CEPSTRUM-BASED FILTER-BANK DESIGN USING DISCRIMINATIVE FEATURE EXTRACTION TRAINING AT VARIOUS LEVELS

Alain BIEM[1] *Shigeru KATAGIRI*[2]

[1]ATR Human Information Processing Research Laboratories
[2]ATR Interpreting Telecommunications Research Laboratories

ABSTRACT

This paper investigates the realization of optimal filter bank-based cepstral parameters. The framework is the Discriminative Feature Extraction method (DFE) which iteratively estimates the filter-bank parameters according to the errors that the system makes. Various parameters of the filter-bank, such as center frequency, bandwidth, gain are optimized using a string-level optimization and a frame-level optimization scheme. Application to vowel and noisy telephone speech recognition tasks shows that the DFE method realizes a more robust classifier by appropriate feature extraction.

1. INTRODUCTION

Cepstrum coefficients, either based on filter-bank or LPC model of speech, constitute the most widely used speech parameterization method. Cepstrum parameterization derives from homomorphic signal processing techniques, which provides a convenient way to separate the influence of the source from the vocal tract in the source/vocal tract model of the speech production. Thus, use of cepstrum coefficients ensures a good compactness of information by representing with few parameters the general aspect of the estimated speech spectrum. Also, cepstra produce decorrelated features without specific use of data statistics. This is particularly useful in the FFT-based estimation of the speech spectrum, where the original spectrum is first smoothed through a set of overlapping filters, which leads to a high degree of correlation among the components of the filter-bank output energies.

The matrix performing the transformation of the filter-bank output energies into cepstral parameters is chosen *a priori*. Consequently, for filter-bank-based cepstrum, performance depends on appropriate design of the filter-bank. Most filter-bank based cepstrum applications have relied on the perception-based Mel scale (e.g. MFCCs). However, the relation between perceptually-motivated feature extraction and statistical pattern recognition remains unclear. Perceptually-motivated cepstral parameters may not be the optimal features within the framework of statistical speech-pattern recognition.

In previous work [1][2], we proposed the Discriminative Feature Extraction (DFE) method as a way to efficiently design a recognizer structure, in which the feature extractor is consistent with the error minimization at the back-end classification process. Other studies have shown that DFE is capable of improving speech recognition performance [3]. Here, we extend DFE application to filter-bank-based cepstral coefficients.

The study has been done in two steps. First, a vowel fragment recognition task was carried out with the motivation of analyzing the way DFE-optimized cepstrum performs feature extraction, given that vowel characteristics are rather well known.

Secondly, the method is applied to a more practical task which consists of recognizing names over the telephone. Since cepstrum parameterization is performed sequentially in time, we have investigated the link between frame-based DFE-optimized filter-bank and string accuracy by comparing a string-level optimization of cepstral parameters to a frame-level optimization.

2. DFE-BASED CEPSTRUM REPRESENTATION DESIGN

2.1. Filter-bank-based cepstrum

In filter-bank modeling of the speech spectrum, cepstrum coefficients are a linear transformation of the filter bank outputs. Here, a filter-bank is simulated in the frequency domain by weighting of DFT bins with the magnitude frequency response of the filter. Let X be a sequence of speech vectors $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_T\}$ in which $\mathbf{x}_t = [x_{t,1}, \ldots, x_{t,f}, \ldots, x_{t,F}]^T$ is the power spectrum of the frame (short time window position); $x_{t,f}$ represents the f-th element of the spectral-vector; F is the maximum frequency index. An N-channel filter-bank model transforms each $\mathbf{x}_t$ into a lower dimensional vector $\mathbf{y}_t = [y_{t,1}, \ldots, y_{t,n}, \ldots, y_{t,N}]^T$ such that an output feature $y_{t,n}$ is the windowed log energy of the n-th channel:

$$y_{t,n} = \log_{10}\left(\sum_{f \in B_n} \theta_n(f) x_{t,f}\right), \quad \text{for } n = 1, \ldots, N, \qquad (1)$$

where B_n represents the channel interval and $\theta_n(f)$ the weighting at frequency f provided the n-th filter.

