EXTENSIONS TO PHONE-STATE DECISION-TREE CLUSTERING: SINGLE TREE AND TAGGED CLUSTERING

Douglas B. Paul

Dragon Systems, Inc.
320 Nevada St.
Newton, MA, 02160, USA

ABSTRACT

The following article describes two extensions to the "traditional" decision tree methods for clustering allophone HMM states in LVCSR systems. The first, *single tree clustering*, combines all allophone states of all phones into a single tree. This can be used to improve performance for very small systems. The single tree clustering structure can also be exploited for speaker and channel adaptation and is shown to provide a 30% reduction in the error rate for an LVCSR task under matched channel conditions and a greater reduction under mismatched channel conditions. The second, *tagged clustering*, is a mechanism for providing additional information to the clustering procedure. The tags are labels for any of a wide variety of factors, such as stress, placed on the triphones. These tags are then accessible to the clustering process. Small improvements in recognition performance were obtained under certain conditions. Both methods can be combined.

INTRODUCTION

Recently, a number of sites have started to use phonological rule based state-wise decision-tree clustered[2] Gaussian mixture pdfs in their hidden Markov model large vocabulary continuous speech recognition (LVCSR) systems[1, 5, 9, 11]. This form of pdf simultaneously improves recognition accuracy, predicts models for unobserved allophones, and allows improved size vs. amount-of-training-data trade-offs compared to some of the earlier forms of pdf. To date, these systems have used separate decision trees for each phone state. Thus, a system with 40 phones and 3 states per phone would have 120 separate trees. These systems have usually used only the identity (and, in some cases, presence of a word boundary) of the context phones in the clustering questions.

The basic paradigm used in this effort for generating the decision trees is as follows:

1. Train single Gaussian per state pdfs for all individual states to be clustered (e.g. all unique triphone states observed in the training data).
2. For each tree (group to be clustered)
   (a) Combine all individual state Gaussians into a single Gaussian for the root node.
   (b) Successively split each leaf node into two leaf nodes, each with a single Gaussian pdf using the best split according to one of a set of questions applied to the context phones. The best split is determined by the largest increase in the log-likelihood of the data. Terminate if the change in the log-likelihood is less than a threshold or the number of observation frames assigned to a child node is less a second threshold.
3. Prune back the trees to the desired number of leaves by iteratively collapsing the smallest split. (The thresholds of (2b) are set to generate larger than desired size trees.) This allows generation of a prespecified number of tree leaves.
4. The leaves of the decision tree now define a state tying or clustering. Use the Gaussian associated with each leaf to seed a Gaussian mixture pdf.

The splitting questions used here are defined by lists of phones typically generated from phonological considerations: for instance, the question "Is the phone a vowel?" would be the list of vowels. The set of questions also includes a set of singleton questions which contain only the individual phone. If an allophone is contained in the question-list, it is placed in the left child, otherwise in the right child. A question may be optionally limited to a particular context position.

These experiments were performed using an LVCSR system derived from one developed at MIT Lincoln Laboratory[8].

SINGLE TREE CLUSTERING

The restriction of one tree per phone state can be relaxed by augmenting the splitting questions in two ways:

1. Apply the splitting questions to the phone itself (in addition to the context phones as above).
2. Add the question "Is the state number=x?" (independently proposed by [4]).

This set of questions now allows the clustering to proceed from a single tree root for all phone states.

In addition, singly-rooted clustering can be performed in two stages:

1. Use a subset of the questions and apply them only to the monophones. Grow tree to exhaustion.
2. Continue growing using all questions and all phone contexts.

