NOVEL LOOKAHEAD DECISION TREE STATE TYING FOR ACOUSTIC MODELING

Jian Xue and Yunxin Zhao

Department of Computer Science
University of Missouri, Columbia, MO 65211 USA
jxwr7@mizzou.edu, zhaoy@missouri.edu

ABSTRACT

This paper presents two new lookahead methods of constructing phonetic decision trees (PDTs) for acoustic model state tying, a constrained method and a stochastic method. The constrained lookahead method searches for optimal phonetic questions among pre-selected question sets, and reduces contributions of deeper decedents as a function of their levels in the tree. The stochastic full lookahead method uses subtree size instead of likelihood gain as a judgment in selecting a phonetic question for a node split, in order to find a compact tree that is consistent with training data. Since the computational cost of exhaustive lookahead is prohibitively high, a stochastic subtree generation method is used to explore most promising question at each node. We also propose using a phone-state dependent threshold instead of a fixed threshold of likelihood gain to decide if a node split should continue or not. Furthermore, we use a fast Confusion Network (CN) algorithm to combine recognition hypotheses produced by using acoustic models from different PDT training methods. Experimental results show that the proposed lookahead methods consistently decrease model size, and the integration of recognition hypotheses consistently improves recognition accuracy.

Index Terms— phonetic decision trees, constrained lookahead, stochastic full lookahead, phone-state dependent threshold

1. INTRODUCTION

Phonetic decision tree (PDT) state tying is commonly used in acoustic modeling for large vocabulary continuous speech recognition since it can model triphone units or contexts which do not occur in training data [1]. Usually, a PDT is built by using a top-down greedy search procedure. Starting from the root node of the tree, each node is split according to the phonetic question which results in the largest increase of likelihood score in the training data under this node. The node splits continue until the likelihood gain falls below a threshold. A threshold of data count is also applied to ensure that all leaf nodes have sufficient training data. Leaf nodes with different parents are merged if the likelihood loss due to the merging is less than a predefined threshold, which usually equals the one for likelihood gain. Since each node split is based on local likelihood gain, the constructed tree is only locally optimal in general. *K*-step lookahead search is a technique for overcoming the limitation of greedy search. When applied to decision tree induction, lookahead method attempts to predict the profitability of a split at a node by estimating its effect on deeper decedents of the node [2]. Since *K*-step lookahead algorithm has an exponential complexity, usually only one-step lookahead is used.

In this paper, we present two novel lookahead methods of constructing PDTs for acoustic model state tying. The first one is called constrained lookahead. In this method, instead of searching for an optimal question within the whole question set, we first find *n* questions which give the *n*-best local increases in likelihood, and then we constrain the lookahead search for the optimal question to be among the *n* questions to split the node, where the contribution of the deeper decedents is reduced as a function of their levels in the tree [2]. The second method is called stochastic full lookahead, originally proposed in [3]. In this method, instead of using likelihood gain as a judgment to select a phonetic question for node split, subtree sizes are used to find a compact tree that is consistent with the training data set. Since finding an optimal subtree at each node is a NP-Complete problem, a stochastic method is used to generate an ensemble of subtrees for each current node split, and the question that minimizes the subtree size is preferred [3]. In both methods we propose to use phone-state dependent threshold of likelihood gain to decide if a node split should go further or not. We also propose to use a fast Confusion Network (CN) algorithm [4] to combine recognition hypotheses produced by acoustic models resulting from different PDT training methods.

The rest of the paper is organized as the following. Section 2 defines the phone-state dependent threshold. Section 3 presents the constrained lookahead algorithm. Section 4 describes the stochastic full lookahead algorithm. Section 5 shows the CN method for combining recognition hypothese. In section 6, experimental results on a telehealth captioning system are presented. We conclude our work in section 7.

This work is supported in part by National Institutes of Health under the grant NIH 1 R01 DC04340.

2. PHONE-STATE DEPENDENT THRESHOLD

In commonly used method of constructing PDTs, for example HTK [5], a node is split if the largest increase of likelihood score due to the split exceeds a predefined threshold, where the threshold is kept identical for all phone states. Here we propose to use a threshold for likelihood gain that is proportional to the occurrence count of data in each phone state. For a PDT of phone *i* and state *j*, let *Lp* be the log likelihood score of a parent node to be split, *Ly* and *Ln* be the log likelihood scores of the two children nodes, and N_{ij} be the total occurrence count under the root node. We select the question that maximizes the likelihood gain scaled by N_{ij} as

$$\Delta L = \frac{L_y + L_n - L_p}{N_{ij}}$$

If the largest ΔL is small than a predefined threshold *C*, then the split will stop, that is

$$\max(\frac{L_y + L_n - L_p}{N_{ij}}) < C \Rightarrow \max(L_y + L_n - L_p) < N_{ij} C = \lambda_{ij}$$

We call λ_{ij} the phone-state dependent threshold. Compared to the fixed threshold in commonly used method, the threshold is proportional to the total occurrence count of each phone state, which helps prevent phone states with large amounts of training data to grow overly large trees. The minimum occurrence count threshold of leaf nodes is still kept as a constant to ensure sufficient training data for estimation of observation probability distributions.

