Tree-Based State Tying for High Accuracy Acoustic Modelling

S.J. Young, J.J. Odell, P.C. Woodland

Cambridge University Engineering Department
Trumpington Street, Cambridge, CB2 1PZ, England

ABSTRACT

The key problem to be faced when building a HMM-based continuous speech recogniser is maintaining the balance between model complexity and available training data. For large vocabulary systems requiring cross-word context dependent modelling, this is particularly acute since many such contexts will never occur in the training data. This paper describes a method of creating a tied-state continuous speech recognition system using a phonetic decision tree. This tree-based clustering is shown to lead to similar recognition performance to that obtained using an earlier data-driven approach but to have the additional advantage of providing a mapping for unseen triphones. State-tying is also compared with traditional model-based tying and shown to be clearly superior. Experimental results are presented for both the Resource Management and Wall Street Journal tasks.

1. INTRODUCTION

Hidden Markov Models (HMMs) have proved to be an effective basis for modelling time-varying sequences of speech spectra. However, in order to accurately capture the variations in real speech spectra (both inter-speaker and intra-speaker), it is necessary to have a large number of models and to use relatively complex output probability distributions. For example, to achieve good performance in a continuous density HMM system, it is necessary to use mixture Gaussian output probability distributions together with context dependent phone models. In practice, this creates a data insufficiency problem due to the resulting large number of model parameters. Furthermore, the data is usually unevenly spread so that some method is needed to balance model complexity against data availability.

This data insufficiency problem becomes acute when a system incorporating cross-word context dependency is used. Because of the large number of possible cross-word triphones, there are many models to estimate and a large number of these triphones will have few, if any, occurrences in the training data. The total number of triphones needed for any particular application depends on the phone set, the dictionary and the grammatical constraints. For example, there are about 12,600 position-independent triphones needed for the Resource Management task when using the standard word pair grammar and 20,000 when no grammar is used. For the 20k Wall Street Journal task, around 55,000 triphones are needed. However, only 6600 triphones occur in the Resource Management training data and only 18,500 in the SI84 section of the Wall Street Journal training data.

Traditional methods of dealing with these problems involve sharing models across differing contexts to form so-called *generalised triphones* and using *a posteriori* smoothing techniques[5]. However, model-based sharing is limited in that the left and right contexts cannot be treated independently and hence this inevitably leads to sub-optimal use of the available data. *A posteriori* smoothing is similarly unsatisfactory in that the models used for smoothing triphones are typically biphones and monophones, and these will be rather too broad when large training sets are used. Furthermore, the need to have cross-validation data unnecessarily complicates the training process.

In previous work, a method of HMM estimation has been described which involves parameter tying at the state rather than the model level[10, 12]. This method assumes that continuous density mixture Gaussian distributions are used and it avoids *a posteriori* smoothing by first training robust single Gaussian models, then tying states using an agglomerative data clustering procedure and finally, converting each tied state to a mixture Gaussian. This works well for systems which have only word internal triphone models and for which it is therefore possible to find some data for every triphone. However, as indicated by the figures given above, systems which utilise cross-word triphones require data for a very large number of triphones and, in practice, many of them will be *unseen* in the training data.

In this paper, the state tying approach is developed further to accommodate the construction of systems which have unseen triphones. The new system is based on the use of phonetic decision trees [1, 2, 6] which are used to determine contextually equivalent sets of HMM states. In order to be able to handle large training sets, the tree building is based only on the statistics encoded within each HMM state and there is no direct reference made to the original data.

This tree-based clustering is shown to lead to similar modelling accuracy to that obtained using the data-driven approach but to have the additional advantage of providing a mapping for unseen triphones[3]. State-tying is also compared with traditional model-based tying and shown to be clearly superior.

The arrangement of this paper is as follows. In the next section, the method of HMM system building using state tying is reviewed and then in section 3, the phonetic decision tree based method is described. Experimental results are presented in section 4 using the HTK speech recognition system[8, 9] for both the Resource Management and Wall Street Journal tasks. Finally, section 5 presents our conclusions from this work.

