Maxium Likelihood Non-linear Transformation for Environment Adaptation in Speech Recognition Systems

M. Padmanabhan and S. Dharanipragada

IBM T. J. Watson Research Center P. O. Box 218, Yorktown Heights, NY 10598
(mukund,saty)@us.ibm.com

Abstract

In this paper, we describe an adaptation method for speech recognition systems that is based on a piecewise-linear approximation to a non-linear transformation of the feature space. The method extends a previously proposed non-linear transformation (NLT) technique by making the transformation function more sophisticated (piecewise-linear instead of piecewise-constant), and by computing the transformation to maximize the likelihood of the adaptation data given its transcription (instead of just matching the global statistics of the test and training data). This method also differs from other linear techniques (such as MLLR, linear feature space transforms, etc.) in two ways - first, the computed transformation is non-linear, second, the tying structure of the transformation depends not on the phonetic class but rather on the location in the feature space. Experimental results show that the method performs well for the case of limited adaptation data, and the performance gains appear to be additive to those provided by MLLR - yielding up to 3.4% relative improvement over MLLR.

1. Introduction

A real-world speech recognition system encounters several acoustic conditions in the course of its application. Currently, it is well known that a system trained only for a particular acoustic condition degrades drastically when it encounters a different acoustic condition. One method of improving recognition accuracy in new conditions during run-time is by identifying a transform that brings the new speech/feature vectors close to the speech/feature vectors seen during training – a technique often called “adaptation via feature space transformation.”

There are several types of transformations that have been considered. They broadly fall into the categories of “data transformation” methods, and “model transformation” methods. The methods described in [2, 3] fall in the former category, and the method described in [1] falls in the latter category. In [3], we proposed a new adaptation scheme, that transformed the features using a piecewise constant approximation to a non-linear transformation in such a way that the cumulative distribution function (cdf) of the transformed adaptation data is made identical to the cdf of the training data. This approach was motivated by the following reasoning:

1. the goal of adaptation is to match the pdf of the test and training data
2. it is well known that the assumed parametization of the true pdf of the data is often inaccurate; consequently, if we could deal with the empirically observed cdf of the data, there would be no need to make any modelling assumptions. Further, the cdf is a well behaved function (monotonic, lying between 0 and 1), consequently it is relatively easy to define a mapping of the feature dimension that equates two cdfs.

One of the distinguishing features of this method was that it relied on using the global statistics of the data and did not rely on any knowledge of the transcription of the adaptation data.

In this paper, we extend this approach to using a piecewise linear approximation of a non-linear transformation and computing the parameters of the transformation to maximize the likelihood of the adaptation data. We refer to this technique as maximum-likelihood non-linear transformation (MLNLT). The technique is especially suited to cases where only a limited amount of adaptation data is available (for example, 1 sentence). Further, it can also be used in conjunction with other linear transformation techniques such as MLLR, and is shown to provide statistically significant improvements over and above those provided by MLLR.

2. Mathematical Formulation

2.1. Background

Let \( p_{\text{adapt}}(x) \) represent the pdf of the adaptation data, \( p_{\text{train}}(y) \) represent the pdf of the training data, \( x = g(y) \) represent a non-linear transformation that maps the training data, \( y \), to \( x \), and \( p'_{\text{train}}(x) \) represent the pdf of the transformed training data. In [3], it was shown that the non-linear mapping, \( g() \), could be computed so as to minimize the Kullback-Liebler distance

\[
D(\cdot, \cdot) = \int_x p_{\text{adapt}}(x) \log \left[ \frac{p_{\text{adapt}}(x)}{p'_{\text{train}}(x)} \right] dx
\]

(1)

between \( p_{\text{adapt}}(x) \) and \( p'_{\text{train}}(x) \), where the quantity affected by the optimization is \( p_{\text{train}}(y) \). As the K-L distance between two distributions is minimized when they are identical, the strategy adopted in [3] was to equate the cdf’s corresponding to the densities \( p_{\text{adapt}}(x) \) and \( p'_{\text{train}}(x) \) at certain discrete points, thus leading to a piecewise constant mapping for \( g(y) \). This is shown as the solid line in Fig 1. A straightforward extension of this transformation function is to use a linear approximation between the discrete points. This would correspond to parametrizing the proposed mapping, \( x = g(y) \), as a piecewise-linear mapping (shown as a dotted line in Fig 1).

