THE GENERAL USE OF TYING IN PHONEME-BASED HMM SPEECH RECOGNISERS

S.J. Young
Cambridge University Engineering Department, England

ABSTRACT

A method of manipulating sets of HMM's by applying various kinds of parameter tying operations is described, the aim being to synthesise compact and robust context dependent models. The method is illustrated via an experiment to build a set of generalised triphone models for the TIMIT database in which triphones are constructed by joining together left and right dependent biphones. Although simple, the method results in good performance and avoids the need to train large numbers of triphones. The use of tying to increase model robustness is also investigated. Tying the centre states within triphones of the same phoneme class and tying variances within states is beneficial, but larger-scale tying of variances leads to degraded performance.

1. INTRODUCTION

The idea of *parameter tying* in Hidden Markov Model (HMM) based systems is not new. In continuous speech recognition, HMM phoneme models are effectively tied across multiple occurrences of each phoneme in each training sentence [1]. Tying all covariance matrices across all models leads to the *Grand Variance* scheme [6] and tying the means within a state leads to Richter-like distributions [7]. HMM states are often tied to achieve some desired topology or when the tied states are meant to model the same sound as in E-set recognition [10]. Finally, tying all mixtures leads to *Semi-Continuous* or *Tied Mixture* systems [4,2]. It should be noted, however, that in all of the above cases, the particular kind of tying used was introduced for some specific purpose and very often it is accompanied by a new set of reestimation formulae. This tends to obscure the fact that parameter tying in a HMM-based system is a general mechanism. Any subset of HMM parameters can be tied, tying does not significantly alter the reestimation formulae and tying does not alter the convergence properties.

Tying enables parameters to be shared between models or parts of a model and this has two principle benefits. Firstly, less training data is needed for robust parameter estimation. Secondly, storage requirements and/or computation times are reduced. Both of these can be critical when building recognisers with a large number of context dependent models.

In this paper, the use of tying as a general tool for building continuous density HMM systems is investigated. The paper is in two parts. In the first part, the basic concept of tying is reviewed and the actual tying methods needed for each specific parameter set are described. In the second part, results of a number of experiments on Phoneme Modelling using the TIMIT database are reported. The focus of these experiments is on using tying to build robust context dependent models with a minimum number of parameters.

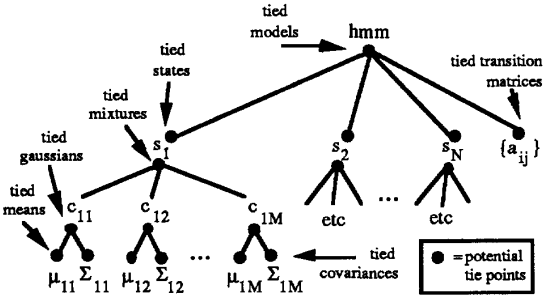

Figure 1: Hierarchical Structure of HMM Parameters

2. GENERALISED PARAMETER TYING

In this paper, we are concerned with continuous density HMM's where the probability of transition between states is given by a matrix $\{a_{ij}\}$ and the output distribution for an observation $\mathbf{o}_t$ in state j at time t is given by

$$b_j(\mathbf{o}_t) = \sum_{m=1}^{M} c_{jm}\mathcal{N}(\mathbf{o}_t; \mu_{jm}, \mathbf{\Sigma}_{jm})$$

where $\mathcal{N}(\cdot; \mu, \mathbf{\Sigma})$ denotes a Gaussian mixture with mean μ and covariance $\mathbf{\Sigma}$, and c_{jm} are the mixture weights. Notice that output distributions are tied to states and not to transitions.

All of the above parameters may be arranged in a hierarchy as shown in Figure 1. Within this hierarchy any point labelled with a large dot can be tied so that the parameters under the dot are shared. This tying can take place both within and across models. For example, in Fig 2a a single covariance matrix is shared between the mixtures of a single distribution. In Fig 2b states 1 and 2 of a set of

0-7803-0532-9/92 $3.00

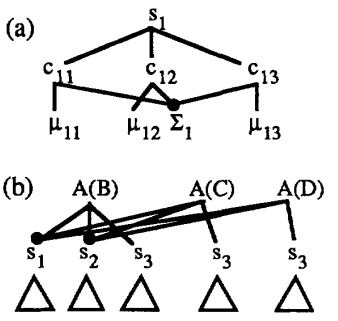

Figure 2: Examples: (a) tied covariance; (b) tied states

right context dependent models are shared across variants of the same phoneme (the notation A(B) means phoneme A in the context of phoneme B following).