From the vector of log energies, the cepstrum vector $\mathbf{c}_t = [c_{t,1}, \ldots, c_{t,i}, \ldots, c_{t,L}]^T$ is computed via a discrete cosine transform:

$$c_{t,i} = \sum_{n=1}^{N} y_{t,n} \cos\left(\frac{i\pi}{N}(n - 0.5)\right), \qquad (2)$$

for $i = 1, \ldots, L$, where L is the number of cepstral coefficients.

2.2. DFE-based design

DFE-based cepstrum design is optimizing various parameters of the filter-bank while using a cepstral distance measure in the classification process. DFE uses the Minimum Classification Error/Generalized Probabilistic Descent method (MCE/GPD) formalism of discriminative training for optimizing the overall recognizer (filter-bank and classifier structure) for the single target of minimizing the error at the back-end classification process. If Λ denotes the set of parameters of the classifier and Θ the parameter set of the filter-bank, the parameter set of the overall recognizer

$\Phi = \{\Theta, \Lambda\}$ is adaptively updated after presentation of each pattern X aiming at minimizing a smooth error count measure.

For commodity of gradient-based optimization, the magnitude response $\theta_n(f)$ of the n-th filter is of a Gaussian-form:

$$\theta_n(f) = \varphi_n \exp\left(-\beta_n \left(p(\gamma_n) - p(f)\right)^2\right), \quad (3)$$

for $n = 1, \ldots, N$, where the trainable parameters $\beta_n > 0$ and γ_n determine bandwidth and center frequency, and φ_n is the trainable "gain" parameter in the n-th channel. $p(f)$ maps the linear frequency f onto the perceptual representation. For instance, a Mel scale mapping will provide Mel cepstral coefficients (MFCCs). The filter-bank parameters are composed of the set of center frequencies, bandwidths and gains, i.e, $\Theta = \{\phi_n = \{\gamma_n, \beta_n, \varphi_n\}\}$ or the set weights, i.e, $\Theta = \{\phi_n = \{\theta_n(f)\}\}$, for $n \in \{1, \cdots, N\}$ and $f \in \{1, \cdots, F\}$. The cepstrum generating process can be modeled as a transformation $\mathbf{c}_t = \mathcal{F}_\Theta(\mathbf{x}_t)$.

DFE-optimized cepstrum coefficients (DFCC) are designed by appropriate optimization of center frequency, bandwidth and gain aiming at minimum error. Thus, center frequency-optimized cepstrum coefficients (C-DFCC), bandwidth-optimized DFCC (B-DFCC), gain-optimized DFCC (G-DFCC) and independent weighting optimization (W-DFCC) could be designed by such a method. Here the term "weighting" refers to optimizing each frequency weight without keeping the Gaussian constraint. For generating a globally efficient model, a simultaneous optimization of center frequency, bandwidth and gain (S-DFCC) could be carried out.

3. IMPLEMENTATION ON A PROTOTYPE-BASED CLASSIFIER

3.1. Recognizer structure

The recognizer used in the following is the Prototype-Based Minimum Error Classifier (PBMEC) structure described in [4] but adapted to handle the DFE optimization process. It is a finite state machine, similar to a Hidden Markov Model (HMM) but with use of Lp-norm of distances instead of probabilities, which embeds a Dynamic Programming (DP) procedure to provide the final score of an input pattern across phonetic models. Thus, the classifier could be thought of as an HMM, using Viterbi decoding. The technical merit is that any distance measure that is consistent with the chosen speech parameterization method can used.

