Abstract

In this paper, we propose a transform-based adaptation technique for robust speech recognition in unknown environments. It uses maximum likelihood spectral transform (MLST) algorithm with additive and convolutional noise parameters. Previously many adaptation algorithms have been proposed in the cepstral domain. Though the cepstral domain may be appropriate for the speech recognition, it is difficult to handle environmental noise directly in the cepstral domain. Therefore, our approach deals with such noise in the linear spectral domain in which speech is directly affected by the noise. As a result, we can use a small number of noise parameters for fast adaptation. The experiments evaluated on the FFMTIMIT corpus show promising result with only a small number of adaptation data.

1. Introduction

Recently, automatic speech recognition (ASR) systems using continuous density hidden Markov models (HMMs) have begun to show good performance. Although perfect speech recognition still seems to be a distant dream, improvement of a computing power and development of new algorithms have increased recognition accuracy. However, the tasks are still being limited in many ways such as using only clean speech, and all speakers having standard accent. Speech recognition in real environments is a difficult task, because the difference between the training and the testing conditions causes the accuracy of ASR systems to be degraded. To cope with this problem, efforts have been made to adapt ASR systems to real testing environments using only a small amount of adaptation data.

Previously, many environmental adaptation algorithms, such as the maximum a posteriori (MAP) based method[1], maximum likelihood linear transformation (MLLT)[2], and parallel model combination (PMC)[3] have been proposed. The MAP algorithm estimates new model parameters using a prior distribution of acoustic models. It is more robust than maximum likelihood estimation (MLE), but it adapts only the observed model parameters, and needs to have a relatively large amount of adaptation data. MLLT or maximum likelihood linear regression (MLLR)[4] is a transform-based algorithm, which adapts the mean vectors and covariance matrices. It requires a smaller amount of data than the MAP algorithm does, but it uses a relatively large number of transformation parameters. So, if too small an amount of data is used, the performance may be degraded. The PMC method combines clean speech models with environmental noise models, in order to construct real environment speech models.

It has the disadvantage of requiring noise samples to train the noise models.

In this paper, we propose the maximum likelihood spectral transform (MLST) as an adaptation algorithm in linear spectral space to handle unknown environments. Most adaptation algorithms work in a cepstral domain. A cepstral vector is an acoustic feature vector computed through a Fourier transformation and logarithm operation. Though it may be appropriate for speech recognition, it is difficult to analyze the additive and convolutional noise components directly from the cepstral vectors. The MLST attempts to control these noise components in the linear spectral domain, by using a transformation function with additive and convolutional noise parameters. The MLST is similar to the PMC in terms of the transformation domain and the use of additive and convolutional noises. Their difference lies in the type of transformation functions that are utilized. Since it requires only a small amount of data, the MLST has advantages over both the MLLT and the MAP. Therefore, the MLST is more useful for rapid adaptation.

This paper is organized as follows. The next section, describes a theoretic motivation of a MLST, new method of adaptation. The parameter estimation procedure for mean transform and feature transform are described in section 3. Then the efficiency of the algorithm is validated in section 4, where MLST based adaptation is evaluated empirically using far field microphone TIMIT (FFMTIMIT) data. Some conclusions of this approach are given in last section.

2. Maximum likelihood spectral transform

Environmental noise, such as which is produced by fans and raining sound, increases the amplitude of noisy speech. This noise is referred to as additive noise. Furthermore, the original speech may be distorted by the response characteristics of microphones and channels while it is being transmitted. This distortion is called convolutional noise. As a result of this noise, the frequency amplitude is multiplied by various coefficients, which correspond to the response characteristics. Therefore, we assume that the relationship between the noisy speech and the clean speech can be written as follows [5]:

$$\psi = a \odot \chi + b,$$  

(1)

where $\psi$ is a noisy speech linear spectrum picked up in a testing environment, $\chi$ is a clean speech linear spectrum obtained under laboratory conditions, $a$ is the convolutional noise of the transmitting channels, $b$ is the additive noise resulting from background sound.
For model space adaptation, therefore, a new noisy mean vector can be estimated using a diagonal matrix of convolutional noise parameters, $\lambda$, times a linear spectral mean vector of a clean acoustic model, $\mu$, plus an additive noise parameter vector, $b$. However, since the acoustic models are represented by cepstral mean vectors, domain transformation is required before applying the above analysis. Therefore, the cepstral mean vectors are first transformed to the corresponding linear spectral mean vectors. Firstly, inverse discrete cosine transformation (DCT) is performed and, then, an exponential operation is applied. In the cepstral domain, the noisy mean vector, $\mu$, is represented by the following equation:

$$
\mu = C \cdot \log(A\mu + b).
$$

where $C[n,m] = c_{n,m} = \cos(n(m+0.5)\pi / N)$.

