A STUDY OF ADAPTATION TECHNIQUES ON A VOICEMAIL TRANSCRIPTION TASK

Jing Huang and Mukund Padmanabhan
IBM T. J. Watson Research Center
P. O. Box 218, Yorktown Heights, NY 10598
email: jhuang,mukund@watson.ibm.com

ABSTRACT

Speaker adaptation techniques have emerged as very effective and practical methods to improve ASR performance on a test speaker with only limited speech data from the speaker. We explore the use of adaptation techniques on a new Voicemail database and present some theoretical extensions of the Cluster Transformation (CT) technique. Our experiments on 40 hours of voicemail data and four clusters shows that using cluster information with MLLR improves over baseline MLLR by 2.2% (relative). When the amount of adaptation data in a short message is insufficient to reliably decide its cluster, higher improvements result when we use MLLR for the very short messages and CT on longer ones.

1. INTRODUCTION

Speaker adaptation techniques have emerged as very effective and practical methods to improve ASR performance on a test speaker with the use of a limited amount of speech data from the speaker. Most of these techniques generally start with speaker-independent (SI) acoustic models, and adapt them in some way [1, 2]. Other techniques attempt to obtain better performance by starting from models that are better matched to the test speaker than the SI model [3, 4, 5]. We explore the use of these techniques on a new Voicemail database [7], and present some theoretical extensions of the Cluster Transformation technique of [3]. In particular, in [3, 6], the training data was clustered into several classes, and one or many clusters that were close to the test speaker were selected, and transformed independently to come closer to the test speaker. Subsequently, the transformed models were linearly combined. Here we establish the theoretical framework for computing these transforms in a joint manner, and also combining them so as to maximize the likelihood of the adaptation data.

2. SPEAKER CLUSTERING

The speaker clustering problem is the following: given voicemail data consisting of telephone messages from unknown speakers, the goal is to group the messages into “acoustically similar” speakers. We model each message (which is on average 18 seconds long) by a Gaussian distribution with a full covariance matrix. Subsequently, we use a bottom-up clustering method that starts with one leaf for each message and merges leaves together based on a loglikelihood ratio distance measure. Note that once the tree is constructed, one can choose any number of clusters as desired.

Hence the training messages are pre-clustered into a specified number of clusters, and for each cluster, a cluster-dependent system is trained using only the speech data from this cluster. In order to account for the possibility that the clusters may not contain enough data to robustly estimate the parameters of the cluster-dependent models, we use Bayesian adaptation techniques [2] to smooth each cluster-dependent model back to the speaker-independent (SI) model.

When a test message is given, the distance between the cluster-dependent models and the test speaker is computed as follows: first, the test message is decoded using a SI system to generate a transcription. Then the data is Viterbi-aligned against the transcription and each acoustic observation is tagged with a phonetic state. The distance is defined as the likelihood of the tagged acoustic observations, computed using the cluster-dependent model (ignoring observations tagged as silence, mumble, etc.). The cluster models are then ranked according to these distances, and the closest cluster or the closest few clusters are chosen. Then, the model for each of the selected cluster(s) is transformed to bring the model closer to the test message (Section 3).

3. CLUSTER TRANSFORMATION

We experimented primarily with linear transformations to bring the cluster models closer to the test speaker’s adaptation data. In the following we describe how to transform one cluster and multiple clusters separately and present the experimental results in Section 4.

3.1. Closest cluster

Only the closest cluster is chosen and the means of its model are transformed to maximize the likelihood of the adaptation data. This can be done either by computing the transformation to be applied on the means of the full scale model directly (MLLR) [1], or by computing a transformation that is applied to the training data, and re-estimating the full scale model means from the transformed training data (Cluster Transformation) [3]. In the latter case a much

We would like to acknowledge the support of DARPA under Grant MDA972-97-C-0012 for funding part of this work.
smaller model than the full scale model is used to compute the transformation; subsequently, the computed transformation is applied to the training data.

