Proceedings of the Institute of Acoustics

A DECISION TREE APPROACH TO TASK-INDEPENDENT SPEECH RECOGNITION

Simon Downey and Martin Russell

Speech Research Unit, DRA Malvern, Malvern, Worcs WR14 3PS, UK

1. INTRODUCTION

It is widely accepted that in order to achieve high recognition accuracy with a phoneme-level hidden Markov model (HMM) based speech recognition system it is necessary to use context-sensitive models. The standard solution to this problem is to use triphone HMMs [12], in which it is assumed that the only significant contextual effects on the acoustic realisation of a particular phoneme are due to the immediately preceeding and succeeding phonemes. For example the phoneme S[1] in the word */bUkSQp/* (bookshop) is represented as the triphone *(S:k_Q)* corresponding to S in the context of k and Q.

There are a number of problems associated with this approach to context modelling. Firstly, the basic triphone assumption is incorrect, since the acoustic realisation of a phoneme may be more strongly influenced by contexts other than its immediate neighbours. In the example given above the preceeding U may have a significant effect on the realisation of the S, but this effect cannot be accomodated in the triphone approach. If contexts other that immediate neighbours are considered in order to overcome this problem then the number of models becomes too large, resulting in the necessity for impractically large training sets for robust model parameter estimation. Indeed, even if attention is restricted to triphones the number of models may be prohibitively large in this respect for significant vocabularies. The second problem is that the triphone approach contains redundancy. Different triphone contexts which may have the same influence on the realisation of a particular phoneme are allocated separate models, which is wasteful in terms of both computation and use of training material. Finally, in the case of task independent speech recognition where the goal is to produce a set of "vocabulary independent" phoneme level HMMs using a corpus of general speech and subsequently to map these models onto specific tasks with no further training, there is the problem of choosing a model for a particular triphone in the application vocabulary which does not exist in the model set. This occurs because there are insufficient examples of that particular triphone context in the training set to reliably estimate the parameters of a triphone HMM.

One approach to the complementary problems of undertraining and redundancy in triphones is the use of clustered, or "generalised", triphones [4, 5, 8]. However, this method leaves two problems unresolved : first, the measure of distance between pairs of triphone HMMs is typically based on differences between the state parameters (the mean and the covariance matrix) of the two models. But these parameters are unreliable for precisely those models which need to be put into equivalence classes because of lack of training data. The second problem is that it is not clear which cluster a new triphone, which was not included in the clustering process, should be assigned to.

This paper describes a decision tree based approach to modelling phonemes in context which overcomes some of the limitations described above. The basic technique is taken from [3]. A given phoneme is associated with a single binary decision tree. In principle the terminal nodes of this tree correspond to equivalence classes of contexts which have the same influence on the acoustic realisation of that phoneme, and a context-sensitive HMM is built for each of these terminal nodes.

The phoneme decision trees are used to create pronunciation dictionaries for new vocabularies in terms of these HMMs. The properties of the trees permit models to be chosen for contexts which do not occur in the training set, thus providing a "vocabulary independent", or "task independent", capability. Methods for constructing such decision trees, using them to create a general model set, and creating task-specific pronunciation dictionaries are described here. The results of recognition experiments based on these methods are also presented. Comparisions are drawn with the best results obtained using conventional triphone models.