The training process of the MLST algorithm proposed in this paper involves finding the values of the parameters, $A$ and $b$, that best model the given noise conditions.

3. Estimation of MLST parameters

For adaptation, the expectation maximization (EM) algorithm incorporating Baum’s auxiliary function [6], $Q(\lambda, \mathbf{X})$, is utilized for the MLE.

3.1. Mean transform

If a state of HMMs is modeled with a mixture of Gaussian distributions, to adapt the acoustic models the parameter estimation formula can be derived as follows;

$$
Q(\lambda, \mathbf{X}) = \phi - \frac{1}{2} P(\mathbf{O} | \lambda) \sum_{g=1}^{G} \sum_{t=1}^{T} y_{t}^{(g)}
\left[ \log(2\pi) + \log |\Sigma_{g}| + h(o^{(g)}, g) \right],
$$

where $\phi$ corresponds to the transition probabilities, which are assumed to have little effect, and $y_{t}^{(g)}$ is the posterior probability of Gaussian component $g$ at time $t$ given the current model and the training data. The noise parameters are estimated to maximize equation (5). So, the derivative of $Q(\lambda, \mathbf{X})$ with respect to $a_{k}$, which is the $k$-th diagonal element of $A$, can be derived as follows for the case of a diagonal covariance matrix:

$$
\frac{\partial Q(\lambda, \mathbf{X})}{\partial a_{k}} = -\frac{1}{2} P(\mathbf{O} | \lambda) \sum_{g=1}^{G} \sum_{t=1}^{T} y_{t}^{(g)} \left[ \log(a_{k} \beta_{a_{k},g} + b_{g}) - a_{k}^{(g)} \right]^{2}.
$$

We can obtain the maximum value of $Q(\lambda, \mathbf{X})$ by solving equation (8) for each noise parameter, $a_{k}$. Equation (8) can be rewritten as follows;

$$
\frac{\partial Q(\lambda, \mathbf{X})}{\partial a_{k}} = \sum_{g=1}^{G} \sum_{t=1}^{T} y_{t}^{(g)} \left[ a_{k} \beta_{a_{k},g} + b_{g} \right] - \sum_{g=1}^{G} \sum_{t=1}^{T} y_{t}^{(g)} a_{k} \beta_{a_{k},g}^{2} + b_{g}^{2}.
$$

A numerical iteration method, such as the bisection method, can be used to solve equation (9). The parameter estimation algorithm using the bisection method may require a large number of iterations. However, if we save and reuse $a_{n,k,t}$ and $\beta_{a_{n,k,t}}^{2}$ during the iterations, the computational cost can be greatly reduced.

3.2. Feature transform

In [7] the feature space transform uses the following expression for the probability of the transformed observation;

$$
L(o^{(g)};\mu_{g},\Sigma_{g}) = \frac{N(g(o^{(g)});\mu_{g},\Sigma_{g})}{|J(o^{(g)})|},
$$

where $L(\cdot)$ is a likelihood, $N(\cdot)$ is a normal distribution, $J(\cdot)$ is Jacobian matrix which is scaling matrix for transforming observation data, and $g(\cdot)$ is the transform function which maps each noisy speech observation to the corresponding clean speech observation, $\hat{o}^{(g)}$, as follows;

$$
g(o^{(g)}) = C \cdot \log(A \cdot \exp(C \cdot \hat{o}^{(g)})) - b = \hat{o}^{(g)}.
$$

