A Robust Training Algorithm Based on Neighborhood Information

Wing-Hei AU, Man-Hung SIU

Department of Electrical and Electronic Engineering,
Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, Hong Kong
ehe@ust.hk, eesiu@ust.hk

Abstract

Robustness is an important issue in automatic speech recognition systems. When the testing conditions do not match the training condition or when there is insufficient training data, the performance of a system trained by maximum likelihood criterion may degrade significantly. Different robust algorithms were proposed especially for cases in which the mismatch condition is known or can be estimated from the test data. In many practical cases, however, the mismatch information may not be available. In this paper, we propose a robust training algorithm that does not make any assumption about the mismatch condition. Instead, it is based on a neighborhood concept that appropriately broaden the model distributions to increase the model robustness. The proposed algorithm is evaluated in the Aurora3 tasks which shows that the neighborhood approach can reduce the degradation in mismatch conditions.

1. Introduction

In recent years, automatic speech recognition has become more and more popular and the number of speech recognition applications is increasing. Weather information retrieval [1] and call routing applications [2] are examples of such applications. Many automatic speech recognition systems can achieve high accuracies in well constrained domains. For example, for recognition of the TIDIGITS corpus, a string error rate of only a few percentages is reported [3].

However, there are still some difficulties for these systems to overcome when they are applied in real-world applications. One of these difficulties is the variations in the acoustic environment in which the systems are used. These variations can come from several sources: additive noise, channel mismatch, different accents or speaking styles, etc. This causes mismatch between training and testing conditions and can result in significant degradation in performance.

Many techniques have been proposed to address these mismatches by either adjusting the model parameters through supervised or unsupervised learning of the environment [4, 5] or via signal processing techniques to reduce the impact of noise or channel effects. If there is enough data from the noisy condition, as in the case of multi-condition training in Aurora2, training can be simply performed on both the clean and noisy data and the accuracy is expected to be higher than that of training with clean data only [6].

Another potential problem is over-training. It occurs in many training algorithms because we usually have finite training data. Very often, the resulting model is so specific that a slight deviation in the testing condition can substantially increase the errors. Purnell and Botha [7] addressed this problem in MCE parameter estimation by penalizing small variance in the loss function. Smoothing the model parameters by interpolating the well-trained general models with the less well-trained but more detailed models is another approach to solve such problem [8].

Unfortunately, in many cases, it is difficult to know the distribution of the mismatch or obtain enough real data that covers all possible conditions. Conceptually, since humans can recognize speech in different environments even the ones that they are unfamiliar with, it should be possible to create models that are robust under different unseen, mismatch conditions.

In this paper, a neighborhood based training method is proposed to improve the robustness of the models. The concept of a neighborhood was first introduced in [9] to adjust the model parameters during the test to improve recognition. In this work, we extend the neighborhood concept. Instead of creating neighborhoods around the model parameters, we create neighborhoods around the observations. To simplify the development, we also use Monte Carlo simulation to generate new observations within the neighborhood.

The rest of the paper is organized as follows. The traditional maximum-likelihood training method is reviewed in the next section. Our neighborhood based training method is described in Section 3. Some practical issues are discussed in Section 4. Experimental results and related discussions are given in Section 5 and are followed by a Conclusion in Section 6.
2. Traditional Maximum-Likelihood Training Algorithm

Suppose \( p(x|\Theta) \) is the probability density function (pdf) of the observation \( x \) parameterized by \( \Theta \). \( N \) observation vectors, denoted as \( X = \{ x_1, x_2, \ldots, x_N \} \) are drawn from this pdf and are assumed to be independent and identically distributed (i.i.d.). The likelihood of \( X \) is a function of \( \Theta \) and is defined as:

\[
p(X|\Theta) = \prod_{i=1}^{N} p(x_i|\Theta)
\]

In maximum-likelihood estimation (MLE) of parameters, the solution is to find a \( \Theta \) that maximizes the likelihood \( p(X|\Theta) \), i.e.,

\[
\Theta_{MLE} = \arg \max_{\Theta} p(X|\Theta)
\]

Such maximum-likelihood algorithm is widely used in the speech recognition communities. The estimated distributional parameters are used to “classify” speech during testing. If there are sufficient training data and the test data have the same distributions as the training data, MLE provides the optimal solution. However, if the amount of data is not sufficient and because maximizing the training likelihood often leads to sharp models, it is well-known that MLE can over-train such that the training likelihood is optimized but not the test performance. The details of MLE in speech recognition can be found in [10].