3.2. Multiple clusters

3.2.1. Independent computation of transforms

For the case where multiple clusters are chosen, the transformation for each cluster model is computed independently of all the rest. Subsequently the independently transformed models are combined to construct the speaker adapted model. For the case of CT, the adapted means are simply re-estimated from the transformed training data of all the selected clusters. For the case where the full scale cluster models have been independently transformed using MLLR, the weights of the linear combination of the transformed models may also be computed to maximize the likelihood of the adaptation data. Similar work has been reported in [8], where the untransformed cluster models are combined to form the adapted model.

Notation: We will use the following notations to describe the algorithms. $y_t^C$ denotes a stream of $T$ feature vectors, with the $i$-th feature vector being denoted $y_t^i$. $s_t^C$ denotes the state sequence (unknown) corresponding to the stream of feature vectors. $\theta_n$ denotes the parameter values at $n$-th iteration and $p_0(\cdot)$ indicates that the probabilities have been computed with the parameter values $\theta$. The number of cluster models is $C$. The $i$-th Gaussian in the $j$-th cluster model is denoted $\mu_{i,j}^d$ — this represents a $d$-dimensional vector, and the $d$-th component of the vector is denoted $\mu_{i,j,d}^d$. The covariance of the $i$-th Gaussian of cluster $c$ is denoted $\Sigma_c$. This is a $d$-dimensional diagonal matrix.

Assume that the adapted mean for the $i$-th Gaussian is denoted $\mu_{i}^{ad}$, and the $d$-th component of this vector is denoted $\mu_{i,d}^{ad}$. The adapted mean is assumed to be related to the cluster means by the formula:

$$\mu_{i,d}^{ad} = \frac{1}{C} \sum_{c=1}^{C} \lambda_c m_c^i$$  \hspace{1cm} (1)

where $m_c^i$ represents the transformed mean of the $c$-th Gaussian (we will assume that the $A_c$ have already been computed by some means, and the parameters to be estimated are $\lambda_c$).

We want to choose the parameters so as to maximize the likelihood of the adaptation data. This requires the maximization of the following objective function:

$$\text{max} \sum_{i,t} p_{\theta_{k-1}}(s_t^i, y_t^i) \log \left[ p_{\theta}(y_t^i / s_t^i) \right]$$  \hspace{1cm} (2)

Using $c_i(t)$ to denote $p_{\theta_{k-1}}(s_t^i = i / y_t^i)$, we can rewrite (2) as

$$\text{max} \sum_{i,t} c_i(t) \left[ \sum_{d} \frac{\left( y_{i,d,t} - \sum_{c} \lambda_c m_c^{i,d} \right)^2}{\sigma_{i,d}^2} \right]$$  \hspace{1cm} (3)

where we have ignored terms that are not dependent on the parameters being optimized.

Differentiating (3) with respect to $\lambda_j$ and setting the derivative equal to zero yields

$$\sum_{i,t} c_i(t) \sum_{d} \left[ \frac{y_{i,d,t} - \sum_{c} \lambda_c m_c^{i,d}}{\sigma_{i,d}^2} \right] \mu_{i,d}^{j,d} = 0$$  \hspace{1cm} (4)

$$\sum_{i,t} c_i(t) \left[ \sum_{d} \frac{y_{i,d,t} - \sum_{c} \lambda_c m_c^{i,d}}{\sigma_{i,d}^2} \right] \mu_{i,d}^{j,d}$$  \hspace{1cm} (5)

$$= \sum_{c} \sum_{i,t} c_i(t) \left[ \sum_{d} \frac{\mu_{i,d}^{c,d} - \mu_{i,d}^{ad}}{\sigma_{i,d}^2} \right] \lambda_c$$  \hspace{1cm} (6)

From (6), we solve a system of $C$ equations for $\lambda$.

3.2.2. Joint computation of transforms

In the previous section, the transforms $A_c$ for the selected clusters were computed independently of each other. However, under the assumption that several cluster models will be used to obtain the adapted model, it makes more sense to compute the transformations of the individual cluster models jointly so as to maximize the likelihood of the adaptation data.