Equation (12) describes that to estimate the likelihood of noisy speech observation given clean speech model, the normal distribution of clean speech obtained from equation (13) needs a Jacobian scaling factor for feature space transform. In practice, equation (13) is only needed to make Jacobian matrix, and equation (4) is efficient for derivation of target function. Then we use another $Q(\lambda, \mathbf{X})$ function including above equations for likelihood. The Jacobian matrix is given by;
which may be written as follows for each element;

\[
\frac{\partial \omega_i^{(r)}}{\partial \theta_j^{(s)}} = \frac{a_n \exp \left( \sum_{c=1}^{C} e_{c,n} \omega_i^{(r)} \right)}{a_n \exp \left( \sum_{j=1}^{J} e_{j,n} \theta_j^{(s)} \right) - b_n}
\]  

(15)

where Jacobian is an \( N \) by \( L \) matrix and can be simplified as a product of some matrices. For transform described in equation (4), the following equation must be optimized;

\[
Q(\lambda, \gamma) = \phi - \frac{1}{2} P(O | \lambda) \sum_{g \in R} \sum_{m=1}^{M} y_{gm}
\]

\[
- \left[ \text{log}(2\pi) + \log |\Sigma| + \log \left| J(\omega^{(r)}) \right|^2 + h(\omega^{(r)}, g) \right].
\]  

(16)

The derivative of \( Q(\lambda, \gamma) \) with respect to \( a_n \) for feature transform can be written as follow;

\[
\frac{\partial Q(\lambda, \gamma)}{\partial a_n} = - \frac{1}{2} P(O | \lambda) \sum_{g \in R} \sum_{m=1}^{M} y_{gm} \frac{\partial}{\partial a_n} \left[ \sum_{c=1}^{C} e_{c,n} \log \left( a_n \omega_i^{(r)} - b_n \right) - \mu_{c,s} \right] + 2 \sum_{m=1}^{M} \log \left( a_n \omega_i^{(r)} - b_n \right)
\]  

(18)

\[
= -P(O | \lambda) \sum_{g \in R} \sum_{m=1}^{M} y_{gm} \left[ \sum_{c=1}^{C} \frac{e_{c,n} \log \left( a_n \omega_i^{(r)} - b_n \right) - \mu_{c,s} \right] \frac{1}{\sigma_{c,s}^2} + \frac{2}{a_n \omega_i^{(r)} - b_n}
\]  

(19)

\[
= 0.
\]

Equation (19) can also use numerical iteration method and reduce the computation with caching.

4. Evaluation of MLST adaptation

4.1. Evaluation background

The MLST adaptation scheme was implemented into an automatic speech recognition system that uses Viterbi decoding algorithm. The noise parameters are estimated in a supervised manner, using labeled adaptation data. The baseline models are speaker-independent cross-word triphone 10-Gaussian tied state HMMs. The 3,696 training utterances from TIMIT data, which were recorded in a quiet environment, were used for training, and 1,296 testing utterances from FFMTIMIT data were used for testing. Firstly, we use one utterance for adaptation to check the effect of dynamic features. Secondly, for each speaker, we used 10 words from one sentence for the adaptation, and performed the recognition test on the remaining sentences of the same speaker to see the effect of the amount of adaptation data. When the training and testing environments were the same, the phone error rate (PER) was 26.4%~26.5%. When the testing was performed with FFMTIMIT data, however, the PER increased to 47%.

4.2. Mean transform of dimensionality

MLST adaptation can be performed with the static portion (13 dimensions) of the mean vectors. It can also be applied to the first and the second order derivatives (39 dimensions), which are referred to as “delta” and “accel”, respectively. The delta and accel are usually approximated using either simple linear difference (LD) operations or complex nonlinear differential (ND) functions. Table 1 shows that the PER for a 39-dimension MLST is lower than that of a 13-dimension MLST, since it contains more information. Systems using ND functions have a lower PER than those using LD operations. The reason for this may be the underlying difference in the baseline systems, i.e. the second column in the table. ND systems using a 13-dimension MLST show comparably good performance (29.0%), with only one-third of the computation time required by the best system (28.9%). For the next experiment, we use the ND system incorporating a 13-dimension MLST, which will be referred to as MLST.