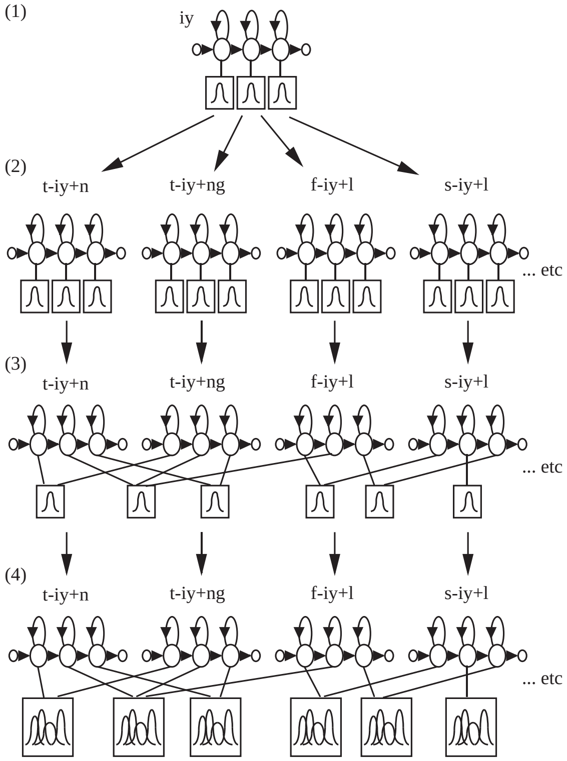

Figure 1: The Tied-State HMM System Build Procedure

2. TIED-STATE HMM SYSTEM

The aim in building a tied-state HMM system is to ensure that there is sufficient training data to robustly estimate each set of state output distribution parameters whilst retaining the important context-dependent acoustic distinctions within each phone class. The method described here uses continuous density mixture Gaussian distributions for two reasons. Firstly, continuous density models are potentially more accurate than discrete (or semi-continuous) systems since they do not require the input feature space to be quantised (or represented by only a few basis functions). This becomes particularly important when derivative features are used since discrete systems have to regard each derivative set as being statistically independent in order to achieve adequate coverage of the feature space. In continuous density systems, derivative features are simply appended to the static parameters and although it is usually necessary to make a diagonal covariance assumption, the feature sets remain coupled through a common set of mixture weights.

The second key advantage of continuous density systems is that the modelling accuracy of any particular distribution can be smoothly adjusted by increasing or decreasing the number of mixture components. This allows simple single Gaussian distributions to be used for an initial untied model set where the training data is very patchy. Then once tying has been performed such that every state has an adequate amount of data, more complex mixture Gaussian distributions can be estimated to give increased accuracy.

The process of building a tied state HMM system is illustrated by Fig. 1. There are 4 main steps

1. An initial set of a 3 state left-right monophone models with single Gaussian output probability density functions is created and trained.
2. The state output distributions of these monophones are then cloned to initialise a set of untied context dependent triphone models which are then trained using Baum-Welch re-estimation. The transition matrix is not cloned but remains tied across all the triphones of each phone.
3. For each set of triphones derived from the same monophone, corresponding states are clustered. In each resulting cluster, a typical state is chosen as exemplar and all cluster members are tied to this state.
4. The number of mixture components in each state is incremented and the models re-estimated until performance on a development test set peaks or the desired number of mixture components is reached.

In the above, all parameter estimation uses embedded Baum-Welch re-estimation for which a transcription is needed for every training utterance. Since the dictionary typically has more than one pronunciation per word, transcriptions are derived from the known orthography by using an initial bootstrap set of monophones to do a *forced recognition* of each training utterance. Since these models will be rather poor, the build procedure may need to be repeated using the models generated from the first pass to re-transcribe the training data.

As noted in the introduction, previous work on state-tying used a data-driven agglomerative clustering procedure in which the distance metric depended on the Euclidean distance between the state means scaled by the state variances. This works well but it provides no easy way of handling unseen triphones. The next section describes an alternative clustering procedure which overcomes this problem.

