HMM STATE CLUSTERING BASED ON EFFICIENT CROSS-VALIDATION

T. Shinozaki

Department of Electrical Engineering,
University of Washington, Washington, U.S.A
staka@ee.washington.edu

ABSTRACT

Decision tree state clustering is explored using a cross validation likelihood criterion. Cross-validation likelihood is more reliable than conventional likelihood and can be efficiently computed using sufficient statistics. It results in a better tying structure and provides a termination criterion that does not rely on empirical thresholds. Large vocabulary recognition experiments on conversational telephone speech show that, for large numbers of tied states, the cross-validation method gives more robust results.

1. INTRODUCTION

Decision tree clustering is an important method for context-dependent HMM modeling, not only for robust parameter estimation but also for predicting probability distributions for unseen contexts [1]. The tree is grown step by step, choosing questions that divide the context using a greedy strategy to maximize some objective function. Several objective functions have been proposed, such as likelihood [1] and cross-entropy [2]. Likelihood-based clustering is the most popular method and known to be effective. The maximum likelihood criterion is also consistent with the overall objective of the standard HMM training algorithm.

A limitation of the likelihood objective, however, is that it is guaranteed to increase as parameters are added, e.g. new nodes in the tree, since the splits are trained and evaluated using the same data. Hence, the tree can potentially grow until all states are untied. Furthermore, the decisions made in tree growing are unreliable when there are a small number of samples associated with a node. The problem is exacerbated when dealing with large candidate question sets, which are of interest for introducing new knowledge sources to characterize acoustic variability, such as syllable structure [3] and acoustic cues such as speaking rate and SNR [4]. To deal with these problems, empirical thresholds are required such as minimum likelihood difference before and after node splitting and minimum occupancy counts of a state [5]. To determine these thresholds, it is necessary to run several recognition experiments, which can be very time consuming for systems using large amounts of data.

These problems of the likelihood based method are due to the lack of a mechanism of balancing the number of parameters and accuracy of parameter estimation. Information-theoretic criteria such as MDL and BIC provide means to select a model with proper model complexity for a given amount of training data and have been applied for tree-based clustering [6, 7]. These criteria have a theoretically-derived penalty term in addition to the likelihood that balances model fit vs. complexity. However, in practice, a generic coefficient for the penalty term often does not work well, and again it requires a empirically tuned weight factor.

In this paper, I investigate a cross-validation-based tree clustering method, similar to that proposed in [8]. Cross-validation is known to be a straightforward and useful method for structure optimization, which can lead to more robust decisions and a simple criterion for determining the optimal tree size. To apply cross-validation for tree clustering, a key issue is how to reduce the computational cost, which can be quite large if the cross-validation likelihood is calculated for each question directly from the observations assigned to a node. The main contribution of this work is the presentation of an efficient approach to cross-validation likelihood computation using sufficient statistics, which makes it possible to use more cross-validation folds than in previous work and train more complex models. While the previous work assumed semi-continuous HMM to save computational cost, the proposed method works for continuous HMM. I also report recognition results on conversational telephone speech comparing the cross-validation technique to the more common likelihood-based approach in small training set scenarios, showing that more robust trees are generated when the complexity is high.

2. DECISION TREE STATE CLUSTERING

Decision tree HMM state clustering is a top-down clustering method to optimize the state tying structure for robust parameter estimation. A leaf of the decision tree corresponds to a set of HMM states to be tied. The tree growing process begins with a root node that may have all HMM states, or all states associated with a particular phone, etc. Then, a question is selected that divides the set of states into two subsets assigned respectively to two child nodes, chosen so that the

corresponding new HMM has the largest likelihood for training set. The tree is grown in a greedy fashion, successively splitting nodes by selecting the question and node that maximize likelihood gain at each step.

In the conventional approach to tree-based clustering, it is assumed that the state alignment does not change with different tying configurations, to reduce the cost of computing likelihoods. In this case, the likelihood change due to expanding the parameter set (splitting a node) is simply given by the change in observation likelihoods of the impacted states. The model parameters and associated observation likelihoods can be computed efficiently by using the pre-computed sufficient statistics associated with each state in the model (which may be based on either Viterbi or EM-statistics) .

In the proposed clustering method, likelihood is estimated using the cross-validation method. For N-fold cross-validation, the training data is randomly divided into N different groups. Then, a model is trained using $N-1$ groups of data, and likelihood is computed for the group excluded in the training. This process is repeated for N times with different combinations of $N-1$ groups. The likelihood is accumulated and used for the question selection. In the next section, I show that this likelihood can also be computed efficiently using sufficient statistics. To distinguish conventional and cross-validation based likelihood, they are hereafter denoted as "self-test likelihood" and "CV likelihood", respectively.

