MAXIMAL RANK LIKELIHOOD AS AN OPTIMIZATION FUNCTION FOR SPEECH RECOGNITION

Yuqing Gao, Yongxin Li, Michael Picheny

IBM Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY10594, USA

ABSTRACT

Research has shown that rank statistics derived from context-dependent state likelihood can provide robust speech recognition. In previous work, empirical distributions were used to characterize the rank statistics. We present parametric models of the state rank and the rank likelihood, and then based on them, present a new objective function, Maximal Rank Likelihood (MRL), for estimating parameters in a HMM based speech recognition system. The objective function optimizes the average logarithm of the rank likelihood of training/adaptation data. It is a discriminative based estimation process and hence makes the training criterion close to the decoding criterion. Three applications of MRL are discussed. First one is a Linear Discriminative Projection, which optimizes the objective function using all training data and projects feature vectors into a discriminative space with a reduced dimension. The second and third applications are a feature space transformation and a model space transformation, respectively, for adaptation. The transformations are optimized to maximize the rank likelihood of the adaptation data. The experimental results show that the MRL adaptation algorithms outperform the MLLR adaptation.

1. INTRODUCTION

The goal of training\(^1\) is to find the HMM parameters which will result in a speech recognizer with the lowest possible recognition error rate. The training or adaptation is done by maximizing some objective function \(F(\lambda)\). A meaningful objective function should satisfy conditions that, whenever \(F(\hat{\lambda}) > F(\lambda)\), \(\hat{\lambda}\) results in a better decoder than \(\lambda\). This is not always true when the likelihood \(P(O|\lambda)\) is used as the objective function because there is no direct relation between the likelihood and the recognition error rate.

It has been shown that an HMM based recognizer trained with Maximum Likelihood Estimation (MLE) can be improved further using discriminative training, such as Maximum Mutual Information Estimation (MMIE, [3]). The purpose of this work is to find an objective function which not only maximizes the discrimination between classes on training data, but also moves the criterion used in parameter estimation of a speech recognition system closer to the decoding criterion, therefore reducing the recognition error rate. In this paper we present a new optimization function which tries to satisfy these two objectives.

Let us briefly review the decoding procedure of the IBM rank based speech recognition system [2]. During decoding, instead of Gaussian likelihoods, rank likelihoods are used to form search paths in order to achieve better robustness.

\(^1\) Adaptation is considered as training, because it is constrained parameter estimation.

The rank likelihood is obtained from a pre-computed rank likelihood distribution, which typically has a peak at rank one and rapidly falls off to low probabilities for lower ranks. Gaussian likelihoods are used to rank all HMM states, given an observation vector of a frame of speech. Each frame has a state list associated to it. In the list all the states are sorted in the decreasing order of the Gaussian likelihood. The decoding performance of a system depends on the quality of the rank lists. One way to improve the system performance is to make the correct state for every frame to appear on the top positions of the rank list, i.e., to make every frame have higher rank, therefore improving the rank likelihoods of the correct states.

Given an observation vector \(o_t\) (corresponding to a frame of speech), the rank \(r(o_t)\) for this observation is defined as the number of other (confusable) states \(l^r\) which have higher Gaussian likelihoods than the correct state \(l^r\) conditioned on the correct transcription and alignment. We use the following equation to formalize this definition.

\[
r(o_t) = \sum_{l^r = 1}^{\hat{L}} u \left( \log \frac{p(o_t|l^r)}{p(o_t|l^*)} \right)
\]

where \(p(o_t|l^r)\) is the likelihood of \(o_t\) given the correct state \(l^r\), \(p(o_t|l^*)\) is the likelihood of \(o_t\) given any other state \(l^*\), \(l^r \in \{all HMM states in the underlying system\}\). \(u(\cdot)\) is a Step function:

\[
u(x) = \begin{cases} 1 & \text{if } x \leq 0 \\ 0 & \text{if } x > 0 \end{cases}
\]

The rank likelihood of \(o_t\) is defined as:

\[
RL(o_t) = L(r(o_t))
\]

where, \(r(o_t)\) is the rank as in Eq.1, \(L(\cdot)\) is a rank likelihood function. Although in practice we pre-compute it as a histogram from a small portion of training data, theoretically, any monotonic decreasing function can be used to represent this function.