The basic mechanism of tying is to firstly construct a non-tied system of models, tie the parameters which are to be shared, and then perform Baum-Welch Reestimation. This process can be repeated through several stages until a very compact set of models is produced. Of course, each tying represents an information loss and hence, care must be taken to only tie parameters where this loss is minimal.

Tying does not alter the standard reestimation formulae provided that they are implemented in the following way. The set of HMM's are represented in memory by a hierarchical data structure similar to that depicted by Fig 1 where tying is implemented simply by arranging for all participants in the tie to point to the same parameter in memory. For reestimation, all that is then needed is to directly attach the storage needed for the accumulators in the reestimation formulae to the associated parameter. Each training sentence is then processed in the usual way. However, the accumulators associated with tied parameters are multiply referenced via each of their different *owners*. The fact that tying alters neither the form nor the convergence properties of the reestimation formulae follows directly from the expansion of the auxiliary function as a summation over all time and all states [2]. In this form, the tying of any parameter can be regarded as simply partitioning the summations and does not otherwise alter the reestimation formulae.

The actual method of tying and choice of initial value depends on the parameter being tied. For the system described here, the following rules were used (refer again to Fig 1)

1. States – from the set of states to tie, the state j with the largest value of $\sum_m ln|\Sigma_{jm}|$ is chosen and all states then share this state. This criterion selects the state with the largest overall variance on the basis that this gives a reasonably robust initial estimate for subsequent reestimation.
2. Mixtures – all mixtures in the set of mixtures to be tied are pooled (not shared) upto some maximum, mixtures in excess of the maximum are discarded based on c_{jm} (tied mixtures are not considered further in this paper).
3. Gaussians – from the set of all Gaussians to tie, the mixture component with the largest value of $ln|\Sigma|$ is selected and all Gaussians then share the parameters of this mixture component.
4. Means – all means share a vector which is their average vector.
5. Covariances – assuming diagonal covariance, all covariances share a matrix whose elements are the maximum values across all of the covariances.
6. Transition Matrices – a random member of the set to be tied is chosen as a typical matrix and then shared by all in the set.

The choice of which parameters to tie is usually determined by either knowledge of the underlying classes or by clustering. Examples of both of these are given below.

3. PHONEME RECOGNITION EXPERIMENTS

In this section, results will be given to illustrate the usefulness of some of the above ideas. The application area is phoneme recognition using the TIMIT database and the goal is to produce a compact set of context dependent HMM's. The experimental conditions are similar to those established by Lee and Hon in their benchmark experiments [5]. There are 48 phonemes for recognition folded into a set of 39 for computing results. Lee and Hon report performance as a percentage of all phonemes correctly recognised (%C), the insertion rate being held at around 10%. Here a similar procedure is adopted to allow direct comparison, however, the accuracy (%A) is also given. The training set consisted of all *si* and *sx* sentences and the test set consisted of 160 *si* and *sx* sentences chosen at random. Each 16 msec window of speech is pre-emphasised, hamming windowed, and then encoded as 12 mfcc coefficients, 1 log energy coefficient, and the corresponding delta coefficients. The frame rate is 10ms. The models used are 3-state, diagonal covariance, left-to-right continuous density HMM's.

3.1. Constructing Generalised Triphones

The aim of this experiment was to construct a small set of generalised triphones for phoneme recognition using the TIMIT database. The *standard* method of generalised triphone construction is to first build models for all possible triphones and then cluster them to form the reduced set of generalised triphones. However, even using the reduced 48 phoneme set, the training section of TIMIT contains 15917 triphones and training this number of models is impractible given the limited training data.

