Feature Mapping and Recuperation by Using Elliptical Basis Function Networks for Robust Speaker Verification

LI Xin, ZHENG Yu, JIANG Fang-Ze

1. School of Electromechanical Engineering and Automation, Shanghai University, Shanghai 200072, China
2. School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China

Abstract The performance of speaker verification systems is often compromised under real-world environments. For example, variations in handset characteristics could cause severe performance degradation. This paper presents a novel method to overcome this problem by using a non-linear handset mapper. Under this method, a mapper is constructed by training an elliptical basis function network using distorted speech features as inputs and the corresponding clean features as the desired outputs. During feature recuperation, clean features are recovered by feeding the distorted features to the feature mapper. The recovered features are then presented to a speaker model as if they were derived from clean speech. Experimental evaluations based on 258 speakers of the TIMIT and NTIMIT corpuses suggest that the feature mappers improve the verification performance remarkably.

Keywords feature mapping and recuperation, elliptical basis function (EBF) networks, speaker verification.

1 Introduction

While today’s speaker verification systems perform reasonably well under controlled conditions, their performances are often compromised under real-world environments. In particular, variations in handset characteristics are known to be the major cause of performance degradation. Although this problem has been addressed by a number of approaches, such as cepstral weighting, adaptive component weighting, cepstral mean subtraction, relative spectral processing, and signal bias removal, most of them operate on the assumption that the channel effect can be approximated by a linear filter, which may be a poor approximation. Therefore, a more complex representation of handset characteristics is required. To this end, this paper investigates the non-linear characteristics of telephone handsets and proposes a handset mapper that overcomes the limitations of the conventional approaches.

2 Problems of Cepstral Mean Subtraction (CMS)

Cepstral mean normalization (CMN), a popular approach to channel mismatch compensation, is based on two assumptions: (1) the cepstral mean of clean speech is zero, (2) the channel is linear. These assumptions lead to a simple formulation of recovering the clean cepstrum from the channel distorted cepstrum.

\[ c_{\text{clean}} = c_{\text{distort}} - c_{\text{chan}} = c_{\text{distort}} - E[c_{\text{distort}}] \]  

where, \( c_{\text{chan}} = E[c_{\text{distort}}] \) represents the channel cepstrum and \( E[\cdot] \) denotes expectation. However, both assumptions are invalid in some applications. To avoid reliance on the first assumption, the differential-partial cepstral mean subtraction (DPCMS) in Ref. [6] were proposed. In DPCMS, the channel cepstrum is given by

\[ c_{\text{chan}} = E[c_{\text{distort}}] - E[c_{\text{clean}}] \]  

where the mean of the clean cepstrum \( E[c_{\text{clean}}] \) is the cepstral mean of the whole TIMIT (The Texas Instruments/Massachusetts Institute of Technology) corpus. This method, however, does not address the problem of channel non-linearity. In this paper, we propose a novel compensation method that does not rely on any of the above assumptions.

3 Non-linear Feature Mapping

Instead of relying on the linearity assumption as in CMS, We hypothesize that clean cepstra and distorted
cepstra are nonlinearly related, \( i.e., \)
\[
ce \text{clean} = f(c_{\text{distorted}}, c_{\text{clean}})
\]
where, \( f(\cdot) \) is a nonlinear function. In our study, we used elliptical basis function (EBF) networks\(^7\) to implement \( f(\cdot) \). An \( n \)-input, \( m \)-output EBF network can be considered as realizing a multi-dimensional non-linear mapper that maps data from \( R^n \) to \( R^m \). More specifically, the \( k \)-th output \( (k = 1, \ldots, K) \) of an EBF network transforms an input vector \( x \) into
\[
w_{k0} + \sum_{j=1}^{J} w_{kj} \exp \left\{ -\frac{1}{2 \gamma_j} (x - \mu_j)^T \Sigma_j^{-1} (x - \mu_j) \right\} = y_k(x) \approx f(x)
\]
where \( \mu_j \) and \( \Sigma_j \) are the mean vector and covariance matrix of the \( j \)-th basis function respectively, \( w_{k0} \) is a bias term, \( w_{kj} \) is an output weight connecting the hidden node \( j \) to the output node \( k \), and \( \gamma_j \) is a smoothing parameter that controls the spread of the \( j \)-th basis function. In our study, \( \gamma_j \) was determined heuristically by
\[
\gamma_j = a \sum_{i=1}^{L} \| \mu_i - \mu_j \| \quad j = 1, \ldots, J
\]
where \( \mu_i \) denotes the \( l \)-th nearest neighbor of \( \mu_j \) in the Euclidean sense, \( L \) is the number of nearest neighbors, and \( a \) is a parameter that controls the spread of the basis functions.

The mean vectors and the covariance matrices of an EBF-based feature mapper can be estimated in three steps\(^7\). In the first step, the \( K \)-means algorithm is applied to determine the cluster means of the training data \( \chi \) in the input feature space. Mathematically, we estimate the function center \( \mu_j \) by the sample average \( \bar{\mu}_j \), \( i.e., \)
\[
\mu_j \approx \bar{\mu}_j = \frac{1}{N_j} \sum_{x \in \chi_j} x
\]
where, \( x \in \chi_j \) if \( \| x - \bar{\mu}_j \| < \| x - \bar{\mu}_j \| \), \( \forall j \neq k, N_j \) is the number of samples in the cluster \( \chi_j \), and \( \| \cdot \| \) is the Euclidean norm. In the second step, the covariance matrices are approximated by the sample covariance.
\[
\Sigma_j \approx \hat{\Sigma}_j = \frac{1}{N_j} \sum_{x \in \chi_j} (x - \bar{\mu}_j)(x - \bar{\mu}_j)^T
\]
Finally, the output weights \(|w_{kj}|\) can be determined by a least squared method.

### 4 Recuperation of Gaussian clusters

We used two Gaussian clusters (with each cluster containing 1000 samples) to demonstrate the idea of feature recuperation. The first cluster had a mean vector at \((0.1, 0.0)\) with covariance matrix \([0.1, 0.4, 0.2, -0.2, -0.4, 1.5, 1.0, -0.5, 0, 0.5, 1.0, 1.5] \)
\]
\( \cdot \) Cluster 2 (desired output)
\]
\]
\( \cdot \) Recovered cluster
\]
Fig. 1 Data recovering by insufficient and sufficient centers