ABSTRACT

Learning the influence of additive noise and channel distortions from training data is an effective approach for robust speech recognition. Most of the previous methods are based on maximum likelihood estimation criterion. In this paper, we propose a new method of discriminative learning environmental parameters, which is based on Minimum Classification Error (MCE) criterion. By using a simple classifier defined by ourselves and the Generalized Probabilistic Descent (GPD) algorithm, we iteratively learn environmental parameters. After getting the parameters, we estimate the clean speech features from the observed speech features and then use the estimation of the clean speech features to train or test the back-end HMM classifier. The best error rate reduction of 32.1% is obtained, tested on a Korean 18 isolated confusion words task, relative to conventional HMM system.

2. DISCRIMINATIVE LEARNING OF ENVIRONMENTAL PARAMETERS

2.1 Environmental Model

We assume that the clean speech signal is first passed through a channel distortions filter whose output is then corrupted by uncorrelated additive noise. If we use \( y_n[k], x_n[k], h_n[k] \) and \( n_n[k] \) to represent the mel-frequency log power spectrums of the observed speech, the clean speech, the channel distortions filter and the additive noise respectively, and \( k \) is a particular mel-frequency band, then we can get

\[
x_n[k] = y_n[k] - h_n[k] + \log(1 - \exp(n_n[k] - y_n[k]))
\]

Most of the current speech recognizer use the cepstral vectors as the features. When the environmental influence is observed in the cepstral domain, the relationship between cepstrums of the speech, the noise and the channel distortions is a rather complicated non-linear function as follow

\[
x = Y - h + C[\log(1 - \exp(C^{-1}(n - Y)))]
\]

where \( X, Y, n, \) and \( h \) are the cepstral vectors of the clean speech, the observed speech, the noise and the channel distortions filter, respectively, \( I \) is the unity vector, \( C \) and \( C^{-1} \) are the cosine transform matrix and inverse cosine transform matrix, respectively.

We assume that there are many kinds of additive noise in the environment, and we can use the weighted combination of multiple types of noise to represent \( n \)

\[
n = \sum_{\gamma} w_{\gamma} \cdot n_{\gamma}
\]
where $\mathbf{n}_i$ is the $v$th type noise, $w_i$ the weight of $\mathbf{n}_i$, $V$ the number of types.

In order to maintain the constraint of the weight, i.e. $\sum_{v=1}^{V} w_v = 1$, the following parameter transformation is adopted during parameters estimation

$$w_v = \frac{e^{\psi v}}{\sum_{v'=1}^{V} e^{\psi v'}}. \quad (4)$$

We also assume the channel distortions $h$ are consisted of the channel distortions of the whole training data and the channel distortions of the current utterance, and it can be implemented as follow

$$h = \alpha \cdot h_{n_{a}} + (1-\alpha) \cdot h_{n_{b}} \quad (5)$$

where $h_{n_{a}}$ and $h_{n_{b}}$ are the whole channel distortions and the current one, respectively, and $\alpha$ is an empirical constant.

In this case, the equation (2) can be rewritten as,

$$X = Y - (\alpha \cdot h_{n_{a}} + (1-\alpha) \cdot h_{n_{b}}) + C[\log(1 - \exp(C \cdot (\sum_{v=1}^{V} w_v \cdot n_v - Y)))] \quad (6)$$

If we use $\Phi$ to represent the environmental parameters $\mathbf{n}_v$, $w_v$, and $h_{n_{a, b}}$, then using the estimated $\Phi$ and $Y$, $X$ can be estimated.

### 2.2 Learning of Environmental Parameters

Let’s suppose we have a set of the observed speech cepstral training sequences $\{Y_1, Y_2, ..., Y_m\}$ and a set of classes $\{\lambda_1, \lambda_2, ..., \lambda_n\}$. We further assume that the all estimated clean speech signal $X_n$ that belong to $\lambda_i$ can be segmented into some equal parts and each same part follows a Gaussian distribution $N_{\mu_{i_{a}}, \Sigma_{i_{a}}}$ (for $a$-th segmentation of $X_n$, $a=1, 2, ..., A$), and the classifier we propose for learning environmental parameters models the estimated clean speech by the probability density function

$$p(X_n|\lambda_i) = \prod_{v=1}^{V} \frac{1}{\sqrt{2\pi\Sigma_{i_{a}}}} \exp\left(-\frac{1}{2}(X_{n,v} - \mu_{i_{a}})^\top \Sigma_{i_{a}}^{-1}(X_{n,v} - \mu_{i_{a}})\right) \quad (7)$$

where $d$ is the dimension of $X_n$, $\mu_{i_{a}}$ and $\Sigma_{i_{a}}$ can be calculated

$$\mu_{i_{a}} = E_{i_{a}}(X_{n,v}) \quad (8)$$

$$\Sigma_{i_{a}} = E_{i_{a}}((X_{n,v} - \mu_{i_{a}})(X_{n,v} - \mu_{i_{a}})^\top) \quad (9)$$

