Lattice Based Discriminative Model Combination Using Automatically Induced Phonetic Contexts

Hao Huang, Bing Hu Li

Laboratory of Multi-lingual Information Technology
Department of Information Science and Engineering, Xinjiang University, Urumqi, P. R. China
{hwanhao,binghulee}@gmail.com

1. Introduction

Discriminative model combination [1] is a popular approach to integrating several knowledge sources in speech recognition. In this approach, multiple model scores are scaled using weighting factors which are trained according to a discriminative training criterion. In our previous work [2], model weights of acoustic scores and tone scores are trained using the extended Baum-Welch (EBW) algorithm under the minimum phone error (MPE) [3] objective function. Other recent works in [4,5] also presented various model combination methods using discriminative model scaling. All have shown improvements on various scale speech recognition tasks.

Moreover, all these works have shown that context-dependent (CD) weighting factors are crucial for highly accurate speech recognition. However, introducing contexts to combine model scores will result in a large number of the weight parameters. The large weight parameter set is often likely to introduce training instability which might degrade the system performance. Moreover, manual context selection is difficult, especially when there are many phonetic/semantic contextual options.

To solve this problem, we propose automatic context induction for discriminative model combination in lattice rescoring. The contexts are modeled by phonetic decision trees. Tree nodes are split using the question in accordance with the maximization of MPE, i.e., the expected accuracy on the training lattices, and the leaf node parameters are updated simultaneously during the tree growing process. Question selection in tree generation is computational expensive because lots of lattice forward-backward calculations are needed. To avoid this, we propose fast question selection, which uses first order approximation to evaluate MPE objective increase. The proposed method is evaluated on a Mandarin speech recognition task by combining the HMM based acoustic model, Gaussian mixture model (GMM) based tone model and multi-layer perceptron (MLP) based phoneme classifier in lattices. Results show the method is able to extract many fewer, but crucial contexts and achieve better accuracy compared with heuristically selected contexts. Results have also shown tree based model combination is superior to the system based on feature space combination.

The remainder of this paper is organized as follows: In section 2, the CD model weight training is briefly reviewed. Section 3 discusses the tree generation. Section 4 presents the experimental results. Section 5 draws the conclusion.

2. Context dependent model combination and discriminative model weight training

2.1. Model combination in lattice

In lattice based model combination, the total score of an arc is computed based on scores of the parallel models:
\[
\psi(a) = \sum_{i} \lambda_i \psi_i(a)
\]
where \( \psi(a) \) is the total score of arc \( a \), \( \psi_i(a) \) is the score from the \( i \)-th parallel model, \( \lambda_i \) is the \( i \)-th model probability.

2.2. Context dependent model weighting

CD model weighting is to scale the model scores using different model weights according to the lattice contexts. Fig. 1 shows the structure of a tonal syllable lattice. The context of each non-silent arc can be expressed as: \( [c|dila-b+e] \), where \( (a) \) current initial; \( (b) \) current tonal final; \( (c) \) left final; \( (d) \) left tone type; \( (e) \) right initial. For example, we assign a weighting component to a tonal syllable \( ["d-ou1"] \) or assign a weighting component to a \( ["d-ou1+sh"] \) context. With more contexts introduced, the number of training parameters will increase drastically. During weight training, only limited amount of training samples might be available. The data sparsity issue must be addressed.

2.3. Discriminative model weight training

The CD weighting parameters are trained according to MPE criterion. Given a training set of acoustic observations \( O_r, r = 1, \ldots, R \), MPE objective is written as [3]:
\[
F_{\text{MPE}} = \sum_{r} \sum_{s} P^\kappa(s|O_r) \cdot A(s, s_r)
\]
where \( P^\kappa(s|O_r) \) is the scaled posterior probability of hypothesis \( s \), \( \kappa \) is a scaling factor. \( Acc(s, s_r) \) is the raw phone accuracy
for hypothesis $s$. When model weights are to be trained, MPE maximization is accomplished iteratively using [2]:

$$
\lambda_{m,i} = \frac{\kappa \gamma^\lambda_{\text{MPE}} \lambda_{m,i} \psi_i(a) |_t + C \lambda_{m,i}}{\sum_i (\kappa \gamma^\lambda_{\text{MPE}} \lambda_{m,i} \psi_i(a) |_t + C \lambda_{m,i})},
$$