2.1. Algorithm for obtaining CV likelihood

Let D be a training set and D_f be a partition for N-fold cross-validation. That is,

$$D = \bigcup_{f=1}^{N} D_f, \qquad D_i \bigcap D_j = \phi \quad (i \neq j). \tag{1}$$

For the f-th evaluation, $\bar{D_f} = \bigcup_{f' \neq f} D_{f'}$ is used to estimate HMM parameters and D_f is used to evaluate likelihood. Let S be a set of states, s be a state, $\gamma_s(t)$ be occupancy probability of state s at time t, and $\boldsymbol{x_t} = (x_1(t), x_2(t), \cdots, x_d(t))^T$ be d-dimensional feature vector at time t. Let $A_f^0(s)$, $\boldsymbol{A}_f^1(s)$ and $\boldsymbol{A}_f^2(s)$ be the sufficient statistics of the observations aligning to state s. For diagonal Gaussian distributions, these are:

$$A_f^0(s) = \sum_{t \in D_f} \gamma_s(t), \tag{2}$$

$$\boldsymbol{A}_f^1(s) = \sum_{t \in D_f} \boldsymbol{x_t} \gamma_s(t), \tag{3}$$

$$\boldsymbol{A}_f^2(s) = \sum_{t \in D_f} \boldsymbol{x_t^2} \gamma_s(t), \tag{4}$$

where $\boldsymbol{x}^2 = \left(x_1^2, x_2^2, \cdots, x_d^2\right)^T$.

Using these statistics, the f-th ML estimates of mean vector $\boldsymbol{\mu}$ and variance vector $\boldsymbol{v}$ of state s from $\bar{D_f}$ are:

$$\boldsymbol{\mu}_f(s) = \frac{\sum_{f' \neq f} \boldsymbol{A}_{f'}^1(s)}{\sum_{f' \neq f} A_{f'}^0(s)}, \tag{5}$$

$$\boldsymbol{v}_f(s) = \frac{\sum_{f' \neq f} \boldsymbol{A}_{f'}^2(s)}{\sum_{f' \neq f} A_{f'}^0(s)} - \boldsymbol{\mu}_f(s)^2. \tag{6}$$

Let L_f be the log likelihood for f-th data fold using Gaussians that have means $\boldsymbol{\mu}_f(s)$ and variances $\boldsymbol{v}_f(s)$, as shown in equation (7), where $\boldsymbol{\Sigma}$ is a diagonal covariance matrix whose main diagonal is $\boldsymbol{v}$. By putting the summation over t inside and utilizing the assumption that $\boldsymbol{\Sigma}$ is a diagonal matrix, equation (7) can be efficiently evaluated using the pre-computed statistics A^0, $\boldsymbol{A}^1$, and $\boldsymbol{A}^2$ as shown in equation (8), where $\boldsymbol{v}^{-1} = \left(v_1^{-1}, v_2^{-1}, \cdots, v_d^{-1}\right)$. Finally, the CV likelihood L is obtained by summing the likelihoods for each fold:

$$L = \sum_{f=1}^{N} L_f. \tag{9}$$

2.2. Splitting gain and termination criterion

Likelihood gain ΔL is defined as the difference of the likelihood before and after splitting:

$$\Delta L(Q) = L' - L, \tag{10}$$

where L and L' are the likelihoods before and after a splitting by a question Q. During decision tree clustering, questions are selected so as to maximize ΔL. Since the likelihood is based on cross-validation, ΔL can take negative values unlike with the conventional self-test likelihood. A negative gain indicates the splitting yields HMM with too many parameters for the training set and the parameters can not be estimated properly. Therefore, when using CV likelihood, a good termination criterion for splitting is

$$\max_Q \Delta L(Q) < 0. \tag{11}$$

This criterion does not require any empirical thresholds. We refer to it as the zero-gain criterion.

2.3. Computational and storage costs

The statistics $A_f^0(s)$, $\boldsymbol{A}_f^1(s)$ and $\boldsymbol{A}_f^2(s)$ are computed prior to clustering for each HMM state, requiring N times the storage for N folds but essentially no additional computation compared to the self-test likelihood approach. During each question evaluation, the computation increases by a factor of N due to the need to compute different sets of model parameters and L_f for each of the N folds. Since the computational cost of clustering is small relative to other aspects of HMM training, an increase by even a factor of $N = 10$ is quite reasonable.