3. TREE-BASED CLUSTERING

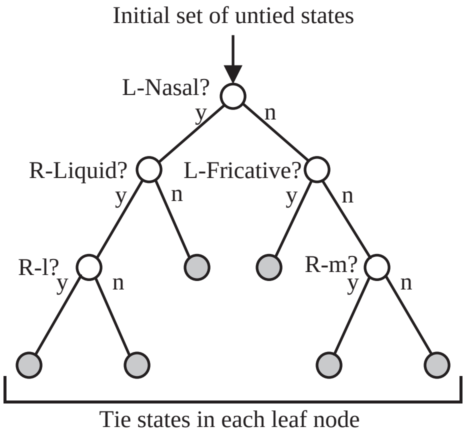

Figure 2: Example of a phonetic decision tree

A phonetic decision tree is a binary tree in which a question is attached to each node. In the system described here, each of these questions relates to the phonetic context to the immediate left or right. For example, in Fig. 2, the question "Is the phone on the left of the current phone a nasal?" is associated with the root node of the tree. One tree is constructed for each state of each phone to cluster all of the corresponding states of all of the associated triphones. For example, the tree shown in Fig. 2 will partition its states into six subsets corresponding to the six terminal nodes. The states in each subset are tied to form a single state and the questions and the tree topology are chosen to maximise the likelihood of the training data given these tied states whilst ensuring that there is sufficient data associated with each tied state to estimate the parameters of a mixture Gaussian PDF. Once all such trees have been constructed, unseen triphones can be synthesised by finding the appropriate terminal tree nodes for that triphone's contexts and then using the tied-states associated with those nodes to construct the triphone.

All of the questions used have the form "Is the left or right phone a member of the set X" where the set X ranges from broad phonetic classes such as Nasal, Fricative, Vowel, etc. through to singleton sets such as {l}, {m}, etc.

Each tree is built using a top-down sequential optimisation procedure [4, 6]. Initially, all of the states to be clustered are placed in the root node of the tree and the log likelihood of the training data calculated on the assumption that all of the states in that node are tied. This node is then split into two by finding the question which partitions the states in the parent node so as to give the maximum increase in log likelihood. This process is then repeated by splitting the node which yields the greatest increase in log likelihood until this increase falls below a threshold. To ensure that all terminal nodes have sufficient training data associated with them, a minimum occupation count is applied.

Let $\mathbf{S}$ be a set of HMM states and let $L(\mathbf{S})$ be the log likelihood of $\mathbf{S}$ generating the set of training frames $\mathbf{F}$ under the assumption that all states in $\mathbf{S}$ are tied i.e. they share a common mean $\mu(\mathbf{S})$ and variance $\Sigma(\mathbf{S})$ and that transition probabilities can be ignored. Then, assuming that tying states does not change the frame/state alignment, a reasonable approximation for $L(\mathbf{S})$ is given by

$$L(\mathbf{S}) = \sum_{f \in F} \sum_{s \in S} \log(Pr(\mathbf{o}_f; \mu(\mathbf{S}), \Sigma(\mathbf{S}))\gamma_s(\mathbf{o}_f) \qquad (1)$$

where $\gamma_s(\mathbf{o}_f)$ is the *a posteriori* probability of the observed frame $\mathbf{o}_f$ being generated by state s. If the output PDFs are Gaussian, then

$$L(\mathbf{S}) = -\frac{1}{2}(\log[(2\pi)^n|\Sigma(\mathbf{S})|] + n)\sum_{s \in S}\sum_{f \in F}\gamma_s(\mathbf{o}_f) \qquad (2)$$

where n is the dimensionality of the data. Thus, the log likelihood of the whole data set depends only on the pooled state variance $\Sigma(\mathbf{S})$ and the total state occupancy of the pool, $\sum_{s \in S}\sum_{f \in F}\gamma_s(\mathbf{o}_f)$. The former can be calculated from the means and variances of the states in the pool, and the state occupancy counts can be saved during the preceding Baum-Welch re-estimation. For a given node with states $\mathbf{S}$ which is partitioned into two subsets $\mathbf{S}_y(q)$ and $\mathbf{S}_n(q)$ by question q, the node is split using the question $q*$ which maximises

$$\Delta L_q = L(\mathbf{S}_y(q)) + L(\mathbf{S}_n(q)) - L(\mathbf{S}) \qquad (3)$$

provided that both ΔL_{q*} and the total pooled state occupation counts for both $\mathbf{S}_y(q*)$ and $\mathbf{S}_n(q*)$ exceed their associated thresholds.