Concretely, we are given a finite set of P phonetic models, i.e,

$$\Lambda = \{\lambda_1, \ldots, \lambda_j, \ldots, \lambda_P\}, \quad 1 \le j \le P \quad (4)$$

where λ_j is composed of a set of prototypes distributed among the states of the model:

$$\lambda_j = \{\mathbf{r}_{j,s,m}\} \begin{array}{l} 1 \le s \le S \\ 1 \le m \le M. \end{array} \quad (5)$$

$\mathbf{r}_{j,s,m}$ represents the m-th prototype vector of the s-th state of model λ_j and $r_{j,s,m,i}$ is the i-th component of $\mathbf{r}_{j,s,m}$. S is the total number of states. The number of reference vector per state is M.

The distances between an input spectral frame-vector $\mathbf{x}$ to state s of category j is an Lp-norm of distances defined as

$$D_{j,s}(\mathbf{x}; \Phi) = \left\{ \sum_{m=1}^{M} \sigma(\mathbf{c} = \mathcal{F}_\Theta(\mathbf{x}), \mathbf{r}_{j,s,m})^{-\nu} \right\}^{-\frac{1}{\nu}} \quad (6)$$

where $\sigma(\mathbf{c}, \mathbf{r}_{j,s,m}) = (\mathbf{c} - \mathbf{r}_{j,s,m})(\mathbf{c} - \mathbf{r}_{j,s,m})^T$ is the Euclidean distance between $\mathbf{c}$ and $\mathbf{r}_{j,s,m}$. $\mathbf{c}$ is the cepstral representation of $\mathbf{x}$ and ν is a positive constant.

The discriminant function $g_k(X; \Phi)$ for each string category k is the sum of states-distances along the best DP paths for that category:

$$g_k(X; \Phi) = \sum_{t=1}^{\mathcal{T}} D_{j_t^k, s_t^k}(\mathbf{x}_t; \Phi), \quad (7)$$

where j_t^k is the current phonetic model at time t and s_t^k is the current state at time t along the best DP path of the string category k.

3.2. DFE training

For X belonging to category k, the discriminative ability of the recognizer is estimated by the use of a misclassification measure $d_k(X; \Phi)$ which emulates the classification decision in scalar values: a positive value means a misclassification and a negative value implies correct classification. The loss (cost) of the decision of assigning X to category C_k, denoted by $\ell(X, \Phi) = \ell(d_k(X; \Phi))$, is a smooth approximation of the minimum error cost function (0-1 cost function) such as a sigmoid.

The target in the DFE paradigm is to find the optimal values of both Θ and Λ minimizing the expected loss $\mathcal{L}(\Phi) = E_X[\ell(X, \Phi)]$ which is closely related to the error rate achieved by the system.

Given a training token X, belonging to a known category (words/phoneme/sentence), the adaptation rule for the classifier parameters is

$$r_{j,s,m,i}[\tau + 1] = r_{j,s,m,i}[\tau] - \epsilon_t \mathbf{U1} \frac{\partial \ell(X, \Phi)}{\partial r_{j,s,m,i}}. \quad (8)$$

We have a similar adaptation rule for filter-bank parameters:

$$\phi_n[\tau + 1] = \phi_n[\tau] - \rho_\tau \mathbf{U2} \frac{\partial \ell(X, \Phi)}{\partial \phi_n}. \quad (9)$$

ϵ_τ and ρ_τ are small positive numbers, representing the classifier learning rate and the feature extractor learning rate, respectively. **U1** and **U2** are positive definite matrices. In practice, the adjustment rule in (9) is done through a logarithmic transformation of each parameter to keep the filter-bank parameters positive.

An important issue is the level at which training should be performed (string-level or frame-level), especially when labeling information is not available. A string-level training will ensure that the correct string must display the smallest accumulated distance. String-level training is the most used method of optimization since it is closely related to the target task (word/sentence recognition).

From the distance measure defined in (6), it is obvious that a filter-bank shall produce cepstrum values that are close to the corresponding local prototype vectors, given the current frame. Thus, we have investigated a frame-level optimization of the overall recognizer along the line given in [5]. That is, the filter-bank is optimized for each frame that causes a deviation from the correct path.