[1] The SAM Phonetic Alphabet (SAMPA) [13] is used throughout this memorandum

2. PHONEME DECISION TREES

2.1 Decision Tree Format

Figure 1 shows an example of a phoneme decision tree for the phoneme /aU/. Each phoneme has its own binary tree and associated with each 'terminal' node of the tree there is a context sensitive HMM. The purpose of the tree is to determine which of these HMMs should be used to model that phoneme in a particular context. For each node in the tree there is a phoneme question set (shown in square brackets), a position and two successor nodes. This set consists of phonemes which, in that relative position, have a similar effect on the acoustic realisation of the decision tree phoneme. For example, node 2 in figure 1 is associated with the question "is the next (relative position 1) phoneme one of the set *[p, t ,k, s, S, f, T, h, tS]* (i.e. a voiceless consonant)". For every occurrence of a phoneme in a new word, it's tree questions are applied in turn to the neighbouring phonemes, starting at node 1. If a neighbouring phoneme in the correct relative position is one of those in the particular question set, the tree is descended along the left hand '*Yes*' path. Otherwise the right hand '*No*' path is followed. Descending the tree represents categorising the contexts into smaller and smaller sub-sets, and in the limit, the terminal nodes of the tree would represent the phoneme in a unique context. If these contexts are constrained to be immediate, they would correspond to triphones.

The construction of the trees and the estimation of HMMs for each terminal node are dealt with in Section 3.

Decision Tree For The Phoneme aU

Number of Nodes = 7

Node Number = 1	[t d n s Z T D l r]	1	2	3
Node Number = 2	[p t k s S f T h tS]	1	4	5
Node Number = 3	[I e { V @ Q]	1	6	7

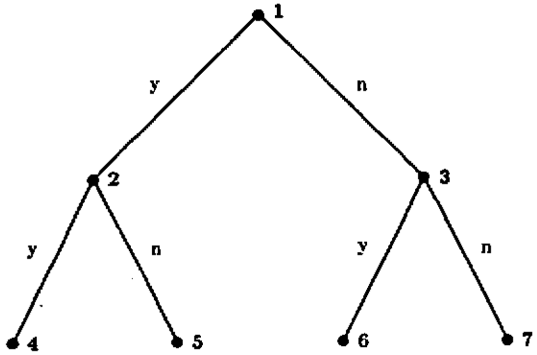

Figure 1: *Example of a Binary Decision Tree.*

2.2 An Example of the Decision Tree Operation

To illustrate how the tree is used in selecting an appropriate model for a phoneme in a particular context, consider the problem of selecting the correct model for the diphthong /aU/ in the word 'about', transcribed phonemically as */@baUt/*, using the decision tree in figure 1. Starting at node 1, the question is "is the phoneme on the immediate right (relative position 1) of *aU* one of the set *[t d n s z T D l r]* (i.e. an alveolar consonant). The phoneme '*t*' is in this set, hence a positive answer is obtained. The successor node for the answer Yes is node 2, corresponding to the question "is the phoneme on the immediate right an unvoiced consonant. Again the answer is yes, leading to node 4. Node 4 is a terminal node and hence corresponds to an equivalence class of contexts which, according to the decision tree, have the same effect on the realisation of the phoneme aU. The model associated with this node is therefore used to model the phoneme aU in "about".

Application of the other phoneme decision trees to the remaining contexts in the word allows it to be written as a sequence of terminal node HMMs.

3 DECISION TREE GENERATION

A data driven method is used to determine the content and ordering of questions in each decision tree. The algorithm is based on [3], it is sequential and non-optimal in terms of the obvious optimality criteria which might be applied.

3.1 Speech Data, Annotation and Question Sets

The basic ingredients of decision tree generation are a corpus of training speech which is annotated at the phoneme level, and the phoneme set and allowed range of relative position components of the questions.

3.2 The Node-Splitting Algorithm

Initially the decision tree for a given phoneme has a single node N_1 (the root node), and the set S_1 of all segments of acoustic speech pattern corresponding to that phoneme are associated with that node. The node-splitting algorithm is applied once to each node of the tree until no further splitting is possible. Each successful application of the node-splitting algorithm generates two new nodes which become new candidates for splitting. When a new node cannot be split it is designated a terminal node. When all unsplit nodes are terminal nodes the splitting process ends.