The above method may be viewed in the context of maximum likelihood as follows: if the term \( \int_x p_{\text{adapt}}(x) \log \left[ p_{\text{adapt}}(x) \right] dx \) in (1), which does not affect the optimization, is ignored it may be seen that minimizing the K-L distance in (1) is equivalent to computing the likelihood of the adaptation data using the pdf \( p'_{\text{train}}(x) \), and
maximizing the likelihood of the adaptation data, with respect to the parameters describing the non-linear transformation \( g(\cdot) \).

Unlike most maximum likelihood methods, the above method does not rely on supervision i.e., the actual transcription corresponding to the adaptation data is not required nor used. The method instead relies on the fact that the global statistics of the speech should be the same irrespective of what was spoken. This could be an incorrect assumption for instance in the case where the ratio of silence to speech is much higher in the adaptation data compared to the original training data. Consequently, there may be some advantage in using additional information that specifies the phonetic class that each frame belongs to.

The two main ideas that are pursued in this paper, and that differentiate it from [3] are:

1. the use of a piecewise-linear transformation
2. maximum likelihood estimation of the parameters of the transformation using a transcription of the adaptation data

For the one-dimensional case, we propose a parametrization of the form:

\[
x = g(y) = X_k + m_k (y - Y_k) \quad \forall x \in [X_k, X_{k+1}] \tag{2}
\]

i.e. the range of values of the transformed feature \( x \) (rather than the original feature \( y \)) is divided into intervals \([X_k, X_{k+1}]\), and the original features \( y \) whose transformed values lie in this range are transformed to \( x \).

With this parametrization, it becomes possible to derive a closed form expression for the log-likelihood of the adaptation data, and consequently use optimization routines to find a local optimum.

The pdf of the transformed training data is given by

\[
p_{trg}(x) = p_{trg}(y = g^{-1}(x)) \left/ m_k \right. \quad x \in [X_k, X_{k+1}] \tag{3}
\]

The log-likelihood of the adaptation data is now given by

\[
\sum_{x} \int_{x \in [X_k, X_{k+1}]} p_{trg}(x) \log[p_{trg}(g^{-1}(x)) \left/ m_k \right.] dx \tag{4}
\]

As the transformation \( g^{-1}(\cdot) \) in (4) is also a piecewise linear transformation, this is equivalent to transforming the adaptation data according to \( g^{-1}(\cdot) \) and computing the likelihood of the transformed feature using the pdf of the training data (scaled by \( m_k \)).

Defining \( n_k = \int_{x \in [X_k, X_{k+1}]} p_{trg}(g^{-1}(x)) dx \), this inverse mapping, \( y = g^{-1}(x) \), is given by

\[
y = Y_k + n_k (x - X_k) \quad \forall x \in [X_k, X_{k+1}] \tag{5}
\]

where

\[
Y_k = Y_0 + \sum_{j=0}^{l-1} n_j [X_{j+1} - X_j] \tag{6}
\]

2.2. Formulation

The above formulation applies to the case of one-dimensional feature vectors. For the multi-dimensional case, we will develop the formulation for the case where each dimension is transformed independently. Further, in computing the log-likelihood of the adaptation data, we will assume that an alignment of the adaptation data is available that specifies the context-dependent sub-phonetic state (henceforth referred to as a “leaf”) for each feature vector, and that the pdf of the multi-dimensional features corresponding to this leaf are modeled by a mixture of diagonal gaussians.