Table 1: Comparison of 13-dimensional MLST and 39-dimensional MLST for 1 utterance adaptation

<table>
<thead>
<tr>
<th>Differential Functions</th>
<th>TIMIT</th>
<th>FFMTIMIT</th>
<th>13-D MLST</th>
<th>39-D MLST</th>
</tr>
</thead>
<tbody>
<tr>
<td>ND function</td>
<td>26.4</td>
<td>47.0</td>
<td>29.0</td>
<td>28.9</td>
</tr>
<tr>
<td>LD function</td>
<td>26.5</td>
<td>47.0</td>
<td>29.6</td>
<td>29.3</td>
</tr>
</tbody>
</table>

4.3. Word adaptation for mean transform

MLST shows better PER than MLLR (diagonal-matrix) and MLLR_F (full-matrix MLLR) when the number of adaptation utterances is small in Table 2. If a large amount of adaptation data is used, MLLR_F having large number of parameters is getting better. But it is not useful for fast adaptation. Therefore we focus on a small amount of data for word adaptation.

Table 2: Comparison of MLST, MLLR and MLLR_F for amount of adaptation utterances

<table>
<thead>
<tr>
<th>FFMTIMIT Uterance</th>
<th>MLST</th>
<th>MLLR</th>
<th>MLLR_F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>29.0</td>
<td>29.8</td>
<td>43.7</td>
</tr>
<tr>
<td>2</td>
<td>28.8</td>
<td>29.4</td>
<td>30.1</td>
</tr>
<tr>
<td>3</td>
<td>28.8</td>
<td>29.3</td>
<td>29.0</td>
</tr>
<tr>
<td>4</td>
<td>28.7</td>
<td>29.2</td>
<td>27.8</td>
</tr>
</tbody>
</table>

Figure 1 shows a comparison between MLST and MLLR. It can be seen in this figure that the MLST has a lower PER than the MLLR. As the number of words used for the adaptation is increased, the recognition performances of both
algorithms are improved, but the gap between them remains the same.

![Graph](image)

**Figure 1:** Average phone error rates (%) of MLST and MLLR for various amounts of adaptation data. TIMIT: TIMIT training and TIMIT testing, FFMTIMIT, MLLR, and MLST: TIMIT training and FFMTIMIT testing.

4.4. Word adaptation for feature transform

Feature adaptation is another approach to handle mismatch between the training and the testing. In above experiment MLST mean adaptation used the first 13-dimension of mean vectors. However, feature adaptation can generate full 39-dimension transformed feature vectors.

![Graph](image)

**Figure 2:** Average phone error rates (%) of MLST-FA and MLLR for various amounts of adaptation data. TIMIT: TIMIT training and TIMIT testing, FFMTIMIT, MLLR, and MLST-FA: TIMIT training and FFMTIMIT testing.

Figure 2 shows that MLST feature adaptation (MLST-FA) has lower PER than the MLLR when a small number of adaptation words are used. There is large difference between MLST-FA and MLLR in PER using only one word, though MLST-FA has poor recognition accuracy than the MLLR after five words. With regard to comparison with MLST mean transform, MLST-FA has better performance when one or two number of words is adapted.

5. Discussion

The transform-based adaptations, such as [2] and [4], are able to have many transform matrices with regression tree when there are a large amount of adaptation data. Therefore the MLST mean adaptation may also use them, and have reasonable performance when a large number of utterances are used. Feature transform cannot use regression tree. Otherwise, it can pursue non-linear noise adaptation described in [7]. If the MLST feature adaptation are performed repeatedly in short time before the testing, non-linear adaptation may be acquired.

6. Conclusions

In this paper, we proposed a novel fast speech adaptation algorithm using MLST. The one approach is MLST mean adaptation using clean spectral mean values of the model, and the other approach is MLST feature adaptation using noisy spectral values of the input data. Both approaches estimate additive and convolutional noise parameters to decrease the difference between the clean model and testing data. From the results of the adaptation experiments, we showed that the proposed algorithm is useful for handling different kinds of microphone data. Since MLST mean adaptation uses a small number of transformation parameters, it is especially suitable when only a small amount of adaptation data is available. Also since MLST feature adaptation can be processed in the preprocessing, it has efficient computation time.

7. References