3.3 Definition of the Node Splitting Algorithm

Suppose that N_i is a node and that it is associated with a set S_i of acoustic segments. Each question Q defines a partition of S_i into two disjoint sets $S_i(Q)$, the set of segments in S_i whose phonemic contexts return a positive answer to question Q, and $S_i(\bar{Q})$, the set of segments whose phonemic contexts return a negative answer to Q. Clearly $S_i(Q) \cup S_i(\bar{Q}) = S_i$. For each pair of segments x and y in $S_i(Q)$, let $D(x,y)$ denote the dissimilarity between x and y. In the experiments described in this paper the dissimilarity measure D is the cummulative distance between x and y, obtained from a standard dynamic programming based template matching algorithm [11].

Define

$$D_{pos}(Q) = \sum_{x,y \in S_i(Q)} D(x,y),\ D_{neg}(Q) = \sum_{x,y \in S_i(\bar{Q})} D(x,y). \qquad (1)$$

If $D_{pos}(Q)$ is small, then the differences between pairs of acoustic segments in the set $S_i(Q)$ are (on average) small, hence the phonemes in the phoneme set corresponding to Q in the position corresponding to Q have a similar acoustic effect on the acoustic realisation of the given phoneme. Similarly, if $D_{neg}(Q)$ is small then the set of phonemes which are not in the phoneme set corresponding to Q in the position corresponding to Q have a similar contextual effect on the given phoneme.

Hence the "best" partition of the node N_i is that which is defined by the question Q for which the quantity $D_Q = D_{pos}(Q) + D_{neg}(Q)$ is minimised. Define

$$Q = argmin_Q D_Q \qquad (2)$$

The splitting algorithm assigns the question $\hat{Q}$ to node N_i and associates the successor nodes, $N_i(\hat{Q})$ and $N_i(\bar{\hat{Q}})$, of N_i with the acoustic segment sets $S_i(\hat{Q})$ and $S_i(\bar{\hat{Q}})$ respectively.

The node-splitting algorithm is initially applied to the root node N_1. Each new node N_i is split unless the stopping criterion in section 3.4 is satisfied, in which case N_i becomes a terminal node. The process stops when all nodes which have not been split are terminal nodes. Notice that the splitting process is implicitly biased towards partitions which associate similar numbers of segments to the sets $S_i(Q)$ and $S_i(\bar{Q})$. This is because an equal split minimises the total number of terms in the summations in (1). An alternative definition of D_Q in which each of the terms $D_{pos}(Q)$ and $D_{neg}(Q)$ is normalised by the number of terms in the summation could be used.

3.4 Stopping Criteria
The current implementation of the node splitting algorithm includes a user specified minimum node size parameter M_{min}. If when a question Q is applied to node N_i either $|S_i(Q)| < M_{min}$ or $|S_i(\bar{Q})| < M_{min}$ then the question Q is disallowed. If all questions are disallowed then N_i cannot be split and becomes a terminal node.

The parameter M_{min} ensures that each terminal node of a decision tree is associated with sufficient examples in the training set to enable the parameters of a HMM to be reliably estimated.

3.5 Terminal Node Model Estimation
It is straightforward to associate each terminal node of a decision tree with a trained HMM. The standard of the Baum-Welch parameter estimation algorithm for sub-word HMMs requires the following ingredients:

- An initial estimate of the parameters of each HMM,
- A set of training utterances annotated orthographically, normally at the sentence or phrase level, and
- A pronunciation dictionary which expresses each word in the application vocabulary as a sequence of phoneme-level labels which correspond to terminal nodes of a decision tree, and hence models in the HMM set.

An initial estimate of the parameters of each context sensitive HMM is obtained from the corresponding monophone HMM. The pronunciation dictionary is generated by applying the procedure described in section 2.2 to each vocabulary word.

4. EXPERIMENTAL METHOD

The experiments were designed to test two properties of the decision tree based approach: (i) the ability of this approach to overcome the undertraining and redundancy problems associated with the conventional triphone based approach, and (ii) the ability to cope with new, unseen vocabularies. To investigate (i) a "task-dependent" experiment was conducted using the Airborne Reconnaissance Mission (ARM) task [9]. The experiment compares the performance obtained using context-sensitive HMMs derived using the decision tree method with that of conventional triphone HMMs on the ARM task. To investigate (ii) a "task-independent" experiment was conducted in which phoneme decision trees and the associated context sensitive HMMs were constructed using a corpus of general English speech. The performance of these models was compared with that of task-dependent models on the ARM task. In addition, both sets of decision trees (together with their associated HMMs) were evaluated on an Air Traffic Control (ATC) task.