3. CONSTRAINED LOOKAHEAD

It is well known that the greedy search method of choosing a phonetic question based on largest likelihood gain in the current node split only leads to a local optimization of the decision tree. It seems plausible that a global optimization on decision tree training would lead to improve acoustic models. Since the computational cost of global optimization is too high, lookahead methods can be used to select the phonetic question that produces largest likelihood gain on its deeper decedents. However, traditional lookahead methods did not yield improved performance in speech recognition [2], [6]. Instead it gave worse results than conventional PDT training, mainly due to overtraining.

Here we present a constrained lookahead method to optimize the PDTs. At each node, instead of searching for the optimal question within the whole question set, we first find *n* questions which give the *n*-best local increases in likelihood, and then we find the optimal question among the *n* questions to split the node through a *K*-step lookahead. The n-best constraint is applied to node splits in each lookahead level as well. The rational of the *n*-best approach is that the optimal question is likely one of the *n*-best questions at the current node, since these questions in general have larger impacts on data partition into leaf nodes than questions at decedent nodes. Narrowing the search space may also decrease the effect of outliers. Fig. 1 illustrates the construction of a PDT that employs *K*-step lookahead.

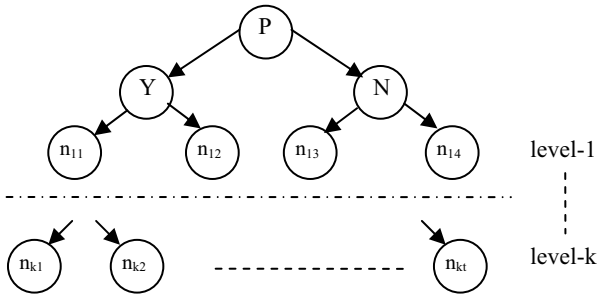

Fig. 1 PDT construction using K-step lookahead.

It was previously reported that the contribution of likelihood gains by the deeper decedents should be reduced with the decedent level [2]. Here we adopt the idea and define the likelihood gain as a weighted average of likelihood gains at successive levels in the K-step lookahead window. Let ΔL_i be the likelihood gain at the *i*th level, and ΔL_p equals *Ly* + *Ln* – *Lp*. At each node, the likelihood gain is defined as

$$\Delta L = \Delta L_p + \sum_{i=1}^{k} \frac{\Delta L_i}{i + tree\_level(P)},$$

where *tree_level*(root)=0, and we select the question that produces the largest ΔL.

4. STOCHASTIC FULL LOOKAHEAD

Tree size has been an important issue in constructing PDTs. One obvious concern is that speech decoding speed is dependent on the number of physical states in acoustic models, since as the number of physical states increases, it takes more time for a decoding engine to compute the likelihood scores. The number of physical states is directly decided by the sizes of the PDTs. Another important concern is that a good model should represent the salient clustering structure of data, instead of over fitting the data, and as such compact model is also preferred.

The goal of the stochastic full lookahead method is to find small trees consistent with the training data. In contrast, traditional tree-pruning based methods first grow a large tree by greedy search and then prune off noncontributing leaves or subtrees to reduce the size of the tree, which in general yields small trees that are not consistent with data.

Exhaustive lookahead at each node will definitely lead to the smallest tree, but the computational cost is prohibitively high, and therefore a stochastic approach is used. For each question under consideration at the current node, the stochastic method generates two ensemble of subtrees, one for its left child node and one for its right child node, respectively. Each ensemble of subtrees provides an estimate of the subtree size. In growing a random subtree, when a phonetic question is used to split a

node, the split question is drawn randomly with a probability proportional to the likelihood gain due to splitting the node by this question. Table 1 describes the question selection procedure used in the stochastic lookahead method, and Table 2 describes the question selection procedure in growing the random subtrees at each node. We modify the algorithm in [3] such that at each node we limit the search space to that defined by the *n*-best questions instead of the whole question set, which decreases the training time greatly. In addition, the n-best approach may yield a good balance between the aspect of model-data fit and that of model compactness.