From Eq.1 and Eq.2, it can be seen that in order to maximize the rank likelihood \(RL(o_t)\), not only \(p(o_t|l^*)\) should be maximized, but also all \(p(o_t|l^r), l^r \in \{all HMM states\}\) should be minimized relatively to \(p(o_t|l^*)\). So the rank likelihood is a discriminative function. If we choose to use the rank likelihood as an objective function, not only the training is discriminative, but also the training criterion is same as the decoding criterion. Unfortunately, the rank is defined as a discrete function in Eq.1 and therefore the rank likelihood Eq.2 cannot be used as an objective function directly. In next section (Section 2), we introduce a pseudo rank, a pseudo rank likelihood and a new objective function which is based on the pseudo rank likelihood.
2. PSEUDO RANK LIKELIHOOD AS AN OBJECTIVE FUNCTION

We choose to use a Sigmoid function \( \sigma(x) \): 
\[
\sigma(x) = \frac{1}{1 + e^{-x}} 
\]
(3)
to replace the Step function \( u(x) \) in Eq.1. 
The larger \( \alpha \) is, the closer is \( \sigma(.) \) to \( u(.) \). However, in practice, a large \( \alpha \) could cause instability of \( \sigma(.) \). 

With this smoothed \( \sigma(.) \), the pseudo rank is defined as: 
\[
r(o_t) = \sum_{i=1}^{L} \sigma \left( \log \frac{p(o_t|F^t)}{p(o_t|F)} \right) 
\]
(4)

In practice, Eq.4 is an excellent approximation to Eq.1. 
Table 1 shows how the pseudo rank compares to the real rank when \( \alpha = 10 \). Each column compares the rank and the pseudo rank on different chunks of training data, each chunk has a size of 1000 sentences and is randomly chosen from the training data.

We choose to use the reciprocal function as the rank likelihood in Eq.2:
\[
L(r) = \frac{1}{r+1} 
\]
(5)

It is called a pseudo rank likelihood and turns out to be similar to the pre-computed rank likelihood distribution.

Given an observation sequence \( o_1, o_2, \ldots, o_T \), the objective function is defined as the average logarithm of the pseudo rank likelihood:
\[
F = \frac{1}{T} \sum_{t=1}^{T} \log \left( L(r(o_t)) \right) 
\]
\[
= \frac{1}{T} \sum_{t=1}^{T} \log \left( \frac{1}{T} \sum_{i=1}^{T} \sigma \left( \log \frac{p(o_t|F^t)}{p(o_t|F)} \right) \right) 
\]
(6)

The objective function in Eq.6 can theoretically replace the Gaussian likelihood to improve all MLE based algorithms. We discuss its application to different fields of speech recognition in the next section.

Before we go further, let us investigate the relation between the rank likelihood and the recognition accuracy from a sample of 10 speakers. Table 2 shows the relative improvement of speaker adapted systems (20 minutes of adaptation speech for each speaker) over a speaker independent baseline system. The 2nd, 3rd and 4th columns are relative improvements on the recognition accuracy, the rank and the pseudo rank likelihood, respectively. The last one is the ratio of the accuracy improvement to the pseudo rank likelihood improvement (column 2 vs. column 4). If the values of this ratio for all the speakers are same, it means that the rank likelihood improvement and the recognition accuracy improvement are linearly related, then maximizing the rank likelihood should result in minimizing the recognition error rate of the system. From Table 2, it can be seen although the values are not exactly same, they are not very different from each other and are all around 2. This example suggests that the objective function in Eq.6 is good.

<table>
<thead>
<tr>
<th>Set</th>
<th>Accuracy</th>
<th>Rank</th>
<th>SRankLik</th>
<th>SRLik/Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28.7</td>
<td>21.4</td>
<td>12.3</td>
<td>2.33</td>
</tr>
<tr>
<td>2</td>
<td>43.4</td>
<td>34.8</td>
<td>19.1</td>
<td>2.27</td>
</tr>
<tr>
<td>3</td>
<td>53.5</td>
<td>42.0</td>
<td>23.9</td>
<td>2.24</td>
</tr>
<tr>
<td>4</td>
<td>27.8</td>
<td>22.9</td>
<td>12.9</td>
<td>2.16</td>
</tr>
<tr>
<td>5</td>
<td>36.0</td>
<td>23.7</td>
<td>13.3</td>
<td>2.71</td>
</tr>
<tr>
<td>6</td>
<td>34.5</td>
<td>28.2</td>
<td>17.0</td>
<td>2.03</td>
</tr>
<tr>
<td>7</td>
<td>32.4</td>
<td>25.7</td>
<td>14.7</td>
<td>2.20</td>
</tr>
<tr>
<td>8</td>
<td>32.1</td>
<td>27.8</td>
<td>14.9</td>
<td>2.15</td>
</tr>
<tr>
<td>9</td>
<td>35.8</td>
<td>31.3</td>
<td>18.3</td>
<td>1.96</td>
</tr>
<tr>
<td>10</td>
<td>23.5</td>
<td>12.6</td>
<td>8.19</td>
<td>2.87</td>
</tr>
</tbody>
</table>