The goal of discriminative learning is to reduce the number of misclassifications through a minimization of the average loss function, the steps are as following.

1. **Discriminative function:** According to the model, the discriminative function is constructed as

$$g_i(X_n, \Phi) = \log p(X_n|\lambda_i) = \log(\prod_{v=1}^{V} p(X_{n,v} | \lambda_i)) = \sum_{v=1}^{V} \log p(X_{n,v} | \lambda_i)$$

$$= \sum_{v=1}^{V} \left(-\frac{d}{2} \log(2\pi) - \frac{1}{2} \log|\Sigma_{i_{a}}| - \frac{1}{2}(X_{n,v} - \mu_{i_{a}})^\top \Sigma_{i_{a}}^{-1}(X_{n,v} - \mu_{i_{a}})\right). \quad (10)$$

The implied decision rule for classification is defined as

$$X_n \in \lambda_i, \quad \text{if} \quad g_i(X_n, \Phi) = \max_j g_j(X_n, \Phi). \quad (11)$$

2. **Misclassification Measure:** Given a discriminative function, the misclassification measure is

$$d_i(X_n, \Phi) = -g_i(X_n, \Phi) + g_i(X_n, \Phi)$$

where $\lambda_i$ is the most confusable class. $d_i(X_n, \Phi) > 0$ implies misclassification and $d_i(X_n, \Phi) \leq 0$ means correct classification.

3. **Loss Function:** The lost function is defined as a sigmoid function of $d_i(X_n, \Phi)$

$$\xi(X_n, \Phi) = \frac{1}{1 + \exp(-d_i(X_n, \Phi))}. \quad (13)$$

4. **Average Loss Function:** For all the training data $X_n$ ($m=1, 2, ..., M$), the average lost function is defined as

$$L(\Phi) = \frac{1}{M} \sum_{n=1}^{M} \sum_{i=1}^{N} \xi(X_n, \Phi) I(X_n \in \lambda_i) \quad (14)$$

where $I(\ell) = \begin{cases} 1, & \text{if } \ell \text{ is true} \\ 0, & \text{otherwise} \end{cases}$

5. **Minimization:** The parameter $\Phi$ can be computed iteratively by minimizing the lost function $L(\Phi)$,

$$\Phi^{t+1} = \Phi^t - \eta(t) \nabla L(\Phi^t) \quad (15)$$

where $\Phi^t$ is the environmental parameter at the $t$th iteration, $\nabla L(\Phi^t)$ is the gradient of the average loss function. To control the convergence of the training procedure, we set the learning step size $\eta(t) = 1/(T_r + 2t)$, with $T_r$ being a prescribed large value ($=50$ in our case).

### 2.3 Gradient Calculation

The environmental parameters are adaptively adjusted to reduce the average loss function along a gradient descent direction. The gradient is obtained by computing the partial derivatives of $L(\Phi)$

$$\frac{\partial}{\partial \Phi} L(\Phi) = \frac{\partial}{\partial \Phi} \left( \frac{1}{M} \sum_{n=1}^{M} \sum_{i=1}^{N} \xi(X_n, \Phi) I(X_n \in \lambda_i) \right)$$

$$= \frac{1}{M} \sum_{n=1}^{M} \sum_{i=1}^{N} \frac{\partial \xi(X_n, \Phi)}{\partial \Phi} I(X_n \in \lambda_i) \quad (16)$$
where
\[ \frac{\partial \zeta_i(X_n;\phi)}{\partial \Phi} = \frac{\partial \zeta_i(X_n;\phi)}{\partial d_i(X_n;\phi)} \frac{\partial d_i(X_n;\phi)}{\partial \Phi} + \frac{\partial \zeta_i(X_n;\phi)}{\partial g_i(X_n;\phi)} \frac{\partial g_i(X_n;\phi)}{\partial \Phi} + \frac{\partial \zeta_i(X_n;\phi)}{\partial X_n} \frac{\partial X_n}{\partial \Phi}. \] (17)