As an alternative, the procedure summarised in Fig 3 has been investigated. Starting from the 48 context independent monophones (MONO), a set of 1690 right context dependent HMM's (RCD) and a set of 1692 left context dependent HMM's (LCD) were created using the corresponding monophone HMM's as initial values. The transition matrices were tied across all biphones from the same phoneme class and remained so for all of the experiments. Many of these biphone models were undertrained and since the complete set of HMM's contains around $550k$ parameters,

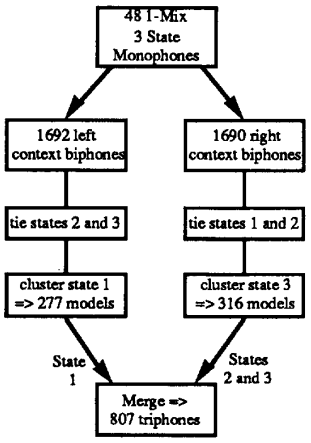

Figure 3: Stages of Triphone Construction

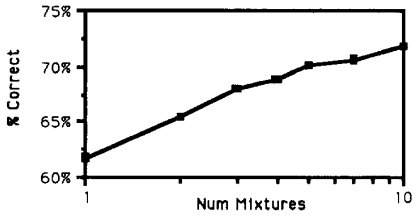

Figure 4: Monophone Performance vs Number of Mixtures

it was unwieldy to manipulate. Hence, to improve robustness and reduce the size of the LCD set, the centre states (state 2) of all models within the same phoneme class were tied and similarly all right-most states (state 3) of all models within the same phoneme class were tied. This resulted in a set of HMM's in which only the left-most state was context dependent (LCD-T). A similar operation was applied to the RCD set to give a set of models in which only the right-most state was context dependent (RCD-T). The remaining context sensitive states in each set of models were then clustered using a simple sequential furthest neighbour algorithm (LCD-C and RCD-C).

As shown in Fig 3, this resulted in approximately 300 generalised biphones for both left and right contexts. These two sets of models were then merged by combining the left-most states (state 2) of the left biphones with the centre and right states (states 3 and 4) of the right biphones. Taking all possible combinations for which there was at least 1 example in the training data yielded a set of 807 generalised triphones (TRI-1). At each each stage of the above manipulations, 3 iterations of embedded Baum-Welch re-estimation were performed.

The final step was then to untie the centre state and increase the number of mixture components to achieve better performance. Figure 4 shows the recognition performance for 3 state monophone HMM's as a function of the number of mixture components. As can be seen, performance appears to improve logarithmically with the number of mixtures. From this graph, 4 mixture components were selected for the generalised triphones (TRI-4) as a compromise between performance and model size. The method of increasing the number of mixture components was to repeatedly split the largest mixture in each state, perturb the means slightly and then perform 3 cycles of Baum-Welch re-estimation. The recognition performance, number of models, number of states and number of free parameters at each of the above stages is shown in Table 1.

	#model	#state	#param	%C/%A
MONO	48	144	23k	61.7/52.7
LCD-T	1692	1788	93k	67.5/58.8
RCD-T	1690	1786	93k	67.9/58.3
LCD-C	277	373	19k	64.6/54.9
RCD-C	316	412	21k	64.7/55.4
TRI-1	807	1614	84k	67.1/56.4
TRI-4	807	2421	514k	73.7/59.9

Table 1: Results for Triphone Construction

3.2. Increasing Model Robustnes

As can be seen from Table 1, the final triphone system gives reasonable performance but it is rather large. Furthermore, many of the variances have been floored suggesting severe undertraining. To investigate the effectiveness of parameter tying in increasing model robustness and reducing model size the following stages of tying were applied to the triphone system.

1. tie centre states within each phoneme class
2. tie variances within left and right states of each individual model so that each set of 4 mixture components share the same variances
3. as for 2 but tie across all models within the same phoneme class
4. tie all variances across all states of all models to give a grand variance system

Table 2 shows the performance, number of states and number of free parameters for each of the above tyings.

	#states	#params	%C/%A
4 Mix Triphones	2421	514k	73.7/59.9
1) Tie centre S	1662	356k	72.7/61.6
2) Tie V in model	1662	212k	72.5/61.7
3) Tie V in phone	1662	182k	71.9/61.6
4) Tie V in system	1662	180k	68.9/56.8

Table 2: Results for Various Parameter Tyings

3.3. Implementation

All of the above operations were performed using a portable software toolkit called *HTK* [11]. This toolkit is written entirely in ANSI C, runs on any 32bit machine and is available from the author.