In determining subtree size, the smallest size of random subtrees in an ensemble is taken, since a random subtree by its nonexhaustive search nature can only overestimate the minimum size of subtree [3].

Table 1 Question selection procedure in stochastic full lookahead.

Get a question set QS including *n* questions which give the *n*-best local increase in likelihood
For each question $q_i \in QS$
Split the node into yes-children and no-children
$\min_{yes} \leftarrow \min_{no} \leftarrow \infty$
Repeat *r* times
Grow a random sub-tree, sub_{yes} of yes-children
$\min_{yes} \leftarrow \min(\min_{yes}, sizeof(sub_{yes}))$
Grow a random sub-tree, sub_{no} of no-children
$\min_{no} \leftarrow \min(\min_{no}, sizeof(sub_{no}))$
$size_i \leftarrow \min_{yes} + \min_{no}$
Return q_i whose $size_i$ is minimal

Table 2 Question selection procedure in growing random subtrees.

Get a question set QS including *n* questions which give the *n*-best local increase in likelihood
$q^* \leftarrow$ Choose question at random from QS; for each question *q*, the probability of selecting it is proportional to the local likelihood increase for the training data set
Return q^*

5. INTEGRATION OF RECOGNITION RESULTS

Upon obtaining recognition hypotheses by using different acoustic models from different PDT training methods, we put the results together into a simple lattice and then align the lattice into a CN by using the fast CN algorithm. The final hypothesis is obtained by picking words with the highest posterior probabilities at each position in the CN. For detail of the fast CN algorithm, please refer to [4]. Fig. 2 is a simple lattice constructed from five recognition hypotheses, and Fig. 3 is the corresponding CN.

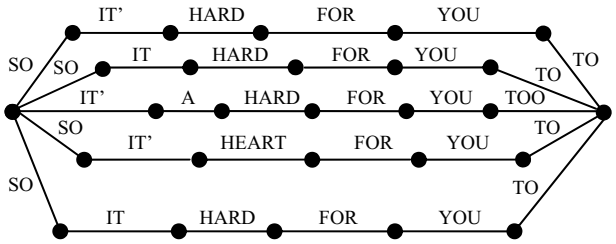

Fig 2. A simple lattice of recognition results

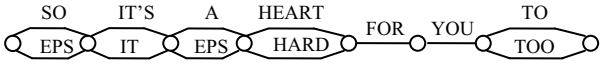

Fig. 3 Confusion network for the lattice in Fig. 2

6. EXPERIMENTAL RESULTS

6.1 Experimental Setup

The proposed phonetic decision tree methods were evaluated on the Telemedicine captioning system developed at the University of Missouri-Columbia. For a detailed description of this project, please refer to [7]. Speaker dependent acoustic models were trained for 5 speakers Dr. 1-Dr. 5. A summary of the data set is provided in Table 3. The training and test datasets were extracted speech data from healthcare speaker's conversation with clients in mock telemedicine interviews. Along with speech durations, word counts from transcription texts are also given in Table 3. Speech features consisted of 39 components including 13 MFCCs and their first and second order time derivations. Feature analysis was made at a 10 ms frame rate with 20 ms window size. Gaussian mixture density based hidden Markov model (GMM-HMM)were used for within-word triphone modeling, where each GMM contained 16 Gaussian components. The task vocabulary is of the size 46,489, with 3.07% of vocabulary words being medical terms.

Table 3. Datasets used: speech (min.)/text (no. of words)

	Training set	Test set
Dr. 1	210/35,348	29.8/5085
Dr. 2	200/39,398	14.3/2759
Dr. 3	145/28,700	19.3/3248
Dr. 4	180/39,148	27.8/6421
Dr. 5	250/44,967	12.1/3988
Total	985/187,561	103.3/21541

6.2 Experimental Results

Table 4 gives the recognition accuracies and table 5 gives the model sizes of 5 different PDT training methods:

- Baseline: baseline method
- B-PSDT: baseline plus phone-state dependent threshold
- CLA-1.: *1*-step constrained lookahead method
- CLA-2: *2*-step constrained lookahead method
- SFLA : stochastic full lookahead method.

We set r =20 for stochastic full lookahead method, and n=20 for both constrained and stochastic lookahead methods. The value C in phone-state dependent threshold is set such that the average threshold of different PDTs equals the one used in baseline method, which is 400 in our experiments.