Table 2: Relation between accuracy and rank likelihood

Since there is no closed form solution to the problem of maximal pseudo-rank likelihood, we have to solve the problem numerically. In fact, we search along the gradient direction of the objective function. Let
\[
d_{\lambda}(o_t, F^t, i^t) = \log \frac{p(o_t|F^t)}{p_{\lambda}(o_t|F^t)}, 
\]
(7)
then the gradient of the objective function \( F \) with respect to \( \lambda \) (the desired parameters of the recognition system to be optimal), \( \frac{\partial F}{\partial \lambda} \), can be written as:
\[
\frac{\partial F}{\partial \lambda} = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{L(r(o_t))} \sum_{i'=1}^{L} \frac{\partial L(r(o_t))}{\partial r(o_t)} \frac{\partial \sigma(d(o_t, F^t, i^t))}{\partial \lambda} \frac{\partial d(o_t, F^t, i^t)}{\partial \lambda} 
\]
(8)

where,
\[
\frac{\partial L(r(o_t))}{\partial r(o_t)} = -\frac{1}{1 + (r(o_t))^2} 
\]
\[
\frac{\partial \sigma(d(o_t, F^t, i^t))}{\partial d(o_t, F^t, i^t)} = \frac{-\alpha}{e^{\alpha d(o_t, F^t, i^t)} + 1} 
\]
For different applications, \( \frac{\partial \sigma(d(o_t, F^t, i^t))}{\partial d(o_t, F^t, i^t)} \) in Eq. 8 will be computed differently corresponding to different sets of parameters, \( \lambda \)'s. The rest will remain same.

3. APPLICATIONS

3.1. Linear Discriminant Projection

Linear Discriminant Analysis (LDA) is popularly used in speech recognition to reduce the dimension of the feature space and retain as much discriminating information as possible. LDA tries to find a \( \theta \), which projects a feature vector from the original feature space to a new feature space with reduced dimensions by maximizing:
\[
\frac{|\theta B \theta^T|}{|\theta B|} 
\]
(9)
where, \( B \) and \( T \) are the between class covariance and total covariance, respectively[1].
We present one application of the objective function (Eq. 6) and use it to replace Eq. 9. We call it a Linear Discriminant Projection (LDP), which projects the feature vector $o_t$ into a new space with reduced dimensions: $o_t \rightarrow \theta o_t$.

When one Gaussian is used to model each state, $d_s(o_t, F, l^t)$ in Eq. 6 becomes:

$$2d(\theta o_t, F, l^t) = \log(p(\theta o_t | F)) - \log(p(\theta o_t | l^t)) = -(o_t - m_{i_c})^T \theta^T \text{diag}^{-1}(\theta \Sigma_i \theta^T)(o_t - m_{i_c})$$

$$+ \log(\text{diag}(\theta \Sigma_i \theta^T)) + \log(\text{diag}(\theta \Sigma_i \theta^T)(o_t - m_{i_c})^T)$$

The gradient is same as in Eq. 8, except $\frac{\partial d_s(o_t, F, l^t)}{\partial \lambda}$ becomes

$$\frac{\partial d(\theta o_t, F, l^t)}{\partial \theta} =$$

$$-\text{diag}^{-1}(\theta \Sigma_i \theta^T)(o_t - m_{i_c})(o_t - m_{i_c})^T$$

$$+ \text{diag}(\theta (o_t - m_{i_c})(o_t - m_{i_c})^T \theta^T \text{diag}^{-1}(\theta \Sigma_i \theta^T)(o_t - m_{i_c})$$

$$- \text{diag}^{-1}(\theta \Sigma_i \theta^T)(o_t - m_{i_c})$$

$$+ \text{diag}^{-1}(\theta \Sigma_i \theta^T)(o_t - m_{i_c})(o_t - m_{i_c})^T$$

$$- \text{diag}^{-1}(\theta \Sigma_i \theta^T)(o_t - m_{i_c})(o_t - m_{i_c})^T$$

$$+ \text{diag}^{-2}(\theta \Sigma_i \theta^T)(o_t - m_{i_c})$$

In Eq. 10, in order to simplify the gradient computation we make $\theta \Sigma_i \theta^T$ and $\theta \Sigma_i \theta^T$ diagonal. It is consistent with the assumption we often make in practice, because in most applications we use Gaussians with diagonal covariances.