The results of stage 1 generate a set of intermediate roots with a decision tree superstructure. Four sets of intermediate roots (depending on the question subset) are plausible:

1. Single root.
2. State roots (3 in the introductory example).
3. Phone roots (40 in the introductory example).

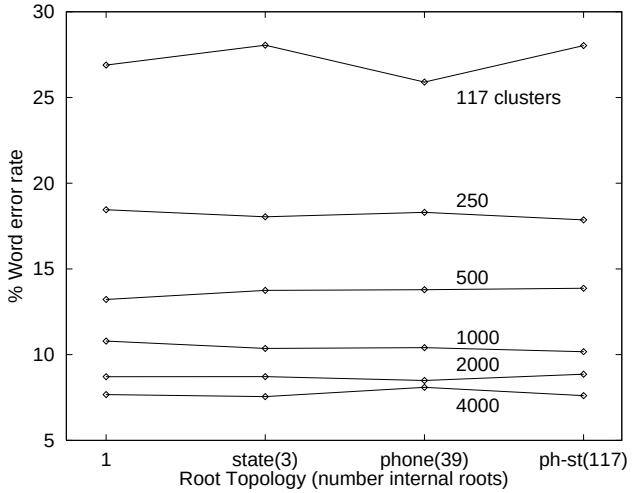

Figure 1. Wd error rate vs. clustering tree topology. The system had 39 phones and 3 states/phone (117=39*3). 5K word DARPA WSJ0 SI task.

4. Phone-state roots (120 in the introductory example, the "traditional" approach plus a superstructure).

Thus this formalism subsumes the traditional one tree per phone-state approach and offers additional variety.

This ability to reduce the number of clusters (pdfs) might be useful in several situations. If the amount of training data is severely restricted, a single tree would have the ability to share pdfs between phones and/or states of the phones. (In fact, a single tree system can have fewer pdfs than phone states.) With large amounts of training data and large numbers of clusters, little difference between the tree topologies would be expected because one would train a very detailed split. Figure 1 shows the results of an experiment which confirms this conjecture and also suggests that the phone root topology also performs well for very small numbers of states. The datapoint at the top right is equivalent to a monophone system.

Tree-based Adaptation

The singly-rooted clustering tree can also be exploited for adaptation. The tree defines a hierarchical set of clusters and a corresponding hierarchy of successive transforms such that the number of adaptation transforms can be increased dynamically as the amount of adaptation data is increased. In the limit of a small amount of adaptation data, there is only one adaptation transform (the root) whose adaptation would produce the equivalent of cepstral mean normalization (commonly used for blind channel equalization). In the opposite limit of a large amount of adaptation data, each mixture Gaussian would be adapted individually. For intermediate amounts of adaptation data, some set of internal tree nodes would define the adaptation transforms. (The mixtures can be treated as just a final N-way split on the tree.)

An algorithm for tree-based adaptation of the Gaussian means is as follows. Each Gaussian is represented by its mean vector μ and estimation count n, and each node is also characterized by a Gaussian. Define the "extended tree" to be the clustering tree and the mixture Gaussians attached to the clustering tree leaves. (Symbol definitions are at the end.)

1. Setup:
   (a) Make a single Gaussian μ_{old} for each extended tree node by a weighted combination of the Gaussians in its immediate children.
   (b) Compute the delta $\Delta_{c,p} = -\Delta_{p,c} = \mu_{old,c} - \mu_{old,p}$ between all non-root Gaussians and their parent's Gaussians for the extended tree.
2. Operation:
   (a) Align the input observations to the clustering tree leaves using Viterbi or forward-backward alignment from a known (supervised) or recognized (unsupervised) transcription and accumulate posterior probability weighted observations (γ_n=accumulators, n_n=count) for each mixture Gaussian and make new estimates of the Gaussian means $\mu_n = \frac{\gamma_n}{n_n}$.
   (b) Propagate new Gaussian means and counts up the extended tree:

$$\mu_{n,p} = \frac{1}{n_{n,p}} \sum_{\{c\}} n_{n,c}(\mu_{n,c} + \Delta_{p,c})$$

$$n_{n,p} = \sum_{\{c\}} n_{n,c}$$

   (c) Adapt the root Gaussian:

$$\mu_{a,r} = \frac{n_{n,r}}{n_{n,r} + n'_{old,r}} \mu_{n,r} + \frac{n'_{old,r}}{n_{n,r} + n'_{old,r}} \mu_{old,r}$$

$$n_{a,r} = n_{n,r} + n'_{old,r}$$

   (d) Adapt the non-root Gaussians (top down):

$$\mu_{a,c} = \frac{n_{n,c}}{n_{n,c} + n'_{a,p}} \mu_{n,c} + \frac{n'_{a,p}}{n_{n,c} + n'_{a,p}} (\mu_{a,p} + \Delta_{c,p})$$

$$n_{a,c} = n_{n,c} + n'_{a,p}$$

   where μ is the Gaussian mean vector, n=the count, x_{old}=the initial value, x_n=value based on new data, x_a=adapted value, x_r=root node, x_c=child node, x_p=parent node, and $n' = discount(n)$.

