept constant throughout this paper.

2.1. Making Unitary Models

Initially, we grew tree states from triphone data. The amount of information contained in triphones is limited: all that is known about the context of the central phone is the identity of the phones immediately surrounding it. In these experiments, not all triphones occurring in the training data were kept in the final set of triphones: the triphones with too low a number of occurrences were eliminated, thus resulting in a loss of information.

To solve both problems, we devised a different way to represent the information, which we called "unitary models". There is one unitary model for each state of each different context of a phone. Any information judged relevant to the realization of a phone and available when doing the segmentation can be included in the models, since they are calculated starting from the labelling. Thus, the word "context" takes on a broader meaning than the *phonetic* context usually used when working with triphones. The stress associated with a phone and the position of the phone within the word are examples of information that can be part of the context of a phone for unitary models. The phonetic context span can also be broadened to an arbitrary width.

To date, we have limited the phonetic context span to the usual immediately preceding and following phones, and we have not considered the information on stress. However, the first thing we were interested in measuring was the effect of including all of the occurrences of a phone, so we started by doing experiments with the same information as for triphones. Note that because of the good results obtained (below), all other experiments used unitary models.

The difference between the results obtained with triphones and with unitary models can also be explained by the use of a state segmentation, rather than the phone segmentation used before.

Training data	# HMM states	W. Err
Triphones	2238	10.08
Unitary models	2102	9.72

Table 1: Results for unitary models on the ATIS corpus

Training data	#HMM states	W. Err
Triphones	6817	18.58
Unitary models	6741	18.16

Table 2: Results for unitary models on the WSJ corpus

2.2. Pruning The Trees With Various Severity Levels

According to the decision tree literature, rather than rely solely on stopping criteria to prevent overtraining, it is better to grow an over-large tree, then prune it back by examining its performance on new data. The traditional way to prune trees is to keep the children (Y and N) of a node only if the sum of their impurity (badness) measure on new data is strictly less than the impurity measure of their parent (P), i.e., if

$$L(Y) + L(N) - L(P) > 0$$

(In our context, the absolute value of the log likelihood L() is the measure of badness or impurity). However, in order to make the trees as small as possible, one may wish to have a more severe pruning criterion; for example, for a given value V, the pruning criterion would become

$$L(Y) + L(N) - L(P) > V$$

We tried different values of V, both on the ATIS and the WSJ corpora. As can be seen in the following tables, the results are quite satisfactory:

V	# HMM states	W. Err
No pruning	2102	9.72
0	1912	9.62
500	1565	9.54
1000	1311	9.74

Table 3: Pruning results on the ATIS corpus

V	#HMM states	W.Err
No pruning	6741	18.16
0	6649	18.34
500	5083	18.45
1000	3648	18.78

Table 4: Pruning results on the WSJ corpus

Note that the variation in the word error rate is small - the main interest of the technique is to cut the total number of states. Pruning the trees for ATIS with V=1000 reduces the number of states by 38% with respect to the non-pruned trees in table 1 while keeping

approximately the same error rate, and pruning the trees for WSJ with V=1000 reduces the number of states by 46%, with respect to the trees in table 2, at the price of a slight loss of performance.

2.3. Pooling Some Leaves

As was pointed out in [2], it is possible, once a tree is created, to pool some of the leaves that are similar enough according to some likelihood criterion. We have applied this idea, with the possibility of specifying the likelihood drop (D) that we are ready to accept due to the pooling of two leaves. The algorithm is thus to sort the pairs of leaves according to the likelihood decrease occurring when the pair is pooled, and to pool all the pairs for which this likelihood decrease is less than D. However, we do not allow for more than two leaves to be pooled together because we have noticed that some leaves tend to be involved in the majority of the valid pairs, and pooling all the leaves that are members of these pairs together gives worse results than pooling the leaves two by two.

The results with D=0 do not give the same results as the corresponding results without pooling because D=0 means that the pair of leaves that bring about some likelihood increase will be pooled.

V	D	# HMM states	W. Err
0	0	1748	9.53
	500	1491	9.46
	1000	1357	9.50
500	0	1490	9.58
	500	1341	9.54
	1000	1215	9.47
1000	0	1270	9.73
	500	1193	9.68
	1000	1095	9.77

Table 5: Pooling results on the ATIS corpus

V	D	# HMM states	W. Err
0	0	6408	18.29
	500	4707	18.55
	1000	4114	18.37
500	0	4973	18.24
	500	4025	18.42
	1000	3465	18.42
1000	0	3603	18.91
	500	3248	18.81
	1000	2868	18.97

Table 6: Pooling results on the WSJ corpus

It is hard to see any clear tendency in these results, but they show that it is possible to cut a lot of nodes with only a slight decrease in performance.

2.4. Refining The Questions

The set of questions that can possibly be asked at any node of a decision tree is very important. However, as the following results show, the trees seem to yield about the same error rate regardless of the set of questions chosen; these results are consistent with what we found in [1]. This time, instead of choosing three orthogonal sets roughly corresponding to well-known phonetic feature definitions as we did in [1], we tried to refine our set of questions by adding questions from other phonetic schemas, and intersecting question classes, *e.g.*, starting with the two classes LABIAL and PLOS, we would produce a third class, $\text{LABIAL} \cap \text{PLOS}$. From our original set of 12 class questions (plus the questions on single phones) called Set 1, we went to a set of 34 class questions (plus the questions on single phones) called Set 2. Note that except for tables 7 and 8, all tables of results in this paper were obtained with questions from Set 1.

Even though the new questions from Set 2 were chosen 50% more often than those also found in Set 1, and were pruned less often, their presence or absence hardly made a difference to the results (compare tables 7 and 8 with tables 3 and 4).