Condition	Question	Total Gain
All states of all models	R-Vowel	25.9
	L-Vowel	23.3
	R-Unrounded	19.7
	L-UnFortisLenis	19.5
	R-UnFortisLenis	18.3
	R-r	17.1
Entry state of all models	L-UnFortisLenis	18.3
	L-Vowel	16.9
	L-Nasal	10.3
	L-CentralFront	7.7
	L-Unrounded	7.4
	L-Fortis	6.2
Exit state of all consonants	R-Vowel	15.2
	R-Unrounded	8.6
	R-High	4.7
	R-ee	3.9
	R-Rounded	3.7
	R-Syllabic	3.6

Table 1: Ranking of most useful questions for the WSJ task.

As a final stage, the decrease in log likelihood is calculated for merging terminal nodes with differing parents. Any pair of nodes for which this decrease is less than the threshold used to stop splitting are then merged. In practice, this reduces the number of states by 10-20% without any degradation in performance.

To gain some impression of question usage, Table 1 shows, for a typical system built for the Wall Street Journal task, the first six most useful questions calculated for all states of all models, the entry state of all models and the exit state of all consonants. The rating given is the total increase in log likelihood achieved by that question. As can be seen, the presence of a following vowel is the most important context-dependent effect. There were 202 questions in total to choose from and in the three cases 195, 182 and 152 questions, respectively were actually used in at least one decision tree.

4. EXPERIMENTS

Experiments have been performed using both the ARPA Resource Management (RM) and Wall Street Journal (WSJ) databases. Results are presented here for the 1000 word RM task using the standard word pair grammar and for 5k closed vocabulary and 20k open vocabulary WSJ test sets. All tables show the percentage word error rate.

For both databases the parameterised data consisted of 12 MFCC coefficients and normalised energy plus 1st and 2nd order derivatives. In addition, for the WSJ data, the cepstral mean was calculated and removed on a sentence by sentence basis.

The RM systems used the standard SI-109 training data and used the pronunciations and phone set (46 phones plus silence) produced by CMU and listed in [5] together with the standard word-pair grammar. The RM systems were tested on the four official evaluation test sets identified by the dates when the tests took place (Feb'89, Oct'89, Feb'91 and Sep'92).

The WSJ systems used training data from the SI84 or the SI284 data sets and the pronunciations and phone set from the Dragon Wall Street Journal Pronunciation Lexicon Version 2.0 together with the standard bigram and trigram language models supplied by Lincoln Labs. Some locally generated additions and corrections to the dictionary were used and the stress markings were ignored resulting in 44 phones plus silence.

Both 5k word and 20k word WSJ systems were tested. Four 5k closed vocabulary test sets were used. These were the Nov'92 and Nov'93 5k evaluation test sets; 202 sentences from the si_dt_s6 'spoke' development test set and 248 sentences from the si_dt_05 'hub' development test set. At 20k, three test sets were used. These were the Nov'92 and Nov'93 evaluation test sets and a 252 sentence subset of the si_dt_20 development test set. For both the 5k and 20k cases, the Nov'93 test data was used just once for the actual evaluation.

All phone models had three emitting states and a left-to-right topology. Training was performed using the HTK toolkit[11]. All recognition networks enforced silence at the start and end of sentences and allowed optional silences between words. All cross-word triphone systems used a one pass decoder that performed a beam search through a tree-structured dynamically constructed network[7]. Word internal systems used the standard HTK decoder, HVite.