4.1 Speech Data
4.1.1 Test Data. Recognition experiments were performed on two test sets, referred to as SI-ARM and ATC. The SI-ARM test set consists of recordings of 10 male subjects each speaking 3 ARM reports. This is the evaluation set from [10]. The ATC test set comprises recordings of 100 ATC sentences, 10 each from the same 10 male speakers. Both test sets are taken from the "SI89" corpus [2].

4.1.2 Training Data. Two sets of training material were used in the experiments described in this paper: the SI-ARM training set, and the SRU-SCRIBE training set. The former consists of speech from 61 male speakers, each of whom spoke 3 complete ARM reports [9], and the latter of speech from 230 male speakers, each of whom spoke 10 sentences based on SCRIBE sentence list B [14]. All recordings are taken from the "SI89" speech corpus [2].

4.1.3 Pre-Processing. All of the speech was preprocessed using the SRUbank filterbank analyser in its deault configuration (27 filters spanning frequencies up to 10kHz, 100 frames per second), followed by variable frame-rate analysis and a cosine transform. The final parameterisation is the $CC8\delta$ "delta-cepstrum" representation from [10].

4.2 Decision Tree Generation
4.2.1 Annotation of the Training Sets. For each training set, phoneme level annotation was obtained using a forced recognition process based on a set of conventional triphone HMMs trained on that set. The forced labelling process requires the speech to be labelled orthographically at the sentence level. Word-level HMMs are constructed for each word in a sentence by concatenation of the appropriate triphone models according to a pronunciation dictionary. The word-level models are then concatenated in turn to produce a sentence-level HMM, which is mapped onto the speech data using a conventional dynamic programming based alignment algorithm. The phoneme end points are then recovered by decoding the optimal state sequence. This makes an implicit assumption that although the triphones set may be undertrained and therefore unable to generalise to unseen data, they are adequate for forced labelling of the training set.

4.2.2 The Question Sets. The phoneme set components of the questions were chosen according to standard phonetic theory of contextual influences in speech at the phoneme level. In the case of the consonants, phoneme sets were chosen according to place of articulation (labial, alveolar, palato-alveolar or velar) and manner (voiced or unvoiced). Vowels were grouped according to tongue position (front, centre or back) and length (long or short, with "long" including all diphthongs). Diphthongs were grouped according to tongue position with respect to first element (front, centre or back) and second element (fronting, centring or backing). In addition, in order to enable triphones to be used in cases where there are sufficient training examples and to allow for unpredicted contextual effects, each individual phoneme was included as a phoneme set in its own right. In this experiment, only immediate left and right contexts (positions -1 and 1) were used in the construction of the trees. A complete list of phoneme sets chosen is included in [7].

A set of phoneme decision trees was produced for each of the two training sets using the construction algorithm described in section 3 with a minimum node size of 50. This ensures that there are at least 50 training examples for each context-sensitive HMM. The SRU-SCRIBE phoneme decision trees are shown in full in [7].

4.3 Context-Sensitive HMM Set Generation
HMMs corresponding to the terminal nodes of the phoneme decision tree were obtained as described in section 3.5, using the same training sets that were used to construct the decision trees. For HMM parameter estimation these sets were labelled orthographically at the sentence level. All of the phoneme-level HMMs have 3 single multivariate gaussian states with diagonal covariance matrices. Two sets of HMMs were estimated for each set of decision trees, one with state specific covariance matrices, and the other with a single shared "grand" covariance matrix [6]. This was done because previous experiments had shown that in the case of the speaker dependent ARM system, which used approximately 1500 conventional triphone HMMs, best perfomance was achieved with a shared covariance matrix [9], while with smaller model sets best performance is achieved with state specific covariance matrices [16]. The current experiments use 318 models (SI-ARM decision trees) and 416 models (SRU-SCRIBE decision trees).