4. DISCUSSION

The construction of generalised triphones by joining compact biphones appears to work reasonably well. The final set of 4 mixture models gives 73.7% correct and 59.9% accurate without any parameter smoothing. These are comparable to the existing published results on TIMIT using HMM's [5,3]. However, they still fall short of those achieved by Robinson using a recurrent neural network[8]. Also, it is interesting to note that 10 mixture monophones give 71.9% correct and 62.8% accurate and this is comparable to the triphone result. One area of weakness in the current procedure is that a rather simplistic sequential algorithm is used to cluster the context dependent biphone states and this needs further work.

The results on tying to increase model robustness and reduce the model size were a little disappointing. It appears that tying the centre state of 3 state triphones has little detrimental effect on performance whilst saving on both storage and computation. Tying the variances of all mixture components in a single state did not degrade performance and gave some reduction in model size. However, larger scale tying across all triphones of the same class degraded performance. This is a somewhat surprising result since it is known that the monophone variances are well-trained and many of the triphone variances were badly undertrained. When all variances are tied together to give a grand variance system performance drops significantly. This is contrary to the improved results cited for grand variance in word recognition [6,9]. One explanation might be that in applying large scale smoothing, the discrimination of those triphones which are well-trained is being degraded and this negates any improved discrimination achieved by the under-trained triphones. It would then follow that a more data sensitive smoothing technique is needed and this will be investigated further.

5. CONCLUSIONS

In this paper, a method of manipulating sets of HMM's by applying various kinds of parameter tying operations has been presented and its use has been illustrated in the form of an experiment to build a set of generalised triphone models for the TIMIT database. The construction method used synthesised triphones by joining together left and right dependent biphones. This method, which depends on the ability to tie and untie states, has the advantage of avoiding the need to train large numbers of triphones. The method was successful in that it allowed the construction of a set of generalised triphones which gave performance comparable to that reported by other researchers.

The use of tying both states and variances in order to increase model robustness was also investigated. The results here were less clear-cut. Tying the centre state of all triphones from the same phoneme class and tying all variances within the same state reduced the total number of parameters within the system by 60% without affecting the recognition performance. However, increased levels of variance tying resulted in degraded performance. This was in spite of the fact that the set of generalised triphones had a large number of floored variance estimates indicating undertraining. This suggests that the parameter smoothing achieved by simple tying schemes is too crude and hence this needs further investigation.

REFERENCES

[1] Bahl LR, Jelinek F, Mercer RL. *A Maximum Likelihood Approach to Continuous Speech Recognition.* IEEE Trans PAMI, Vol 5, No 2, pp179-190, 1983

[2] Bellegarda JR, Nahamoo D. *Tied Mixture Continous Parameter Modeling for Speech Recognition.* IEEE Trans ASSP Vol 38, No 12, pp2033-2045, 1990

[3] Digalakis V, Ostendorf M, Rohlicek JR. *Fast Search Algorithms for Connect Phone Recognition Using the Stochastic Segment Model.* Proc DARPA Speech and Natural Language Workshop, Hidden Valley, Pennsylvania, pp173-178, June, 1990

[4] Huang XD, Jack MA. *Semi-continuous hidden Markov models for Speech Signals.* Computer Speech and Language, Vol 3, No 3, pp239-252, 1989

[5] Lee K-F, Hon H-W. *Speaker Independent Phone Recognition Using Hidden Markov Models.* IEEE Trans ASSP, Vol 37, No 11, pp1641-1648, 1989

[6] Paul DB. *The Lincoln Robust Continuous Speech Recogniser.* Proc ICASSP, S9.4, pp449-452, Glasgow, Scotland, 1989

[7] Richter AG. *Modeling of Continuous Speech Observations.* Conf on Advances in Speech Processing, IBM Europe Institute, Oberlech, Austria, July, 1986

[8] Robinson AJ, Fallside F. *A Recurrent Error Propagation Network Speech Recognition System.* Computer Speech and Language, Vol 5, No 3, pp259-274, 1991

[9] Russell MJ, Ponting KM. *Experiments with Grand Variance in the ARM Continuous Speech Recognition System.* RSRE Memorandum 4359, RSRE, Malvern, Worc, England, 1990

[10] Woodland PC, Cole DR. *Optimising Hidden Markov Models using Discriminative Output Distributions.* Proc ICASSP, S8.5, pp545-548, Toronto, Canada, 1991

[11] Young SJ *HTK: Hidden Markov Model Toolkit Reference Manual - Version 1.3.* Speech Group, Cambridge University Engineering Dept, Cambridge, England, 1992