We will adopt the following notation:

\[
T \quad \text{number of adaptation feature vectors}
\]

\[
D \quad \text{dimensionality of feature vector}
\]

\[
D_t \quad t^{th} \text{ dimension of feature vector}
\]

\[
x_{t,d} \quad t^{th} \text{ feature vector in the adaptation data}
\]

\[
y_{t,d} \quad t^{th} \text{ transformed feature vector}
\]

\[
X_{d,k} \quad \text{bin boundaries for } D_t \text{ dimension of } X
\]

\[
Y_{d,k} \quad \text{bin corresponding to this leaf are modeled by a mixture of diagonal gaussians.}
\]

We will adopt the following notation:

\[
T \quad \text{number of adaptation feature vectors}
\]

\[
D \quad \text{dimensionality of feature vector}
\]

\[
D_t \quad t^{th} \text{ dimension of feature vector}
\]

\[
x_{t,d} \quad t^{th} \text{ feature vector in the adaptation data}
\]

\[
y_{t,d} \quad t^{th} \text{ transformed feature vector}
\]

\[
X_{d,k} \quad \text{bin boundaries for } D_t \text{ dimension of } X
\]

\[
Y_{d,k} \quad \text{bin corresponding to this leaf are modeled by a mixture of diagonal gaussians.}
\]

We will adopt the following notation:

\[
T \quad \text{number of adaptation feature vectors}
\]

\[
D \quad \text{dimensionality of feature vector}
\]

\[
D_t \quad t^{th} \text{ dimension of feature vector}
\]

\[
x_{t,d} \quad t^{th} \text{ feature vector in the adaptation data}
\]

\[
y_{t,d} \quad t^{th} \text{ transformed feature vector}
\]

\[
X_{d,k} \quad \text{bin boundaries for } D_t \text{ dimension of } X
\]

\[
Y_{d,k} \quad \text{bin corresponding to this leaf are modeled by a mixture of diagonal gaussians.}
\]

We will adopt the following notation:

\[
T \quad \text{number of adaptation feature vectors}
\]

\[
D \quad \text{dimensionality of feature vector}
\]

\[
D_t \quad t^{th} \text{ dimension of feature vector}
\]

\[
x_{t,d} \quad t^{th} \text{ feature vector in the adaptation data}
\]

\[
y_{t,d} \quad t^{th} \text{ transformed feature vector}
\]

\[
X_{d,k} \quad \text{bin boundaries for } D_t \text{ dimension of } X
\]

\[
Y_{d,k} \quad \text{bin corresponding to this leaf are modeled by a mixture of diagonal gaussians.}
\]

We will adopt the following notation:

\[
T \quad \text{number of adaptation feature vectors}
\]

\[
D \quad \text{dimensionality of feature vector}
\]

\[
D_t \quad t^{th} \text{ dimension of feature vector}
\]

\[
x_{t,d} \quad t^{th} \text{ feature vector in the adaptation data}
\]

\[
y_{t,d} \quad t^{th} \text{ transformed feature vector}
\]

\[
X_{d,k} \quad \text{bin boundaries for } D_t \text{ dimension of } X
\]

\[
Y_{d,k} \quad \text{bin corresponding to this leaf are modeled by a mixture of diagonal gaussians.}
\]

We will adopt the following notation:

\[
T \quad \text{number of adaptation feature vectors}
\]

\[
D \quad \text{dimensionality of feature vector}
\]

\[
D_t \quad t^{th} \text{ dimension of feature vector}
\]

\[
x_{t,d} \quad t^{th} \text{ feature vector in the adaptation data}
\]

\[
y_{t,d} \quad t^{th} \text{ transformed feature vector}
\]

\[
X_{d,k} \quad \text{bin boundaries for } D_t \text{ dimension of } X
\]

\[
Y_{d,k} \quad \text{bin corresponding to this leaf are modeled by a mixture of diagonal gaussians.}
\]