Table 4 Recognition accuracies of different methods (%)

	Dr. 1	Dr. 2	Dr. 3	Dr. 4	Dr. 5	Avg.
Baseline	81.40	74.16	76.29	78.31	82.40	78.96
B-PSDT	81.39	73.65	76.29	78.32	82.97	79.00
CLA-1	81.44	72.45	76.76	78.31	81.80	78.71
CLA-2	81.46	72.42	76.95	78.20	81.95	78.74
SFLA	81.14	73.90	76.35	78.45	82.92	79.01

Table 5. Model sizes of different methods (number of tied states or physical models)

	Dr. 1	Dr. 2	Dr. 3	Dr. 4	Dr. 5	Avg.
Baseline	2070	1480	1093	1415	1735	1559
B-PSDT	1603	1425	908	1117	1440	1299
CLA-1	1611	1484	918	1131	1476	1324
CLA-2	1625	1509	931	1134	1501	1340
SFLA	1431	1274	879	973	1237	1159

From Table 4 and Table 5 we observe that stochastic full lookahead method reduces model size significantly (26% relative). Constrained lookahead methods do not produce consistent improvements in recognition accuracy.

Table 6 summarizes recognition word accuracy after combining hypotheses from different models, where n-best is based on ranking the five methods by their accuracy performance. From Table 6 we observe that the CN-based hypotheses combination produces a consistent improvement in recognition accuracy. Integrating more hypotheses in general yielded higher accuracy.

Table 6 Recognition accuracy after combination

	Accuracy after combining n-best results			
	n=2	n=3	n=4	n=5
Dr. 1	81.67	81.65	81.97	81.81
Dr. 2	74.52	74.52	74.48	74.45
Dr. 3	76.91	76.88	76.94	77.03
Dr. 4	78.87	78.82	78.74	78.76
Dr. 5	83.20	83.45	83.50	83.55
Avg.	79.48	79.50	79.56	79.56

To investigate the effect of the repeat number r in generating random subtrees on performance of stochastic full lookahead, we trained PDTs with different values of r on Dr. 3 dataset and compared the model sizes and recognition accuracies. From Table 7 we do not see a consistent tendency. Since this is not a deterministic algorithm, different runs of the algorithm might return different results. The best way is to run it a lot of times and get the average result. Due to the time consuming nature of such a procedure, this is left for a future work.

Table 7 Performance of stochastic full lookahead with different r

r	5	10	20	50
Model size	877	861	879	855
Accuracy (%)	76.60	76.63	76.35	76.57

We also compared the performance of constrained lookahead and traditional lookahead (TLA) methods on Dr. 3 dataset. Table 8 gives the results. We can see that the constrained methods give better performance than the unconstrained one both in recognition accuracy and in model size.

Table 8 Comparison of constrained lookahead and traditional lookahead method

	Accuracy (%)	Model size
TLA-1	76.51	1064
TLA-2	76.08	1082
CLA-1	76.76	918
CLA-2	76.95	931

7. CONCLUSION

In this paper we present two new lookahead methods of constructing PDTs: a constrained lookahead method and a stochastic full lookahead method. The stochastic full lookahead method significantly decreases model size without sacrificing average recognition accuracy. Combining the recognition hypotheses from acoustic models of different PDTs through CN yields consistent recognition accuracy improvement. Since these methods use different judgments to train PDTs, the resulting trees have complementary properties to some extent. The phone-state dependent likelihood gain threshold is effective in producing small PDTs without hurting accuracy, and the strategy of using n-best questions in PDT lookahead search reduces both model size and search time.

8. REFERENCES

[1] S. J. Young, J. J. Odell and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modeling," in Proc. ARPA Human Lang. Tech. Workshop, pp. 307-312, 1994.

[2] C. Chesta, P. Laface and F. Ravera, "Bottom-Up and Top-Down State Clustering for Robust Acoustic Modeling," Proc. EUROSPEECH, pp. 11-14, 1997.

[3] S. Esmeir and S. Markovitch, "Lookahead-based Algorithms for Anytime Induction of Decision Trees," Proc. ICML, Vol 69, pp. 257-264, 2004.

[4] J. Xue and Y. Zhao, "Improved Confusion Network Algorithm and Shortest Path Search from Word Lattice," Proc. ICASSP, pp 853-856, 2005.

[5] HTK Toolkit, http://htk.eng.cam.ac. uk

[6] A. Lazaridès, Y. Normandin and R. Kuhn, "Improving Decision Tree for Acoustic Modeling," Proc. ICSLP, pp. 1053-1056, 1996.

[7] Y. Zhao et al, "An Automatic Captioning System for Telemedicine," Proc. ICASSP, pp. I-957 – I-960, 2006.