4.4 Recognition Experiments.
Once the context sensitive HMMs have been trained, they can be used to recognise words from a given vocabulary. The decision trees are applied to each word in the vocabulary to produce an appropriate pronunciation dictionary in terms of their terminal nodes. Even if there are contexts in the new vocabulary which did not occur in the original training data, application of the tree questions will lead to a terminal node model for that context. The pronunciation dictionary and model set are then downloaded onto a recognition engine, to provide a task specific recogniser.

In the current experiments recognition uses the conventional one-pass algorithm with beam search and partial traceback [1].

5. RESULTS

Table 1 shows the results of recognition experiments performed using the SI-ARM decision tree and context sensitive model sets applied to the SI-ARM and ATC test sets. The table includes results for models with state-specific and "grand" covariance matrices. All experiments were conducted with a word insertion penalty of 30 [15] and no explicit syntax. The table also shows the result for conventional triphones with grand variance trained on the same training set, allowing a direct comparision. A number of interesting conclusions can be drawn. Concentrating on the % word accuracy column, for a word transition penalty of 30 the decision tree models score slightly (but not significantly) better than the triphone models, but with a much smaller model set. This is consistent with results obtained using clustered triphones presented in [8]. Note that the triphone models have a common grand covariance matrix, whereas the best SI-ARM decision tree based models retain their individual state covariances. Indeed, it can be seen that performing a grand variance calculation on these models reduces the performance, as predicted in [16]. For the ATC test set the performance drops. This is consistent with results from other laboratories and reflects the fact that in a constrained task like ARM, phoneme models which are nominally concerned only with immediate left and right contexts actually take on much wider vocabulary-specific characteristics which do not generalise to other vocabularies.

The results obtained with the SRU-SCRIBE decision tree based models are shown in table 2. For the SI-ARM task performance is significantly worse than that obtained with the SI-ARM decision trees and models, again reflecting the fact that the latter method models effects due to contexts wider than immediate neighbours. The performance on the ATC task is also worse than that obtained with the SI-ARM decision trees and models. This can be explained by the overlap between the ARM and ATC vocabularies.

ARM TRAINING SET	ARM DATA TEST SET			ATC DATA TEST SET	
	TRIPHONE MODELS WP=30, GV	DECISION TREE WP=30, GV	DECISION TREE WP=30	DECISION TREE WP=30, GV	DECISION TREE WP=30
%ACCURACY	74.7	70.8	74.9	46.3	48.6
%CORRECT	79.3	75.1	79.8	54.8	56.8
%INSERTIONS	4.6	4.3	4.9	8.6	8.2
%DELETIONS	9.9	10.9	8.8	14.3	14.8

Table 1: *Results of experiments performed using the SI-ARM decision tree*

5. CONCLUSIONS

The baseline result of 74.9% word accuracy on the SI-ARM evaluation set using the SI-ARM decision tree shows that decision tree modelling can provide good recognition performance with many less models than equivalent triphone methods. To obtain the results in table 1, 1494 triphones were used compared to only 318 decision tree models. The "task independent" results obtained on the ARM vocabulary with the SRU-SCRIBE decision trees are disappointing. In principle one would expect the SRU-SCRIBE models to generalise well, since the SCRIBE sentences were chosen to give good coverage of phonemes in context. However, it is clear that the models associated with the SI-ARM decision trees, although nominally generalised triphones, are

SCRIBE TRAINING SET	ARM DATA TEST SET		ATC DATA TEST SET	
	DECISION TREE WP=30	DECISION TREE WP=30, GV	DECISION TREE WP=30	DECISION TREE WP=30, GV
%ACCURACY	52.1	50.6	45.4	40.9
%CORRECT	60.8	57.9	55.1	50.2
%INSERTIONS	8.8	7.3	9.8	9.3
%DELETIONS	12.0	13.2	16.6	17.7

Table 2: *Results of experiments performed using the SRU-SCRIBE decision tree*

modelling vocabulary specific contextual influences which extend beyond immediate neighbours, and that it is this vocabulary-specific context modelling which leads to good performance.