4.1. Data-Driven vs. Tree-based Clustering

In order to compare top-down tree clustering with the bottom-up agglomerative approach used in previous systems, an RM system was constructed using each of the two methods. Both systems used the same initial set of untied triphones. Agglomerative data-driven clustering was then applied to create a word-internal triphone system and decision tree-based clustering was used to create a second word-internal triphone system. The cluster thresholds in each case were adjusted to obtain sys-

System	Feb'89	Oct'89	Feb'91	Sep'92
Agg D-D	4.10	4.84	3.78	8.05
Tree	3.87	4.99	3.74	7.31

Table 2: Comparison of Agglomerative Data-driven vs. Tree-based clustering using the RM task. Each recogniser used word-internal triphones, had approximately 1600 tied-states and 6 mixture components per state.

tems with approximately equal numbers of states, 1655 and 1581, respectively. After clustering, the construction of the two systems was completed by applying identical mixture-splitting and Baum-Welch re-estimation procedures to produce systems in which all states had 6 component mixture Gaussian PDFs and both systems had a total of approximately 750k parameters.

The results are shown in Table 2. As can be seen, the performance of the tree clustered models is similar to that of the agglomeratively clustered system but the tree-based models have the advantage that, were it necessary, they would allow the construction of unseen triphones.

4.2. State- vs Model-based clustering

As noted in the introduction, the traditional approach to reducing the total number of parameters in a system is to use model-based clustering to produce generalised triphones. To compare this with the state-based approach, systems of similar complexity were constructed using both methods for the RM task and the 5k closed vocabulary WSJ task. For RM, each system had approximately 2400 states with 4 mixture components per state giving about 800k parameters in total. The WSJ systems were trained on the SI84 data set and had approximately 4800 states with 8 mixture components per state giving about 3000k parameters in total.

Tables 3 and 4 show the results. As can be seen, the state-clustered systems consistently out-performed the model-clustered systems (by 3-20% and an average of 14%).

System	Feb'89	Oct'89	Feb'91	Sep'92
Model	3.71	4.58	4.19	7.03
State	3.12	3.76	3.38	6.25

Table 3: Comparison of Model-based vs. State-based clustering using the RM task. Each recogniser used cross-word triphones, had approximately 2400 tied-states and 4 mixture components per state.

System	Nov'92	si_dt_s6	si_dt_05	Nov'93
Model	7.17	10.61	12.17	11.22
State	5.90	10.33	10.73	9.89

Table 4: Comparison of Model-based vs. State-based clustering on the 5k WSJ task. Each recogniser used cross-word triphones and a bigram language model, and had approximately 4800 tied-states and 8 mixture components per state.

4.3. Overall Performance

To determine the overall performance of the tree-clustered tied-state approach, a number of systems were constructed for both the RM and WSJ tasks in order to establish absolute performance levels.

For the RM task, a gender independent cross word triphone system was constructed with 1778 states each with 6 mixture components per state. The performance of this system on the four test sets is shown in Table 5. For the WSJ task, two gender dependent cross-word triphone systems were constructed. The first used the SI-84 training set with 3820 tied-states per gender and 8 mixture components per state. The variances across corresponding male and female states were tied leading to a system with approximately 3600k parameters. The second system was similar but used the larger SI284 training set. It had 7558 tied-states per gender, 10 mixture components per state and about 8900k parameters in total. The results for the the 5k tests are shown in Table 6 and for the 20k tests in Table 7. These systems achieved the lowest error rates reported for the November 1993 WSJ evaluations on the H2-C1 and H2-P0 5k closed vocabulary tasks, and the H1-C2 20k open vocabulary task; and the second lowest on the H1-C1 20k open vocabulary task. A full description of these Wall Street Journal systems can be found in [9].

Feb'89	Oct'89	Feb'91	Sep'92
3.05	2.91	2.46	5.78

Table 5: Performance of the HTK recogniser on the RM task. It used cross-word triphones, had approximately 1800 tied-states and 6 mixture components per state.

Train/LM	Nov'92	si_dt_s6	si_dt_05	Nov'93
SI84/bg	6.58	9.13	9.67	8.67 †
SI284/bg	5.14	6.63	7.58	6.77
SI284/tg	3.19	5.27	6.09	4.90 †

Table 6: Performance of the HTK recogniser on the WSJ 5k task using bigram (bg) and trigram (tg) language models. † denotes systems used for the ARPA November 1993 WSJ evaluation.