$$L_f = \sum_{t \in D_f} \sum_{s \in S} \log \left\{ \frac{1}{\sqrt{(2\pi)^d \left|\boldsymbol{\Sigma}_f(s)\right|}} \exp\left(-\frac{1}{2} \left(\boldsymbol{x}_t - \boldsymbol{\mu}_f(s)\right)^T \boldsymbol{\Sigma}_f(s)^{-1} \left(\boldsymbol{x}_t - \boldsymbol{\mu}_f(s)\right) \right) \right\} \gamma_s(t) \tag{7}$$

$$= -\frac{1}{2} \sum_{s \in S} \left\{ \log\left((2\pi)^d \left|\boldsymbol{\Sigma}_f(s)\right| \right) A_f^0(s) + \left(\boldsymbol{v}_f(s)^{-1}\right)^T \boldsymbol{A}_f^2 - 2\left(\boldsymbol{\Sigma}_f(s)^{-1} \boldsymbol{\mu}_f(s)\right)^T \boldsymbol{A}_f^1 + \left(\boldsymbol{v}_f(s)^{-1}\right)^T \boldsymbol{\mu}_f(s)^2 A_f^0 \right\} \tag{8}$$

3. EXPERIMENTS

Large vocabulary recognition experiments were conducted on conversational telephone speech to compare the usefulness of CV vs. self-test likelihood with different amounts of training data. In these initial experiments, small training sets are used (rather than increased question sets) in order to better explore the effect of the different criteria on sparse contexts.

3.1. Paradigm

The Decipher [9] system was used for model training and testing. The dictionary is based on 38k-word vocabulary and has 83k entries including multi-words and multiple pronunciations. Triphone HMMs are used with a three-state left-to-right topology and 47 phone set. Decoding involves rescoring a lattice of initial pass hypotheses with a speaker-adapted model (MLLR) and a 4-gram language model. Note that this system is different from the standard SRI recognition system in that it has only PLP cross-word triphone models, uses single mixture distributions (except where noted), uses only ML training (vs. MMIE and MPE), and the training set is much smaller than in the full system. The training sets were randomly sampled from the Fisher corpus, using subsets of sizes 16, 32 and 64 hours. The test set was the RT04 DevTest set.

For the tree based clustering, a total of 567 questions are used. These questions are about left and right phone context, phone category such as nasal and front vowel. Syllable and word attribute questions are also included that ask whether the phone is at beginning, middle, or the end of the units. The identity of the center phone and the state position are also part of the question set. To make self-test likelihood based method work properly for large trees, a variance floor of $1.0E-6$ was used, and the same setting was applied for the CV likelihood method. The sufficient statistics were calculated from Viterbi alignment. The cross validation uses 10 folds.

3.2. Results

The first set of experiments, based on 16 hours of training data, looked at likelihood and word error rate (WER) behavior for trees designed using the two methods for a wide range of complexities. Figures 1 and 2 show self-test and cross-validation likelihood gain and likelihood, respectively, obtained for different numbers of clusters during clustering. As shown in the figures, CV-likelihood gains decrease much more rapidly than the optimistic self-test likelihood gains and (unlike the self-test gains) eventually become negative at the point of overtraining. The optimal number of clusters indicated by the zero-gain criterion was 12k in this case.

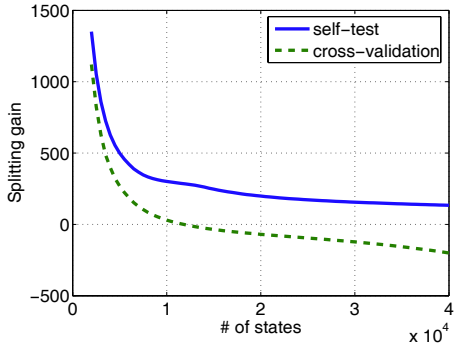

Fig. 1. Number of states vs. splitting gain.

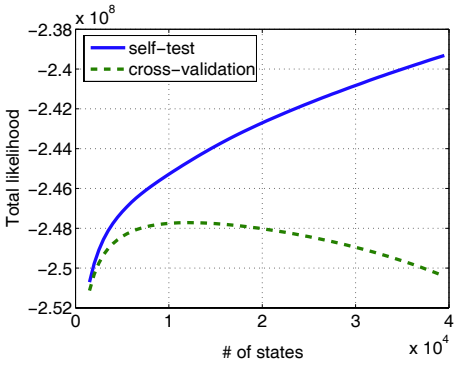

Fig. 2. Number of states vs. total likelihood.