The simplest discount is just a constant, but better performance is achieved by using

$$discount(n) = \frac{nm}{n+m}$$

[8] where m is the maximum value. (This has the net effect of shrinking the prior count as one moves down the tree and thus more transforms are activated.) Each non-root stage is a local Bayesian adaptation using the parent plus the delta as a prior. This, combined with an appropriate discount on the prior count, produces the desired behavior: if there is only a small amount of adaptation data for a node, then the adapted value for the node follows the changes of the parent and if the node has a large amount of adaptation data, it uses that data and ignores its parent. As more and more adaptation data accumulates, the surface dividing these two conditions moves down the tree and the number of active adaptation transforms increases.

Two experiments were performed to test the above procedure. Both experiments started with an SI DARPA

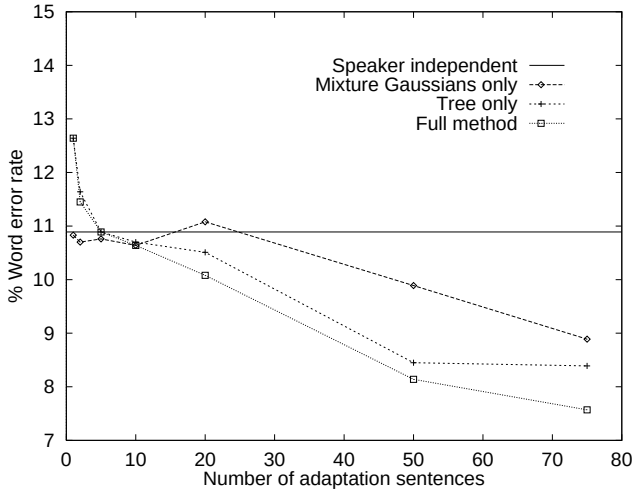

Figure 2. Tree based adaptation (matched train and test)

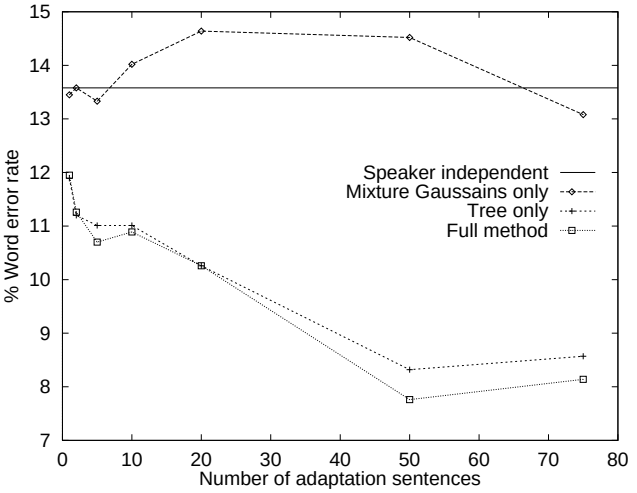

Figure 3. Tree based adaptation (mismatched train and test)

WSJ0 trained set of phone-state-topololgy decision-tree-clustered grand-diagonal-variance Gaussian-mixture models with cepstral mean normalization. There were 4000 clusters and up to 8 Gaussians per mixture. A DARPA baseline 5K word trigram language model produced by Lincoln Lab[7] was used. The test data consists of 100 sentences from each of four speakers (the S4 development test set from the DARPA WSJ1 dataset)–a varying subset of the first 75 sentences was used for recognition-time (incremental) adaptation and the final 25 sentences were used for testing without adaptation. The training and test speakers were all native speakers of American English and all used the same model high-quality boom microphone. All adaptation was supervised, but the algorithm has been shown to perform similarly for unsupervised adaptation.