Since the basic HMMs are of the same type for both the SI-ARM and SRU-SCRIBE decision trees (3 single multivariate gaussian states with diagonal covariance matrices), the results suggest that task-independent performance can be improved by the use of more comprehensive contextual modelling. This suggests that further experiments should be conducted to exploit the ability of the decision tree method to consider contexts wider than immediate neighbours. However, this implies a need for a larger number of context-sensitive models and more training material to represent a richer set of contexts.

References

[1] J S BRIDLE, M D BROWN & R M CHAMBERLAIN, "A One-Pass Algorithm for Connected Word Recognition", Proc ICASSP'82, Paris, 1982.

[2] S R BROWNING, J MCQUILLAN, M J RUSSELL & M J TOMLINSON, "Texts of Material Recorded in the SI89 Speech Corpus", SP4 Research Note number 142, RSRE, February 1991.

[3] L R BAHL, P V DE SOUZA, P S GOPALAKRISHNAN, D NAHAMOO, & M A PICHENY, "Decision Trees for Phonological Rules in Continuous Speech", Proc ICASSP'91, Toronto, 1991.

[4] K-F LEE, "Large Vocabulary Speaker-Independent Continuous Speech Recognition: the SPHINX System", PhD Thesis, Carnegie Mellon University, 1988.

[5] D B PAUL, "Speaker-Stress Resistant Continuous Speech Recognition", Proc ICASSP'88, New York, 1988.

[6] D B PAUL, "A Speaker-Stress Resistant Isolated Word Recognizer", Proc ICASSP'87, Dallas, 1987

[7] M RUSSELL, S R BROWNING & S DOWNEY, "A Method for the Construction of Phoneme Decision Trees", RSRE Memorandum 4666

[8] M J RUSSELL, K M PONTING, S R BROWNING, S DOWNEY & P HOWELL, "Triphone Clustering in the ARM System", RSRE Memorandum 4357, February 1990.

[9] M J RUSSELL, K M PONTING, S M PEELING, S R BROWNING, J S BRIDLE & R K MOORE, "The ARM Continuous Speech Recognition System", Proc. IEEE ICASSP 1990, Albuquerque, New Mexico, April 1990.

[10] M J RUSSELL, "The Development of the Speaker Independent ARM Continuous Speech Recognition System", RSRE Memorandum 4473, January 1992.

[11] H SAKOE & S CHIBA, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", IEEE Trans. Acoustics Speech and Signal Processing, ASSP-26, No 1, Feb 1978, 43-49.

[12] R M SCHWARTZ, Y L CHOW, O A KIMBALL, S ROUCOS, M KRASNER & J MAKHOUL, "Context dependent modelling for Acoustic-Phonetic Recognition of Continuous Speech", Proc. ICASSP'85, Tampa, April 1985.

[13] J WELLS et al., "Specification of SAM Phonetic Alphabet (SAMPA)" Included in: P WINSKI, W J BARRY & A FOURCIN (Eds), "Support Available from SAM Project for other ESPRIT Speech and Language Work", The SAM Project, Department of Phonetics, University College London.

[14] "SCRIBE-Spoken Corpus Recordings In British English : Text of speech material" SCRIBE Document SCRIBE-23, *available from the Speech Research Unit*

[15] K M PONTING & S M PEELING, "Word Transition Penalties in the ARM Continuous Speech Recognition System", RSRE Memorandum No. 4362, December 1990.

[16] M J RUSSELL & K M PONTING, "Recent Results from the ARM Continuous Speech Recognition Project", Proc DARPA Speech and Natural Language Workshop, June 1990.