3. Neighborhood-Based Training Algorithm

The concept of neighborhood was first introduced by N. Merhav and C.-H. Lee [9]. To address the fact that the ML parameters may not be optimal for recognition, they assumed the true model parameters to lie within the neighborhood of the trained model parameters. With this approach, the search decision is not based on the likelihood derived from the ML parameters. Instead, their search algorithm would find a path that minimizes the worst case probability of error. Further extensions of the neighborhood approaches were reported in [3, 9].

3.1. Definition of Neighborhood

In this paper, we try to extend the idea of neighborhood and apply it in training. Instead of assuming the true parameters to lie in the neighborhood of the ML parameters, we assume that the observations seen in training are points sampled within a specific neighborhood. This can also be viewed as estimating a model for mismatched observations which are lying within the neighborhood of the observed training observations.

If the observations are \( D \)-dimensional, we introduce the mismatch vector \( y_i \). The observed training vectors \( x_i \) are the clean observations and the corrupted observations, \( z_i \) can be generated by summing \( x_i \) and \( y_i \). That is,

\[
z_i = x_i + y_i
\]

Using the same uniform neighborhood as in [3], [9], [11], the neighborhood, \( \eta \), is defined as the set of all possible mismatches \( y_i \). I.e.,

\[
\eta = \{ y_i : -Cd^{-1}p^d \leq y_i, d \leq Cd^{-1}p^d, \\
1 \leq d \leq D \}
\]

where \( y_i,d \) is the \( d \)-th element of \( y_i \), \( C (C > 0) \) controls the size of the uniform neighborhood and \( \rho \) \((0 \leq \rho < 1)\) controls the rate of decay across the different dimensions. This uniform neighborhood explicitly describes our lack of knowledge of possible mismatches.

3.2. Modified ML Training Algorithm

By using the neighborhood information, a new “likelihood function” \( p^\star(x_i|\Theta) \) that contains the mismatch information of an observation \( x_i \) is defined as follows:

\[
p^\star(x_i|\Theta) = \int_{y_i \in \eta} p(y_i|\varphi)p(x_i + y_i|\Theta)dy_i
\]

where \( p(y_i|\varphi) \) is the prior distribution of the mismatch \( y_i \) with parameters \( \varphi \).

Similar to MLE, the model parameters \( \Theta \) estimated from the observation set \( X \) are shown as follows:

\[
\Theta_{NML} = \arg \max_{\Theta} \prod_{i=1}^{N} p^\star(x_i|\Theta)
\]

4. Practical Issues

Similar to many automatic speech recognition systems, \( p(x_i|\Theta) \) is modeled as Gaussian mixtures. The likelihood of vector \( x_i \) would be:

\[
p(x_i|\Theta) = \sum_{k=1}^{K} \omega_k N(x_i; \mu_k, \Sigma_k)
\]

where \( \Theta \) and \( \omega_k \) represent the model parameters and the weighting of the \( k \)-th mixture respectively and \( \sum_{k=1}^{K} \omega_k = 1 \). The mixtures follow Gaussian distribution with mean \( \mu_k \) and covariance \( \Sigma_k \).

Each dimension of the possible mismatch \( y_i \) is assumed independent to each other, we perform the integration over the uniformly distributed neighborhoods, which is a “cube-like” volume centered at \( x_i \). The new likelihood function would be:
\[ p^*(x_i|\theta) = \int_{y_i \in \mathcal{Y}} p(y_i|\phi)p(x_i + y_i|\theta)dy_i \]
\[ = \prod_{d=1}^{D} \frac{1}{2\pi \sigma^d} \exp \left( -\frac{1}{2\sigma^d} (x_{i,d} - m_{i,d})^2 \right) \]
\[ \cdot \int_{y_i \in \mathcal{Y}} p(x_i + y_i|\theta)dy_i \]  
(8)

However, a direct maximization of Equation 8 is not trivial. One alternative is the Monte Carlo technique [12] in which we use the average of random samples from the neighborhood to approximate the integral. This approximated “likelihood function” \( p^{**}(x_i|\Theta) \) is:

\[ p^{**}(x_i|\Theta) = \frac{1}{M} \sum_{m=1}^{M} p_m(x_i + y_i|\Theta) \]  
(9)

where \( M \) is the number of samples taken and \( p_m(x_i + y_i|\Theta) \) is the likelihood of \( m \)-th sample. Finally, \( p^{**}(x_i|\Theta) \) replaces the \( p^*(x_i|\Theta) \) in Equation 6.