The first experiment tested matched training and testing environments to examine pure speaker adaptation. Three conditions were tested: (1) adapting only the mixture Gaussians (pure Bayesian adaptation using the trained means μ_{old} as priors), (2) adapting only the tree (the mixture Gaussians were moved as a group so that their weighted average μ equaled that of the corresponding tree leaf), and (3) adapting the tree and mixture Gaussians according to the full scheme described above. Figure 2 shows (1) to be initially slow to adapt, (2) to be faster, and (3) to be fast and to provide the best performance.

The second experiment was the same as the first, except that channel/microphone mismatch was simulated by disabling cepstral mean normalization in the recognizer (Figure 3). Now adapting only the mixture Gaussians (case 1) causes the system to get worse with a very slow recovery because the Gaussians become unbalanced (i.e. some are adapted to the new environment, some still model the old environment.) The tree adaptation (2) now adapts rapidly to the new environment and propagates the gross changes to all of the mixture Gaussians. The full method (3) does better than either alone.

TAGGED CLUSTERING

As described so far, the only way to include such factors as lexical stress, tones, function word dependency, and speaker sex in the clustering is to expand the set of monophones. (For instance, many sites use totally separate models for male and female speakers.) This rapidly becomes self-defeating because the training data is split into smaller and smaller groups until inadequate amounts of data are available to train the individual models. One mechanism for solving this problem is *tagging*.

In tagging, each phone (including the context phones) of the triphone (or any higher order context phone) can be augmented with any number of tags. For instance, the triphone x-y-z would become x-y_1-z if the phone y had a level 1 stress tag. (Tags can also include boundary marks, such as syllable and word boundary.) The decision tree clustering then includes questions about the tags to allow it to trade off all combined data-driven and external knowledge factors without expanding the monophone set. Thus, tagging is a very general method for including additional information in the clustering process. It is controlled simply by choosing the tag sets in augmented versions of a triphone (or larger-phone) dictionary and propagation of the context tags and phones over word boundaries according to the sentence transcriptions.

One experiment with tagging evaluated the use of syllable boundary tags and stress tags. The first test simply added syllable boundary tags to the context phones of the syllable boundary triphones. The second test added stress tags to the vowels in both center and context positions in the triphones. The third contained syllable boundary tags and tagged all phones of the syllable with the vowel stress tag. There were three levels of vowel stress.

System	No. preclustered states	Wd err rate
Triphone (control)	83K	8.41%
Syllable bdry tags	92K	8.47%
Vowel stress tags	136K	7.97%
Syllable bdry tags + syllable stress tags	192K	7.89%

WSJ0 SI trained, 4000 clusters, 3 states/phone, 5K vocab, 10 WSJ0 dev test speakers, std dev $\sim$.33%

These results show no gain for syllable boundary modeling

and a moderate gain for stress modeling. (Experiments on the same data with a different dictionary, however, show little gain for stress modeling.)

A second experiment explored speaker sex-dependent phone modeling by tags. Both systems were identical except that the first used sex-dependent phone sets and the second used a single phone set with sex tags.

System	No. preclustered states	Wd err rate
Sex-dependent triphones	41K	8.43%
Sex-tagged triphones	41K	8.59%

WSJ0 SI trained, 4000 clusters, 3 states/phone, 5K vocab, 10 WSJ0 dev test speakers, std dev ∼.34%

Both methods yielded similar results in this experiment.

DISCUSSION AND CONCLUSIONS

The single tree clustering provides several advantages over the traditional one tree per phone-state clustering. It allows several variations in the structure which may be used to advantage in small systems. One of these structures (the phone-state intermediate root topology) subsumes the traditional method.