V	# HMM states	W. Err
No pruning	2096	9.66
0	1905	9.66
500	1552	9.41
1000	1275	9.70

Table 7: Results with question set 2 on the ATIS corpus

V	# HMM states	W. Err
No pruning	6575	18.21
0	6478	18.47
500	4854	18.13
1000	3486	18.73

Table 8: Results with question set 2 on the WSJ corpus

2.5. Questions About The Position Of The Phone Within The Word

We have tried to exploit the new possibilities offered by the use of unitary models. With unitary models, it is possible to ask questions about other features than the phonetic context. We therefore experimented with questions about the position of the phone within the word, for instance, “Is the phone the second phone of the word ?”, or, “Is the phone the third to last phone of the word ?”.

We examined the questions chosen and found out that all the questions about the position of the phone were “Is the phone the first phone of the word ?” and “Is the phone the last phone of the word ?”. These questions were picked at the nodes where, in our previous experiments, the question was “Is the phone preceded by a pause ?” and “Is the phone followed by a pause ?”, respectively. No other questions about the position of the phone were asked.

Thus, we had just found another way to ask the same thing, with, of course, the same results. We did this experiment on ATIS only and did not bother trying it on WSJ.

2.6. Lookahead Search

Picking the questions for each node the way we do (by comparing the impurity of a node with that of its children) is a local optimization of the performance of the tree. It seems plausible that if it were computationally feasible, we should globally optimize the performance, *i.e.*, choose each question according to the impact it would have on the whole tree. This is far from being possible for any reasonably sized set of data, but what *is* possible is to extend a little further the locality of the optimization done. For example, one may compare the impurity of a node with that of the eventual best grandchildren for a given question.

To our great puzzlement, this method did **not** yield improved performance. It consistently gave worse results than when we employed the standard greedy criterion. We tried many different values for the stopping criteria, to no avail. The stopping criteria used when growing the trees are the occupancy counter for a leaf, and the difference in likelihood (Δ) between a node and its children.

Here is a comparative table for some of the experiments we did, with and without the lookahead search. Note that these results are for tree states grown from triphones:

	Without Lookahead		**With Lookahead**	
Stop./ prun. crit.	**# HMM states**	**W. Err**	**# HMM states**	**W. Err**
No pruning	2238	10.08	2715	10.56
$\Delta = 800$	1712	10.06	2150	10.15
$V = 0$	2068	9.99	2505	10.44

Table 9: Results for the ATIS corpus with lookahead search

The time taken by the lookahead heuristic was about 2.5 times the time without lookahead.

Subsequently, we encountered some fascinating recent work from the general classification tree literature (*i.e.*, from outside the speech recognition world) that make our results more explicable [4], [5]. Based on large-scale experiments with a variety of data sets, these authors conclude that the standard greedy criterion almost always outperforms the apparently more sophisticated lookahead approach, particularly when the greedy criterion is used to generate a tree that is subsequently pruned (as in all our work). Apparently, lookahead produces a type of overtraining.

2.7. Making A Single Tree For Each Phone

When creating tree states, we grow a separate tree for each of the 3 states of each phone. However, it is possible that for a given phone, the main factors that affect the realizations of the phone are the same for all the states. To experiment with this idea, we built only one tree for each state, but allowed questions about the state number ("Is this data item for state #1 of the phone ?"). What we observed when examining the trees produced was that for 37 of the 40 trees, the question at the root was a question about the state number; for most of the trees, there was another question on the state number at the second level. In other words, the single tree for a given phone was nothing more than the three previous trees joined by two nodes.

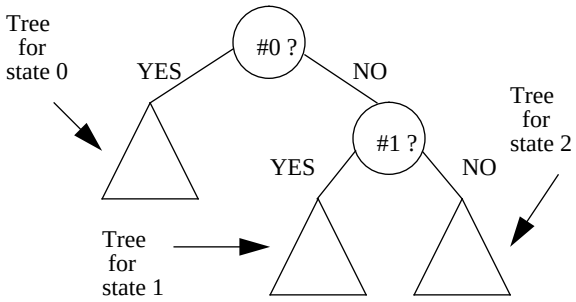

Figure 1: Typical single tree for a phone

The difference between each single trees and the corresponding set of three trees was so small that we did not try to perform recognition using them.

3. FUTURE WORK

Our future work involves trying trees with questions about stress, which we have not done yet because of the modifications needed in the segmentation module. We would also like to try something along the lines of maximum mutual information estimation applied to trees: the idea would be to grow a set of trees the usual way, then to iteratively grow new sets of trees by choosing the question at each node for which the YES and NO children have the least probability of being confused with models for other phones.

We have also undertaken work on French, on which we will concentrate in the near future.

4. REFERENCES

[1] R.Kuhn, A. Lazaridès, Y. Normandin, and J. Brousseau, "Improved Decision Trees for Acoustic Modeling", *Proc. ICASSP 95, Vol. 1, p. 552-555*, May 1995.

[2] S.Young, J.J. Odell, and P. Woodland, "Tree-Based State-Tying for Acoustic Modeling", *ARPA Workshop on Human Language Technology*, pp. 286-291, Mar. 1994.

[3] S. B. Gelfand, C.S. Ravishankar, and E. J. Delp, "An Iterative Growing and Pruning Algorithm for Classification Tree Design", *IEEE PAMI, Vol. 13, No. 2*, February 1991.

[4] S. Murthy and S. Salzberg, ''Lookahead and Pathology in Decision Tree Induction'', *IJCAI-95, V. 2*, pp. 1025-1031, August 1995.

[5] S. Murthy, ''On Growing Better Decision Trees from Data'', *Ph.D. thesis*, Johns Hopkins University, 1995.