(3)

where $\lambda_{m,i}$ and $\lambda_{m,i}'$ are respectively current and newly estimated weights for the $i$th model in weight component $m$. $\gamma^\lambda_{\text{MPE}} = \gamma_a (c(a) - \text{error})$. $\gamma_a$ is the posterior probability of passing arc $q$. $c(a)$ is the average phone accuracy for all the sentence hypothesis that contains arc $a$ and $\text{error}$ is the average accuracy of all the hypothesis. More details about these statistics can be found in [3]. $C$ is a constant used to ensure positive probability weight. More details of derivation and implementation of weight training can be found in [2].

3. Model combination using automatically induced contexts

3.1. Decision tree based context modeling and tree learning

As the number of contexts grows, the compactness of the parameter set is crucial to ensure training robustness. In speech recognition, phonetic contexts are often modeled by decision trees [6]. However, they were either phone or frame classification inative criteria for growing decision tree have been proposed in [7,8]. As the number of contexts grows, the compactness of the parameter set is crucial to ensure training robustness. In speech recognition, phonetic contexts are often modeled by decision trees [6]. However, they were either phone or frame classification inative criteria for growing decision tree have been proposed in [7,8].

Figure 1: Lattice and phonetic contexts

Figure 2: Tree node splitting.

from global weight $\lambda^g = (\lambda_g, \lambda_u, \lambda_p)$ to a new parameter set $\lambda^s = \{\lambda'_1, \lambda'_2, \lambda'_k\}$ and calculate the corresponding objective $F_{\text{MPE}}(\lambda^s)$. Finally, we evaluate the difference between the two objective function values:

$$
G_{\text{MPE}}(q) = F_{\text{MPE}}(\lambda^s) - F_{\text{MPE}}(\lambda)
$$

(4)

The node is split by a question that gives the largest MPE gain:

$$
q = \arg \max_{q \in Q} G_{\text{MPE}}(q)
$$

(5)

where $Q$ is the entire possible question sets. It should be noted that weight training in Eq.(4) needs at least several iterations to converge to an optimal. To find a best question $q$ for a certain node, whenever the splitting question $q$ changes, several epochs of weight training has to be run and MPE objective are to be evaluated using the updated parameters. The computations of statistic $\gamma^\lambda_{\text{MPE}}$ used in Eq. (4) and $F_{\text{MPE}}(\lambda^s)$ calculation in Eq. (3) need forward-backward computation of the lattices that contain the clustered contexts. It would be extremely expensive when the number of training utterances and question sets are large. Efficient question selection is particularly important.

3.2. Fast question selection

In EBW based optimization, the first training iteration normally obtains the largest objective increase. Then finding the best question can be accomplished by only evaluating MPE increase after the first training iteration. Since derivative statistics $\gamma^\lambda_{\text{MPE}}$ in the first iteration can be computed offline, we only need to accumulate differential according to the coming question and update model weights using Eq. (4) and the updated parameter can be obtained immediately. Next we compute objective gain $G_{\text{MPE}}(q)$. In fact, removing $F_{\text{MPE}}(\lambda)$ in Eq. (4) does not matter because it is the same for any question $q$. Therefore, the best question can be obtained by finding the largest $F_{\text{MPE}}(\lambda^s)$:

$$
q = \arg \max_{q} F_{\text{MPE}}(\lambda^s).
$$

(6)

By using first-order approximation, we obtain:

$$
F_{\text{MPE}}(\lambda^s) \approx F_{\text{MPE}}(\lambda^g) + \frac{\partial F(\lambda)}{\partial \lambda} |_{\lambda^g} \left(\lambda^s - \lambda^g\right)
$$

(7)

where the first item on the right side is the objective using global weight and remains a constant. We denote the second item as $G^\lambda_{\text{MPE}}(q) = \frac{\partial F(\lambda)}{\partial \lambda} |_{\lambda^g} \left(\lambda^s - \lambda^g\right)$. Then we can maximize the objective by finding a question:

$$
q = \arg \max_{q} G^\lambda_{\text{MPE}}(q).
$$

(8)

Approximate Splitting, and that using Eq.(5) as Exact Splitting.
4. Experiments and results