As before, assume that the adapted mean for the $i$-th Gaussian is denoted $\mu_{i}^{ad}$, and the $d$-th component of this vector is denoted $\mu_{i,d}^{ad}$. The adapted mean is assumed to be related to the cluster means by the formula:

$$\mu_{i,d}^{ad} = \sum_{c=1}^{C} \lambda_c A_c m_c^i$$  \hspace{1cm} (7)

where $m_c^i$ is a vector defined as $m_{c,i}^{c,T} = \left[ m_{c,i}^{c,T} \right]_1$, and $\lambda_c$ (a scalar) and $A_c$ (a matrix) are the parameters to be estimated. Also, let $A_{c,d}$ denote the $d$-th row of the matrix $A_c$.

The objective function to be maximized now becomes

$$\text{max} \sum_{i,t} c_i(t) \left[ \sum_{d} \frac{\left( y_{i,d,t} - \sum_{c} \lambda_c A_{c,d} m_c^{i,d} \right)^2}{\sigma_{i,d}^2} \right]$$  \hspace{1cm} (8)

where we have ignored terms that are not dependent on the parameters being optimized.

The quantities to be solved for in (8) are $\lambda_c$ and $A_{c,d}$. We will adopt a two-stage approach to optimizing these — first, we will solve for $A_{c,d}$ assuming that $\lambda_c$ are known, and then solve for $\lambda_c$ plugging in the values of $A_{c,d}$ computed previously. It can be shown that this procedure will result in a non-decreasing objective function from iteration to iteration.

Assuming that $\lambda_c$ is known, to simplify notation, defining the vectors $A_{c,d} = \left[ A_{c,d}^{1} \cdots A_{c,d}^{C} \right]$, and $m_{c,i}^{c,T} = \left[ \lambda_c m_{c,i}^{c,T} \cdots \lambda_c m_{c,i}^{c,T} \right]$, we can rewrite (3) as

$$\text{max} \sum_{i,t} c_i(t) \left[ \sum_{d} \frac{\left( y_{i,d,t} - A_{c,d} m_{c,i}^{c,T} \right)^2}{\sigma_{i,d}^2} \right]$$  \hspace{1cm} (9)
Differentiating (9) with respect to $A_{il,t,d}$, we get

$$
\sum_{i,t,d} c_i(t) \left( \frac{y_{il,t,d} - A_{il,t,d}^T m_{il,t}^{all}}{\sigma_{il,t,d}^2} \right) m_{il,t}^{all} = 0
$$

Solving the system of linear equations in (10) yields $A_{il,t,d}$. Next, using the computed value of $A_{il,t,d}$, the $\lambda_c$ can be solved for using equation 6.

4. EXPERIMENTAL RESULTS

This section summarizes the results of various experiments. The speaker-independent system has 2709 context-dependent leaves (states) and 71379 Gaussians modeling the leaves. The test data consists of 43 voicemail messages. The length of the messages ranges from 5 secs to 36 secs. The baseline WER of SI system is 38.0%. The training data is divided into 2, 4, 8, or 16 clusters using the unsupervised clustering technique in Section 2.

4.1. Closest cluster

In this subsection, we report the results of adaptation experiments that select the closest cluster and build the speaker-dependent system from the transformed cluster model. We use both MLLR on the closest cluster (referred to as (CMLLR)) and CT for computing the cluster transformation. In the former case, the variances of the gaussian models may be cluster-independent or cluster-dependent, and we present results for both cases. In the latter case, the variances are cluster-independent.

The performance for various numbers of clusters is shown in Table 1. The notation (m, n) indicates that the training data is divided into m clusters, and we select n of them as being close to the test speaker. The baseline number to compare with is MLLR (WER is 36.0%) which gives a 5.3% relative improvement over the speaker-independent error rate. In contrast, the clustering of the training data does appear to help; CMLLR with 4 clusters is better than MLLR by about 2.2%, and CT with 4 clusters is about 3.6% better than MLLR.

<table>
<thead>
<tr>
<th>Method</th>
<th>(2, 1)</th>
<th>(4, 1)</th>
<th>(8, 1)</th>
<th>(16, 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMLLR-civar</td>
<td>36.0</td>
<td>35.8</td>
<td>35.4</td>
<td>35.3</td>
</tr>
<tr>
<td>(Rel Impr)</td>
<td></td>
<td></td>
<td></td>
<td>(1.9)</td>
</tr>
<tr>
<td>CMLLR-cdvar</td>
<td>35.3</td>
<td>35.2</td>
<td>35.8</td>
<td>35.9</td>
</tr>
<tr>
<td>(Rel Impr)</td>
<td></td>
<td>(2.2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT</td>
<td>35.7</td>
<td>34.7</td>
<td>36.0</td>
<td>36.2</td>
</tr>
<tr>
<td>(Rel Impr)</td>
<td></td>
<td>(3.6)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Closest cluster with CMLLR and CT.