5. Experiments

5.1. Experimental Setup

The proposed training method is evaluated on the Aurora3 tasks which contain connected digits utterances from different conditions. The corpus is also divided into four language subsets, Danish, Finnish, German and Spanish. The model configuration we used was the same as the one in the reference scripts provided with the databases. That is, each digit model was represented by 16 real HMM states and each state was modeled by three Gaussian mixtures.

The ETSI Advanced DSR (WI008) standard front-end was used [13]. The features included the 12 static cepstrum coefficients, energy and their first and second order derivatives. Similar to [3], the sizes of the neighborhoods used in the first and second order derivatives were the same as that of the neighborhood of their corresponding static coefficients. Variance normalization mentioned in [14] is applied on the features. Speech files were segmented according to the endpoints derived from the channel 0 data and they are padded with 200ms at each end.

The constants in the neighborhood, \( C \) and \( \rho \), were tuned in the Danish subset and applied to the other subsets. Because variance normalization were applied which also removed the dynamic range difference between different cepstral coefficients, it was expected that \( \rho \) obtained from tuning will be very close to one. The number of samples to take during training, \( M \), is another parameter to be set. Obviously, larger \( M \) results in better approximation but the computation time increases significantly. Under this constraint, the number of samples, \( M \), was set to 10 in all the experiments reported in this paper.

5.2. Results

In the Danish subset, the best neighborhood is \( C = 1.5, \rho = 0.99 \). The overall results of the four subsets that using this neighborhood is shown in Table 1. The accuracy of the baseline system is 91.01% and applying variance normalization (Var-norm) provides 11.23% relative improvement in the WER. Applying the neighborhood based training (N-train) provides a further 4.01% relative improvement in the WER.

<table>
<thead>
<tr>
<th></th>
<th>WM</th>
<th>MM</th>
<th>HM</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>95.63</td>
<td>87.70</td>
<td>88.26</td>
<td>91.01</td>
</tr>
<tr>
<td>Var-norm</td>
<td>95.81</td>
<td>89.62</td>
<td>89.31</td>
<td>92.02</td>
</tr>
<tr>
<td>N-train</td>
<td>95.54</td>
<td>90.13</td>
<td>90.32</td>
<td>92.34</td>
</tr>
</tbody>
</table>

Table 1: Overall accuracy (%) of the neighborhood based training algorithm and baseline

Comparing the first column with the second and the third columns of Table 1, the performance improved in the medium-mismatch and the high-mismatch conditions but the accuracy dropped in the well-matched condition. Degradation in the well-matched condition is expected because the models are not only estimated for the condition of the training data, the presence of mismatch is now taken into account in the training phase also. Surprisingly, there is still improvement even with a small \( M \).

![Figure 1: Size of the neighborhood (C) versus accuracy at \( \rho = 0.99 \), in the Danish subset](image)

There is a comparison of the performance of different neighborhood sizes for the Danish subset in Figure 1. An interesting observation from the figure is that the model trained with the larger of neighborhood size, \( C \), performs better in the higher mismatch conditions. This suggests that it is possible to make a trade off between the robustness of the model and the performance under matching conditions.

![Graph](image)
conditions by controlling the size of neighborhood.

6. Conclusion

In this paper, a neighborhood based training method is proposed. This training method can successfully improve the performance of a recognition system in mismatch conditions, at the cost of small accuracy degradation under a matching condition.

The experimental results show that a more robust model can be obtained without knowledge of the mismatch. While the gain reported is not large, it is on top of a fairly optimized result that uses the best algorithm and signal processing for the task. In addition, while the size of the neighborhood needs to be tuned, the result indicates that performance is not sensitive to its values, a larger value performs better under the higher mismatches and a smaller value favors more matching conditions.

We have shown that the robust neighborhood training can improve performance on top of the gains from variance normalization and ETSI front-end, it is highly probable that it can work in conjunction with other robustness techniques and other type of training criteria such as MCE or discriminative training.

7. Acknowledgement

This work is partially supported by RGC CERG number HKUST6210/03E and RGC central allocation grant number CA02/03.eg05.

8. References