4.1. Database and configurations
The proposed method is evaluated on a tonal syllable output Mandarin speech recognition task provided by the Microsoft Research Asia speech toolbox [9]. Language model is removed from decoding process to obtain a good evaluation of the acoustic resolution. The database contains 19,688 training and 500 testing utterances. Tree learning, model weight training and testing process is summarized as Fig.3: Training and testing lattice are generated using the acoustic model. The initial-final model time alignments are obtained. Tone features are extracted from voiced part and tone posterior probabilities of the arcs are calculated. MLP score of an arc is the log summation of frame based phone posterior probabilities within the arc. Using the training lattices, we grow the phonetic trees. Because the parameters in leaf nodes are not fully optimized during tree growth, 10 weight training epochs are performed after the trees are built. Using the optimized parameters in the tree nodes, the last phase is to rescore within the testing lattices by combining scores from the models/classifiers.

4.2. The integrated model/classifiers
(1) Acoustic Model (AM). The acoustic model is MPE trained, tied-state triphone HMMs. The spectral front-end uses 39-dim vector, consisting of 12 MFCCs and normalized log energy and their $\Delta$ and $\Delta\Delta$. The HMM set has 2392 tied states with 8 Gaussians per state. Then the training and testing lattices are regenerated using the MPE trained acoustic model. The acoustic score of each arc is calculated and time alignments within the arcs are obtained for tone score and phone score calculations.

(2) Tone Model (TM). The tone models are GMMs trained on overlapped diphone segmental feature [10]. The time-scale normalized $F_0$, normalized log energy, average $\Delta F_0$ of current syllable and time normalized $F_0$ of preceding syllable are used as the input. EM training is used to initialize the GMMs and discriminative training is run to get better tone classification rate. The GMMs have variant number of Gaussians of 10, 10, 9, 16 and 3 for Tone1 to Tone5. Tone classification error rate on the test set is 28.5%.

(3) MLP phone classifier (MLP). A context window of 9 successive MFCC frames was used as the input, which amounts to 351 input units. The numbers of hidden units is 5000 and the number of output unit corresponds to the number of toneless phonemes, which is 66 (including 'sil' and 'sp'). The phone classification error rate is 23.3% on the test set. The performances of the three models are summarized in Table 1.

4.3. Tree building and question set
Tree growth is started by placing each tonal syllable to a tree root and 1,497 trees are initialized. Note that the approximate condition in Eq.(7) is not always satisfied. But we can observe the tree node questions selected by approximate splitting are about 60% similar to those obtained by exact splitting. We also observed for a tree node, the best question selected by exact splitting is always among the $N_q$ best questions selected by approximate splitting. Then the exact splitting process can be accelerated by first pruning all the question lists to $N_q$ best using approximate splitting and find the exact best question within the $N_q$ questions. In fact, the difference of the results between exact splitting and approximate splitting is trivial, but approximate splitting is much faster, which can finish tree growth within less than 1xRT on a Intel Q9400 CPU in our experiments. The results in latter experiments are reported from approximate splitting.

The question set we use are modified from those in MSR speech toolbox [9] which had been used to build triphone models. The set were designed according to the articulatory attributes of mandarin speech. There are 99 question sets used for final tree building. The question consider the final type of the preceding arc, the initial type of the following arc; the tone type of its preceding arc; whether current arc precedes or follows a silence portion within the hypothesis. Here are some sample question lists:

- QS_0 {'*:/*-*/+b', '*:/*-*/+p', '*:/*-*/+m'}
- QS_35 {'a:*/*/++*, an:*/*/++*, ao:*/*/++'}
- QS_92 {'*:/*++*'}
- QS_97 {'sil:0/*++*'}
- QS_98 {'*:/*++*sil'}

4.4. Results and discussions
Table 2 demonstrate direct integration without weight training, the model scores are combined using global settings:

$$\psi(a) = \lambda_A \psi_A(a) + \lambda_T \psi_T(a) + \lambda_M \psi_M(a) + \psi_{WP}$$

where $\psi_A$, $\psi_T$ and $\psi_M$ are respectively the AM, TM and MLP score for arc $a$. We first fix the global weight $\lambda = (\lambda_A, \lambda_T, \lambda_M) = (1/3, 1/3, 1/3)$, then the constant factors $\alpha = 3.0$, $\beta = 45.0$, $\gamma = 2.4$ and word penalty $\psi_{WP} = 30$ are selected by using cross validation and remain fixed during latter weight training. When the tree models are combined using global weighting, tonal syllable error rate (TSER) is 32.7%.