5. CONCLUSIONS

This paper has described an efficient method of state clustering based on the use of phonetic decision trees and its use has been demonstrated in the HTK tied-state recognition system. It has been shown that tying at the state rather than the model level gives improved accuracy and that phonetic decision trees are as effective for clustering as data-driven methods but have the key advantage of providing a mapping for unseen triphones.

The overall results on both the RM and WSJ tasks indicate that the proposed approach leads to a recogniser with state-of-the-art performance but which is relatively compact and easy to construct. The method depends crucially on the use of continuous density HMMs since they provide a simple way of manipulating complexity. Initially when the data for some triphones is sparse, the use of simple single Gaussian distributions still allows reasonable parameter estimates to be made. The use of single Gaussians in the initial stages also allows very efficient tree-building since the required likelihood-based objective function can be computed without reference to the training data. However, once the amount of data per state has been increased by the state tying procedure, the single Gaussians can easily be converted to mixture Gaussians by splitting components and re-estimating. Model complexity can then be increased smoothly in this way until optimal performance is achieved.

Train/LM	Nov'92	si_dt_20	Nov'93
SI284/bg	11.08	16.17	14.35 †
SI284/tg	9.46	13.71	12.67 †

Table 7: Performance of the HTK recogniser on the WSJ 20k task using bigram (bg) and trigram (tg) language models. † denotes systems used for the ARPA November 1993 WSJ evaluation.

6. ACKNOWLEDGEMENTS

The WSJ pronunciation dictionary was provided by Dragon Systems Inc. J. Odell is funded by a SERC studentship and part of this work was funded by SERC grant GR/J10204.

References

1. Bahl LR, de Souza PV, Gopalakrishnan PS, Nahamoo D, Picheny MA (1991). *Context Dependent Modeling of Phones in Continuous Speech Using Decision Trees.* Proc DARPA Speech and Natural Language Processing Workshop, pp264-270, Pacific Grove, Calif.
2. Downey S, Russell MJ (1992). *A Decision Tree Approach to Task Independent Speech Recognition.* Proc Inst Acoustics Autumn Conf on Speech and Hearing, Vol 14, Part 6, pp181-188.
3. Hwang M-Y, Huang X, Alleva F (1993). *Predicting Unseen Triphones with Senones.* Proc ICASSP'93, Vol II, pp. 311-314, Minneapolis.
4. Kannan A, Ostendorf M, Rohlicek JR (1994). *Maximum Likelihood Clustering of Gaussians for Speech Recognition.* to appear, IEEE Trans on Speech and Audio Processing.
5. Lee K-F (1989). *Automatic Speech Recognition: The Development of the SPHINX System.* Kluwer Academic Publishers, Boston.
6. Odell JJ. (1992) *The Use of Decision Trees with Context Sensitive Phoneme Modelling.* MPhil Thesis, Cambridge University Engineering Department.
7. Odell JJ, Valtchev V, Woodland PC, Young SJ (1994) *A One-Pass Decoder Design for Large Vocabulary Recognition.* ARPA Workshop on Human Language Technology, Merrill Lynch Conference Centre, March.
8. Woodland PC, Young SJ (1993). *The HTK Continuous Speech Recogniser.* Proc Eurospeech '93, pp2207-2219, Berlin.
9. Woodland PC, Odell JJ, Valtchev V, Young SJ (1994). *Large Vocabulary Continuous Speech Recognition Using HTK.* Proc ICASSP, Adelaide.
10. Young SJ (1992). *The General Use of Tying in Phoneme-Based HMM Speech Recognisers.* Proc ICASSP, Vol 1, pp569-572, San Francisco.
11. Young SJ (1993). *The HTK Hidden Markov Model Toolkit: Design and Philosophy.* TR 152, Cambridge University Engineering Dept, Speech Group.
12. Young SJ, Woodland PC (1993). *The Use of State Tying in Continuous Speech Recognition.* Proc Eurospeech '93, pp2203-2206, Berlin.