We will adopt the following notation:  

\[
T \quad \text{number of adaptation feature vectors}
\]

\[
D \quad \text{dimensionality of feature vector}
\]

\[
D_t \quad t^{th} \text{ dimension of feature vector}
\]

\[
x_{t,d} \quad t^{th} \text{ feature vector in the adaptation data}
\]

\[
y_{t,d} \quad t^{th} \text{ transformed feature vector}
\]

\[
X_{d,k} \quad \text{bin boundaries for } D_t \text{ dimension of } X
\]

\[
Y_{d,k} \quad \text{bin corresponding to this leaf are modeled by a mixture of diagonal gaussians.}
\]
we can write
\[
\frac{d L}{d n_{i,k}} (t) = - \sum_j \frac{g(t_{s,i}, t_{s,j})}{\sigma^2_{t_{s,i}, t_{s,j}}} \left[ X_{d, i} - X_{d, j} \right] \quad (8)
\]

\[
0 \leq k < B_{d}(t)
\]

\[
\frac{d L}{d n_{i,k}} (t) = - \sum_j \frac{g(t_{s,i}, t_{s,j})}{\sigma^2_{t_{s,i}, t_{s,j}}} \left[ X_{d, i} - X_{d, j} \right] \quad (9)
\]

\[
k = B_{d}(t)
\]

\[
\frac{d L}{d n_{i,k}} (t) = - \sum_j \frac{g(t_{s,i}, t_{s,j})}{\sigma^2_{t_{s,i}, t_{s,j}}} \left[ X_{d, i} - X_{d, j} \right] \quad (10)
\]

The expressions (7, 8, 9, 10) are then used by a numerical optimization package [5] to find a local maximum of the objective function, \(L\).

The intervals \([X_{d, i}, X_{d, i+1}]\) along each dimension define hyper-rectangles in the D-dimensional feature-space. Each hyper-rectangle has simple diagonal linear transform and bias (affine) associated with it. If \(N_{d}\) is the number of intervals in the \(d^{th}\) dimension, then the nonlinear transformation is composed of \(\prod_{d=1}^{D} N_{d}\) number of affine transforms. However, these transforms are all described by only \(D + \sum_{d=1}^{D} N_{d}\) parameters (the slopes at each interval, and the biases). This is achieved by sharing the parameters across different hyper-rectangles: all hyper-rectangles that originate from the same interval in a particular dimension, \(d\), share the same \(d^{th}\) diagonal value in the transform. This is an important difference between MLNLT and techniques such as MLLR. In MLLR, the transform, that is applied either to the models or the feature space, is associated the phonetic class identity whereas in MLNLT the transforms are associated with regions in the D-dimensional feature space. An important consequence of this form of association is that we do not need to explicitly take the Jacobian of the transform into account while computing the likelihoods during recognition [8]. The choice of hyper-rectangles and diagonal transforms was merely out of simplicity. Clearly, any form of partitioning of the feature space and structured or unstructured linear transforms can be used with this technique [9].

3. Experimental Results

We experimented with this technique to compensate for the mismatch between handset and speaker-phone telephone data. The baseline system was trained using handset data and the test data was speakerphone data.

3.1. System Description

All experiments were conducted on the IBM rank-based LVCSR system. The IBM LVCSR system uses context-dependent sub-phone classes which are identified by growing a decision tree using the training data and specifying the terminal nodes of the tree as the relevant instances of these classes [6, 7]. The training feature vectors are poured down this tree and the vectors that collect at each leaf are modeled by a mixture of Gaussian pdf’s, with diagonal covariance matrices. Each leaf of the decision tree is modeled by a 1-state Hidden Markov Model with a self loop and a forward transition. Output distributions on the state transitions are expressed in terms of the rank of the leaf instead of in terms of the feature vector and the mixture of Gaussian pdf’s modeling the training data at the leaf. The rank of a leaf is obtained by computing the log-likelihood of the acoustic vector using the model at each leaf, and then ranking the leaves on the basis of their log-likelihoods. The decision tree had 2347 leaves, and each leaf had 15 Gaussian mixture components to model the pdf.

Speech is coded into 25 ms frames, with a frame-shift of 10 ms. Each frame is represented by a 39 component vector consisting of 13 MFCCs and their first and second time derivatives. A baseline model was trained on several hours of the telephone handset data.