Then we experiment with CD weighting using manually designed contexts. After the context set is selected, weight parameters are optimized using Eq.(4) for 10 iterations, and lattices are rescored using the resulting parameters. All the weighting factors are initialized from global weight (1/3, 1/3, 1/3). We evaluated 4 weighting schemes: The scheme of CT considers the center initial-final type of the arc, with all the contexts of $[a-b]$. CR considers center syllable with its right initial type $[a-b+e]$, and CLR considers the entire contexts $[c-dla-b+e]$. Table 3 gives the results of CD weighting. $N_w$ is the number of the tunable weighting components. For the four weighting schemes, $N_w$ are 1,497, 231K, 42K and 4.7M, respectively. The TSERs of CT, CR and CLR considerably reduced from 32.7% achieved by global weight baseline to 31.9%, 28.6%, 30.9%, and 28.8%, respectively. The CLR context introduce the largest error reduction (4.1% better than global weight baseline). CL is 2.3% absolute better.

<table>
<thead>
<tr>
<th>Model</th>
<th>Test</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM(MPE)</td>
<td>Tonal syllable recognition</td>
<td>40.9</td>
</tr>
<tr>
<td>TM</td>
<td>Tone classification</td>
<td>28.5</td>
</tr>
<tr>
<td>MLP</td>
<td>Phone classification</td>
<td>23.3</td>
</tr>
</tbody>
</table>
Table 2: Results of global combination.

<table>
<thead>
<tr>
<th>Context</th>
<th>AM</th>
<th>TM</th>
<th>MLP</th>
<th>TSER(%)</th>
<th>∆ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPE no yes</td>
<td>40.9</td>
<td>34.8</td>
<td>37.1</td>
<td>32.7</td>
<td></td>
</tr>
<tr>
<td>MPE yes no</td>
<td>0</td>
<td>14.9</td>
<td>9.3</td>
<td>20.1</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Recognition results of CD weighting.

<table>
<thead>
<tr>
<th>Context</th>
<th>N_w</th>
<th>TSER(%)</th>
<th>∆ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>1</td>
<td>32.7</td>
<td>0</td>
</tr>
<tr>
<td>Center (CT)</td>
<td>1.5K</td>
<td>31.9</td>
<td>2.4</td>
</tr>
<tr>
<td>CT+Left (CL)</td>
<td>231K</td>
<td>28.6</td>
<td>12.5</td>
</tr>
<tr>
<td>CT+Right (CR)</td>
<td>42K</td>
<td>30.9</td>
<td>5.5</td>
</tr>
<tr>
<td>CT+Left+Right (CLR)</td>
<td>4.7M</td>
<td>29.8</td>
<td>8.9</td>
</tr>
<tr>
<td>Tree QSET1</td>
<td>7.7K</td>
<td>28.9</td>
<td>11.3</td>
</tr>
<tr>
<td>Tree QSET2</td>
<td>9.3K</td>
<td>27.5</td>
<td>15.9</td>
</tr>
</tbody>
</table>

whitened MLP log posteriors and 3-dim tonal feature (interpolated \( F_0, \Delta \) and \( \Delta \Delta \)). The TSER of the feature-combined system is 29.6%. As shown, our tree based result is 2.1% better. We think the gain is mainly due to model scaling using long-span contextual information. The most attractive of our method is that it does not require too many manual tests and selections for a model combination task as the potential number of heterogeneous models and phonetic context option increases.

5. Conclusion

We have explored automatic induction of contexts for model combination in lattice rescoring. The contexts are modeled by phonetic decision trees which are built according to the maximization of MPE objective. Results have shown the method is effective in finding useful contexts and reducing the number of underlying parameters, and hence improving the robustness to overtraining. The method is promising for optimal integration of heterogeneous model sources without manual decision of what contexts are to be utilized.

6. Acknowledgements

This work was supported by the NSFC (60965002), Scientific Research Program of the Higher Education Institution of Xinjiang (XJEDU2008S15), and Ph.D research fund in Xinjiang university (BS090143).

7. References