The first factor in the right-hand-side of equation (17) can be simplified to
\[ \frac{\partial \zeta_i(X_n;\phi)}{\partial d_i(X_n;\phi)} = \frac{\partial}{\partial d_i(X_n;\phi)} \left( \psi(X_n;\phi)(1 - \zeta_i(X_n;\phi)) \right). \] (18)

The second factor of the right-hand-side of equation (17) can be simplified as follow
\[ \frac{\partial d_i(X_n;\phi)}{\partial g_j(X_n;\phi)} = \frac{\partial}{\partial g_j(X_n;\phi)} \left( -g_i(X_n;\phi) + g_j(X_n;\phi) \right) = \begin{cases} -1, & \text{if } j = i \\ 1, & \text{if } j = \Psi. \end{cases} \] (19)

The third factor of the right-hand-side of equation (17) can be modified to
\[ \frac{\partial g_j(X_n;\phi)}{\partial X_n} = \frac{1}{2} \left( \frac{\partial}{\partial X_n} \left( -\log(2\pi) - \frac{1}{2} \log|\sum_{\omega}^j| + \frac{1}{2} \sum_{\omega}^j \left( (X_n_\omega - \mu_\omega) \right)^2 \right) \right). \] (20)

For the whole channel distortions and noise, the fourth factors in equation (17) are
\[ \frac{\partial X_n}{\partial \Phi} = -\alpha \] (21)
\[ \frac{\partial h_{nh}}{\partial \Phi} = \gamma(w_c) \] (22)
\[ \frac{\partial n_i}{\partial \Phi} = \gamma(n_i) \] (23)

where
\[ \gamma(\Omega) = C \left( \frac{-\exp(C^{-1}(\sum \omega \cdot w_c \cdot n_i - Y_n))}{1 - \exp(C^{-1}(\sum \omega \cdot w_c \cdot n_i - Y_n))} \right). \] (24)

We first select the parameter \( \alpha \) in the equation (6) by using four segmentation for all \( X_n \) (A=4) and four types of noise (V=4), the results of ten iterations are shown in Fig. 1. It shows that \( \alpha = 0.3 \) exhibits an optimum, which is adopted in the following experiments. We find when \( \alpha = 1.0 \) the cepstral mean of the whole training data is used as the channel distortions and not consider the influence of the current utterance, it is not very good. And when \( \alpha = 0.0 \) the cepstral mean of the current speech is used as the channel distortions, which is very similar to CMS, the result is also not very good.

Table 1. Word error rates using various segmentation

<table>
<thead>
<tr>
<th>Segment</th>
<th>Two segment</th>
<th>Three segment</th>
<th>Four segment</th>
<th>Phoneme segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error rate</td>
<td>8.39%</td>
<td>9.04%</td>
<td>8.1%</td>
<td>7.73%</td>
</tr>
</tbody>
</table>

We then compare the results of equal segmentation for all words and the different segmentation for different words based on the number of phonemes, the word error rates for V=4 are listed in Table 1. It shows that the segmentation based on phoneme is good, which is used in the following experiments. We also evaluate the performances under the various combination types of noise, and Fig.2 shows the results. It is seen that the types of noise V=8 is enough for our database. Finally, we compare the proposed method EDFE (Environment Discriminative Feature Extraction) with CMS, our previous method RMFCC [11], and DFE, the results are listed in Table 2. It is seen that EDFE is better than CMS and RMFCC. Compared with DFE, EDFE gets near the same performance but needs almost half computational complexity. With respect to both the performance and the computational complexity, EDFE is the best one.
4. CONCLUSION

We have proposed a new DFE-like method, which learns the environmental parameters by using a simple classifier defined by ourselves and the MCE/GPD algorithm. After getting the parameters, we estimate the clean speech features from the observed speech features and then use the estimation of the clean speech features to train or test the back-end HMM classifier. Compared with DFE, the proposed method gets the environmental parameters by using a simple classifier and does not affect the back-end HMM classifier, thus it is simpler than the current DFE methods. In our experiments, the best error rate reduction of 32.1% is obtained, tested on a Korean 18 isolate confusion words task, relative to conventional HMM system.

5. REFERENCES