3.2. Maximal Rank Likelihood Adaptation - A Feature Space Transformation (MRL-FST)

In the application of LDP described in Section 3.1, the gradient $\frac{\partial d_s}{\partial \theta}$ is computed for every frame of training speech. The computation cost is proportional to the amount of training speech, which makes the algorithm impractical when the amount of training data is huge.

We present another application which optimizes the feature space transformation on adaptation data, which is usually relatively small. In this application, the dimensions of the feature vector remain same, the transformation is optimized to maximize the rank likelihood of the adaptation data.

When mixture of Gaussians is used to model a state, $p(o_t | l^t)$ becomes:

$$p(o_t | l^t) = \sum_{g=1}^{M} p_g(o_t | G_g^t)$$

where $p_g$ is the mixture weight for $g$-th Gaussian $G_g^t$ for state $l$, $p(o_t | G_g^t)$ is the Gaussian likelihood of $o_t$ conditioned on the $g$-th Gaussian $G_g^t$.

$$p(o_t | G_g^t) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_g^t|} \exp\left\{-\frac{1}{2} (o_t - m_{g_c})^T (\Sigma_g^t)^{-1} (o_t - m_{g_c}) \right\}$$

The gradient becomes:

$$\frac{\partial d(o_t, F, l^t)}{\partial \theta} =$$

$$- \sum_{g=1}^{M} \frac{f_{g}^l}{2} (\Sigma_g^t)^{-1} (\theta o_t - m_{g_c})^T (\Sigma_g^t)^{-1} (\theta o_t - m_{g_c})$$

$$+ \sum_{g=1}^{M} \frac{f_{g}^l}{2} (\Sigma_g^t)^{-1} (\theta o_t - m_{g_c})^T$$

where

$$f_{g}^l = \frac{p_g(o_t | G_g^t)}{\sum_{g=1}^{M} p_g(o_t | G_g^t)}$$

The $f_{g}^l$ can be viewed as normalization factors based on the posterior likelihoods.

3.3. Maximal Rank Likelihood Adaptation - A Model Space Transformation (MRL-MST)

Another application of the MRL estimation is also for speaker adaptation. Different from MRL-FST in Section 3.2, a model space linear transformation, which is applied to Gaussian means (similar to MLLR), can be computed to maximize the rank likelihood for observations. The objective function for this application is same as Eq. 6, however the Gaussian means become the adapted means $m_{i_c} = \theta m_{i_c}$.

When Gaussian mixtures are used to model each state, $p(o_t | l^t)$ is same as in Eq. 11.

$$p(o_t | G_g^t) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_g^t|} \exp\left\{-\frac{1}{2} (o_t - \theta m_{g_c})^T (\Sigma_g^t)^{-1} (o_t - \theta m_{g_c}) \right\}$$

The gradient becomes:

$$\frac{\partial d(o_t, F, l^t)}{\partial \theta} =$$

$$- \sum_{g=1}^{M} \frac{f_{g}^l}{2} (\Sigma_g^t)^{-1} (o_t - \theta m_{g_c})^T (\Sigma_g^t)^{-1} (o_t - \theta m_{g_c})$$

$$+ \sum_{g=1}^{M} \frac{f_{g}^l}{2} (\Sigma_g^t)^{-1} (o_t - \theta m_{g_c})^T$$

where, $f_{g}^l$ are normalization factors as in Eq.14.

4. EXPERIMENTAL RESULTS

4.1. Data and Baseline Systems

The training data is IBM’s in-house database of 200 hours of speech. The test data is from 10 speakers (6 females and 4 males), each has 10 minutes of read speech. Each speaker also has 20 minutes of adaptation speech. The baseline system is trained using all 200 hours of speech. The signal processor produces 24-dimension MFCCs. 9 frames of speech are then concatenated into a 216-dimension vector, an LDA transformation is computed reducing it to 40-dimension. The large vocabulary continuous speech recognition system has 3400 context dependent states and 42,000 Gaussians.

4.2. Experiments for LDP

The LDP matrix is 216 by 40. We use an LDA matrix as its initialization, and search along the gradient direction. The computational cost for the gradient is high due to the
algorithm complexity and the large amount training data we used. Table 4.2 shows the improvement on the average rank and the average logarithm rank likelihood of the training data, for first 3 iterations.