The singly-rooted clustering also provides an effective structure for adaptation. The extended tree structure sets up a hierarchical inheritance structure which defines a continuously variable number of adaptation transforms. Thus the adaptation will only use a small number of transforms when a small amount of data is available and as more data accumulates, the number of transforms increases until, in the limit, each mixture Gaussian is adapted individually. The full range is accomplished by a single unified method. (Others have combined a small number of top level transforms with bottom level Bayesian adaptation, but required two dissimilar algorithms[3].)

The adaptation algorithm presented here uses only Bayesian adaptation with appropriately chosen priors and simple delta adaptation transforms to propagate the priors. It is quite likely that this algorithm can be extended to use more complicated transforms[6, 10]. This method also works for simple Gaussian mixtures by treating them as a two-level tree. Then if any member of the mixture is adapted, all are modified.

The experiments performed here used supervised infinite-memory incremental adaptation. Other tests (not presented here) show this algorithm to adapt effectively without supervision. The infinite-memory adaptation is appropriate for research experiments, but is generally a poor model of the real world. The algorithm described here can be trivially modified to incorporate a more practical exponentially decaying memory. The algorithm can also be used for block mode adaptation.

Tagged clustering is a fairly general mechanism for making potentially useful information available to the clustering process. Potential uses for tagging include [function] word dependence, linguistic factors, phonological factors, prosodic factors, tones, speaker groups, and environmental factors. A number of these factors have been used previously–tagging simply provides a single easily used mechanism for incorporating them into the clustering process to allow the data to determine which of these factors should be used.

So far, the experiments using tagging have provided mixed results. However, where recognition comparisons are available, these results appear to be consistent with the same factor modeled by a different mechanism. Hopefully, the simplicity of the mechanism will allow more factors and combinations of factors to be tested.

REFERENCES

[1] L. R. Bahl, P. V. de Souza, P. S. Gopalakrishnan, D. Nahamoo, and M. A. Picheny, "Decision Trees for Phonological Rules in Continuous Speech," Proc. ICASSP 89, Glasgow, Scotland, May 1989.

[2] F. R. Chen and J. Shrager, "Automatic Discovery of Contextual Factors Describing Phonological Variation," Proc. DARPA Speech and Natural Language Workshop, Morgan Kaufmann Publishers, Feb. 1989.

[3] V. V. Digalakis and L. G. Neumeyer, "Speaker Adaptation Using Combined Transformation and Bayesian Methods," IEEE Trans. on Speech and Audio Processing, July 1996.

[4] A. Lazaridès, Y. Normandin, and R. Kuhn, "Improving Decision Trees for Acoustic Modeling," Proc. ICSLP 96, Philadelphia, Oct. 1996.

[5] H. Hon and K. Lee, "Recent Progress in Robust Vocabulary-Independent Speech Recognition," Proc. DARPA Speech and Natural Language Workshop, Morgan Kaufmann Publishers, Feb. 1991.

[6] C. J. Leggeter and P. C. Woodland, "Flexible Speaker Adaptation Using Maximum Likelihood Linear Regression," Proc. ARPA Spoken Language Systems Technology Workshop, Austin, Texas, Morgan Kaufmann Publishers, Jan. 1995.

[7] D. B. Paul and J. K. Baker, "The Design for the Wall Street Journal-based CSR Corpus," Proc. DARPA Speech and Natural Language Workshop, Morgan Kaufmann Publishers, Feb. 1992.

[8] D. B. Paul, "New Developments in the Lincoln Stack-Decoder Based Large-Vocabulary CSR System," ICASSP 95, Detroit, May 1995.

[9] R. Roth, L. Gillick, J. Orloff, F. Scattone, G. Gao, S. Wegmann, and J. Baker, "Dragon Systems' 1994 Large Vocabulary Continuous Speech Recognizer," Proc. ARPA Spoken Language Systems Technology Workshop, Austin, Texas, Morgan Kaufmann Publishers, Jan. 1995.

[10] V. Nagesha and L. Gillick, "Studies in Transformation-Based Adaptation," ICASSP 97, Munich, Apr. 1997.

[11] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-Based State Tying for High Accuracy Acoustic Modeling," Proc. ARPA Workshop on Human Language Technology, Princeton, NJ, Morgan Kaufmann Publishers, March 1994.