3.2. Experimental Set-up

The test data comprised of 25 sentences each (from the air travel domain) from 31 speakers, recorded on speakerphones. The adaptation data comprised of an additional set of up to 25 sentences from each speaker recorded on speakerphone. We experimented with using 1, 2, 5 and all the adaptation sentences in supervised adaptation mode, where the correct word script (and corresponding leaf level alignment) was assumed to be known for the adaptation data. The adaptation data was used to compute a feature space mapping (NLT or MLNLT), or to transform the means of the baseline model (MLLR) or to do both.

For the case of MLLR, the transforms were constrained to be block-diagonal with 3 blocks (corresponding to the cepstra, deltas and double-deltas) 1. Each transformation could be shared between models for several leaves with the tying structure and the transformations determined by a regression tree 2.

We also provide results with the NLT adaptation method described in [3]. This results in a warping function where the test feature space, \(x\) is partitioned such that \(X_{d,i}\) correspond to uniform sampling of the CDF of \(x_{d}\), and the training feature space \(y\) is partitioned such that \(Y_{d,i}\) correspond to uniform sampling of the CDF of \(y_{d}\). A plot of the warping function is shown in Fig 1 (solid line).

For the MLNLT experiments, we experimented with 2 different setups

1. the values of \(X_{d,i}\) in (5) were chosen to be the same

1 It was necessary to do this because for the 1 sentence adaptation case, use of a full transformation significantly degraded the performance after adaptation.

2 For the case of 1 sentence of adaptation data, only 1 transform was computed, and for the case of 25 sentences, approximately 10 transforms were computed.
<table>
<thead>
<tr>
<th>Method</th>
<th>Supervised</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Adaptn sent</td>
</tr>
<tr>
<td>Baseline</td>
<td></td>
</tr>
<tr>
<td>MLLR</td>
<td></td>
</tr>
<tr>
<td>NLT + MLLR</td>
<td></td>
</tr>
<tr>
<td>MLNLT-1</td>
<td></td>
</tr>
<tr>
<td>MLNLT-1+ MLLR</td>
<td></td>
</tr>
<tr>
<td>MLNLT-2</td>
<td></td>
</tr>
<tr>
<td>MLNLT-2+ MLLR</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Word Error Rate comparisons on a speakerphone test set.

as those provided by the NLT method, and the slopes $\alpha_{d,i}$ were initialized to the values computed from these intervals (MLNLT-1) - this corresponds to the top part of Fig 2.

2. the values of $X_{d,i}$ were chosen to lie uniformly between the minimum and maximum values of the feature dimension, and all slopes $\alpha_{d,i}$ were initialized to 1.0 (MLNLT-2) - this corresponds to the lower part of Fig 2.

The word error rate results are tabulated in Table 1. Based on the experimental results, we can draw the following conclusions:

1. Statistically significant gains can be obtained over MLLR by combining MLNLT-2 and MLLR. The gains are larger for smaller amounts of adaptation : 3.4% relative improvement (1 sentence adaptation) vs 1.7% relative improvement (25 sentence adaptation).

2. Interestingly enough, MLNLT-2 (uniform spacing of bin boundaries) performs better than MLNLT-1 (bin boundaries initialized from NLT, i.e. placed so as to have equal probability mass in each bin)

4. Conclusion

In this paper, we describe an adaptation method for speech recognition systems that is based on a piecewise-linear approximation to a non-linear transformation of the feature space. The formulation assumes that the transformation is applied on the training data, resulting in a corresponding transformation of the trained models. The transformation is then computed so as to minimize the K-L distance between the pdf’s of the adaptation data and the transformed training data for each context dependent sub-phonetic state (this turns out to be equivalent to maximizing the log-likelihood of the adaptation data). One of the unique features of the transformation is the tying structure associated with it - the transformations are tied not across phonetic classes, but rather on the location in the feature space. The experimental results show that the method complements MLLR for the case of limited adaptation data, and the performance gains appear to be additive to those provided by MLLR - yielding upto 3.4% relative improvement over MLLR for the case of supervised adaptation with only one sentence.

5. References