The absolute rank values are high in Table 4.2 because we use one Gaussian for each state in this experiment, as we often do during computing the LDA matrix. Due to the limitation of available resources, we are unable to see results from more iterations or to build a recognition system using feature vectors from the new transformed space yet. Nevertheless, the rank and the rank likelihood are being improved after each iteration.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Rank-Lik</th>
<th>Error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI</td>
<td>18.25</td>
<td>-1.85600</td>
</tr>
<tr>
<td>MLLR</td>
<td>17.43</td>
<td>-1.82374</td>
</tr>
<tr>
<td>MRL-MST</td>
<td>17.06</td>
<td>-1.7943</td>
</tr>
<tr>
<td>SI</td>
<td>16.56</td>
<td>-1.8453</td>
</tr>
<tr>
<td>MLLR</td>
<td>15.27</td>
<td>-1.7860</td>
</tr>
<tr>
<td>MRL-MST</td>
<td>14.33</td>
<td>-1.7033</td>
</tr>
</tbody>
</table>

Table 5: MRL adaptation: Model Space Transformation

4.3. Experiments for MRL-FST

In this section and Section 4.4, we present speaker adaptation experiments using MRL algorithms. Mixtures of Gaussians with diagonal covariances are used for HMM states. Results are presented only on 6 female speakers. The adaptation data used is 5 minutes of speech for each speaker.

We use the identity matrix as initials for the gradient search. All Gaussian parameters are fixed, only the feature space transformations are estimated. The rank and the rank likelihood with an associated improvement in the recognition accuracy are presented after 80 iterations (Table 4).

<table>
<thead>
<tr>
<th>Rank</th>
<th>Rank-Lik</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>15.54</td>
<td>-1.85</td>
</tr>
<tr>
<td>After iterations</td>
<td>13.76</td>
<td>-1.75</td>
</tr>
</tbody>
</table>

Table 4: MRL adaptation: Feature Space Transformation

4.4. Experiments for MRL-MST

5 minute adaptation speech is used to run MLLR and MRL-MST adaptation for each speaker. In Table 5, three systems are compared. In both MLLR and MRL-MST, only one model space transformation (for each speaker) is used for all Gaussians. The matrices are initialized with very small values, which are uniformly distributed between (-0.05, 0.05). The MRL-MST results in Table 5 are from the 66th iteration. At the point of 66th iteration, the objective function is still increasing for each speaker, i.e., the numerical search is not converged yet. The first 3 lines are for the test data, while the rest are for the adaptation data. It can be seen that the average log rank likelihood for the adaptation data has increased a lot more compared to the increase for the test data, therefore it is not surprising that the error rate of the adaptation data has decreased a lot more than the decrease for the test data.

5. DISCUSSIONS AND CONCLUSIONS

Based on the experiments we have done so far, improvement on rank likelihood has consistently resulted in higher decoding accuracy in all applications. This shows the power of the new objective function we presented in this paper. However, because MRL algorithm is frame-based discriminative and there is no closed form solution, the computational cost is extremely high. We have only tried a simple gradient search method, which turns out to converge very slowly. To make the MRL algorithm practical, further efforts are needed to speed up the convergence and to compute the gradients in a more efficient way. There are many potential ways to speed up the computation. For instance, in computing the rank, it is not necessary to compute Gaussian likelihoods for all the states. Very likely, only a small number of states are confusable with the correct state and we only need to compare them with the correct state during rank computing.

The Maximal Rank Likelihood adaptation is discriminative and is superior to MLLR adaptation. In MLLR, the Gaussian likelihoods of the adaptation data are maximized, all the means of the states which share a transformation are rotated towards the same direction, and no discrimination between these states or means are considered during the optimization. While in MRL adaptation, the rank likelihoods of the adaptation data are maximized, even when a transformation is shared by many Gaussians of different states, the transformation is computed to maximize the ratio between the Gaussian likelihood of \( \alpha_t \) given the correct state and the Gaussian likelihood of \( \alpha_t \) given any in-correct states for every observation vector.

MRL estimation can be applied to other problems, such as, estimating Gaussian means and variances of HMMs.

After finishing the research in this paper, we noticed a recent work [4]. It is to compute the model space adaptation transform using MMIE. We believe that MRL is a better objective function than MMIE. In MMIE, the maximization is for the ratio of the likelihood conditioned on the correct state to the summation of all other likelihoods conditioned on other states, this does not guarantee that the ratio of the correct state likelihood to each individual other (confusable) state likelihood is maximized, while in MRL, it is guaranteed. The decoding performance of recognition system depends on the latter ratio.

6. REFERENCES