Figure 3 shows the relation between the number of parameters and the WER of Gaussian HMMs using the 16 hour training set. The WERs of the self-test and cross-validation based clustering are similar when the number of states is relatively small, because many samples are assigned to a state and the probability distribution is estimated properly regardless of the method. Not surprisingly, the first 42 questions selected by the two methods were identical. When the number of states

is large, the CV likelihood criterion gives lower WER, because the number of samples per node becomes small and there is more potential for the optimistic self-test criterion to lead to a poor choice of questions. The lowest WER is obtained with 6K states and 18K states for the self-test and CV likelihood criteria, 42.0% and 40.8% respectively, indicating that the CV criterion is more robust for complex models. In addition, we observe that both the likelihood and WER are relatively stable for the CV criterion over a wide range of state sizes. In fact, the 12k-state stopping point predicted by the zero-gain criterion is conservative in this case, and slightly better performance is obtained with a larger tree (40.8% vs. 41.5% WER).

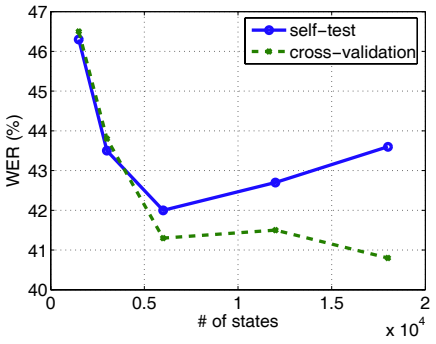

Fig. 3. Number of states vs. word error rate.

For 32 hours of data, the optimal stopping point based on the zero-gain criterion was 18k. For this number of clusters, the WER of self-test and cross-validation based method were 40.9% and 39.6%, respectively, and the absolute WER reduction was 1.3%. When number of the mixtures were increased to four, their WER were 32.9% and 32.1%, respectively, and the absolute WER reduction was 0.8%. For 64 hours of data, the optimal number of states would be much greater than this, yet the standard heuristics associated with the self-test method result in 1.5k clusters.

4. DISCUSSION

The results above show that the CV likelihood criterion leads to more robust decision tree state tying and a more principled stopping criterion, with bigger potentially gains for more complex systems indicated by the differences in the 16 vs. 32 hour systems. The likelihood is evaluated using an efficient algorithm based on sufficient statistics, so the increase in computation is a factor of N (for N-fold CV), which is reasonable for $N = 10$ given the low relative cost of tree design.

Unfortunately, because the algorithm is more robust, a strict application of the zero-gain criterion with diagonal Gaussian distributions leads to a very large model space. In the conventional approach, a much smaller model space would be designed, and then complexity introduced through using Gaussian mixture distributions. While the CV likelihood criterion could be used in this framework to obtain more robust question choices, the stopping criterion is no longer valid because it is based on single Gaussians in a context where mixtures will be used. Furthermore, if the tree size is small relative to the optimum, then the differences between the self-test and CV likelihood trees are minimal, and there will be little difference in the systems after the mixtures are introduced, as evidence in our experiments with 1.5k clusters and 128 mixture components trained on 64 hours of data. These problems can be solved by extending the CV likelihood criterion to a search space covering state clustering plus mixture splitting. Such an extension will be important for gains to be realized on larger training sets.

Acknowledgments

This work was supported by DARPA under contract MDA972-02-C-0038. Distribution is unlimited.

5. REFERENCES

[1] S. Young, J. Odell, and P. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proc. ARPA Workshop on Human Language Technology*, 1994, pp. 307–312.

[2] M. Y. Hwang, X. Huang, and F. Alleva, "Predicting unseen triphones with senones," in *Proc. ICASSP*, Minneapolis, 1993, vol. II, pp. 311–314.

[3] I. Shafran and M. Ostendorf, "Acoustic model clustering based on syllable structure," *Computer Speech and Language*, vol. 17, no. 4, pp. 311–328, 2003.

[4] C. Füegen and I. Rogina, "Integrating dynamic speech modalities into context decision trees," in *Proc. ICASSP*, Istanbul, 2000, vol. III, pp. 1277–1280.

[5] S. Young *et al.*, *The HTK Book*, Cambridge University Engineering Department, 2005.

[6] K. Shinoda and T. Watanabe, "Acoustic modeling based on the MDL criterion for speech recognition," in *Proc. EuroSpeech*, Rhodes, 1997, vol. 1, pp. 99–102.

[7] W. Chou and W. Reichl, "Decision tree state tying based on penalized Bayesian information criterion," in *Proc. ICASSP*, Phoenix, 1999, vol. 1, pp. 345–348.

[8] I. Rogina, "Automatic architecture design by likelihood-based context clustering with crossvalidation," in *Proc. Eurospeech*, Rhodes, Greece, 1997, pp. 1223–1226.

[9] A. Stolcke *et al.*, "The SRI March 2000 Hub-5 conversational speech transcription system," in *Proc. NIST Speech Transcription Workshop*, 2002.