4.2. Multiple clusters

In this subsection, we report the results of adaptation experiments that select several clusters as being closest to the test speaker. Again the transformations are computed by either MLLR or CT. Additionally, in the former case, the transformations of the individual cluster means can be computed independently of one another (CMLLR-i), or jointly (CMLLR-j). In both cases, once the transformation for the selected clusters have been computed, the transformed means are combined using the $\lambda$’s computed in (6). We present results for both cases (see Table 2). All experiments use cluster-independent variances.

Among the three approaches, CT gave the best word error rate. This may be because a much smaller model was used to compute the MLLR transforms which are consequently more robust.

<table>
<thead>
<tr>
<th>Method</th>
<th>(4, 2)</th>
<th>(8, 2)</th>
<th>(16, 2)</th>
<th>(8, 4)</th>
<th>(16, 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMLLR-i</td>
<td>36.9</td>
<td>36.5</td>
<td>36.1</td>
<td>36.2</td>
<td>35.9</td>
</tr>
<tr>
<td>(Rel Impr)</td>
<td></td>
<td>(0.3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMLLR-j</td>
<td>36.4</td>
<td>36.0</td>
<td>35.9</td>
<td>36.5</td>
<td>35.9</td>
</tr>
<tr>
<td>(Rel Impr)</td>
<td></td>
<td>(0.3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT</td>
<td>35.8</td>
<td>35.1</td>
<td>36.9</td>
<td>35.6</td>
<td>34.7</td>
</tr>
<tr>
<td>(Rel Impr)</td>
<td></td>
<td>(3.6)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Performance with closest few clusters.

4.3. Gender-dependent models

We also experiment with gender-dependent clusters to compare the performance with our unsupervised speaker clustering. The training data of each gender is divided into 2 clusters using the speaker clustering procedure in Section 2. The results are tabulated in Table 3, and seem to suggest that unsupervised speaker clustering gives better performance than gender supervised clustering.

<table>
<thead>
<tr>
<th>Method</th>
<th>baseline</th>
<th>MLLR</th>
<th>(4, 1) MLLR</th>
<th>(4, 1) CT</th>
</tr>
</thead>
<tbody>
<tr>
<td>WER</td>
<td>38.0</td>
<td>36.0</td>
<td>35.8</td>
<td>35.6</td>
</tr>
<tr>
<td>Rel Impr</td>
<td></td>
<td></td>
<td>(1.2)</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Gender-dependent clustering and adaptation.

4.4. The effect of adaptation data

An important problem in voicemail is data sparsity: lack of both training data and adaptation data, as phone mail messages may be very short in reality. In addition, if we are constrained to use only the nearest cluster, the data is too little for this approach to be reliable. Therefore we may end up selecting the wrong cluster-dependent model for this message. Figure 1 shows how MLLR and cluster transformation behave on messages of different length.
Comparison of MLLR and CT test messages in the order of increasing length

Figure 1: Comparing MLLR and CT on the 43 test messages.

It is clear that CT does not perform as well as MLLR on very short messages; however, in most cases it is better than MLLR on relatively long messages. This led to the following strategy: use MLLR for very short messages and use CT on (relatively) long messages. This approach gives WER of 34.00%, which is an 10.5% improvement over baseline, and 5.6% improvement over MLLR.

5. CONCLUSION

In this paper we presented studies of adaptation techniques on the Voicemail transcription task. The adaptation techniques are based on pre-clustering training data and building cluster-dependent systems. We extended the framework of Cluster-Transformation to estimate the transforms for several clusters jointly. Our experiments have shown that these adaptation techniques reduce the word error rate on the voicemail transcription task and improve upon MLLR. In our future work we will investigate the effectiveness of transforming covariances and the use of regression class trees for MLLR.

6. REFERENCES