3.2.1. String-level optimization

For a pattern X of category C, the misclassification measure $d_C(X; \Phi)$ which reflects the overall string error is defined as

$$d_C(X; \Phi) = 1 - \frac{g_W(X; \Phi)}{g_C(X; \Phi)}. \quad (10)$$

$\mathcal{W}$ refers to the best incorrect category. The gradient of the string-level loss is given by

$$\frac{\partial \ell(X,\Phi)}{\partial \phi_n} = \tag{11}$$

$$\ell'(d_C(X;\Phi)) \sum_{k=C,k=\mathcal{W}} \sum_{t=1}^{\mathcal{T}} \frac{\partial d_C(X;\Phi)}{\partial D_{j_t^k,s_t^k}(\mathbf{x}_t;\Phi)} \frac{\partial D_{j_t^k,s_t^k}(\mathbf{x}_t;\Phi)}{\partial \phi_n}$$

From (11), it can be seen that all phonetic models belonging to the correct path and the ones belonging to the incorrect path are updated at each data presentation. Consequently, acoustic models that belong to both paths are updated twice.i.e, within the correct path and within the incorrect path. This is likely to complicate the filter-bank optimization scheme since a frame is supposed to belong to a specific acoustic model.

3.2.2. Frame-level optimization

In the frame level optimization [5], the loss is the sum of a local frame-based losses:

$$\ell(d_C(X;\Phi)) = \sum_{t=1}^{\mathcal{T}} \ell(\delta_C^{\mathcal{W}}(\mathbf{x}_t;\Phi))) \tag{12}$$

where

$$\delta_C^{\mathcal{W}}(\mathbf{x}_t;\Phi)) = 1 - \frac{D_{j_t^{\mathcal{W}},s_t^{\mathcal{W}}}(\mathbf{x}_t;\Phi)}{D_{j_t^C,s_t^C}(\mathbf{x}_t;\Phi)} \tag{13}$$

represents the local misclassification of the t-th frame. It can be seen from (12) that only phonetic models belonging to different paths, given a frame, are updated. Consequently, the filter-bank parameters are updated according to those frames that have produced the mismatch between the two paths.

4. PRELIMINARY STUDY ON A VOWEL FRAGMENT RECOGNITION TASK

Vowel recognition provides a tractable framework for analysis of DFCC in a simple task, since the spectral characteristics of vowels are well known. The framework is recognizing the 5-class Japanese vowels. A database of 500 sentences spoken by 5 speakers (3 males and 2 females) was used to extract 1750 tokens for training and 1750 token for testing. The training body and the testing body was balanced among the speakers and the vowels.

The speech signal was digitized at 12kHz and stored at 16 bits. A Hamming window of 21 ms was used to extract the center-frame of each vowel, given labeling information.

Twenty channels initially aligned in the Mel scale were used to produce 10 cepstral coefficients. Gain values were initially set to one. The K-means algorithm was performed to design the PBMEC, prior to MCE/GPD or DFE training. MCE/GPD training was carried out as baseline for MFCC testing. DFE training produced the various DFCCs. Note that before DFE training, the initial configuration is similar to MFCC. DFCC was produced in a segment classification basis, using only 1 state of PBMEC with 1 prototype per vowel.

Fig. 1 shows the resulting filter-bank as well as the error rates for the corresponding feature types. As expected from a vowel recognition ask, all MCE-based systems achieved relatively similar results but higher than the maximum-likelihood (21.0% for training and 22.3% for testing).

The point of interest here is to analyze the resulting filter-bank model. Vowels are mainly characterized by their formant values. Consequently, a histogram of formant frequencies as contained in the database is shown in the top of the figure. The formants were computed using an LPC-based root finding method followed by human verification.

When spacing adjustment is involved (C-DFCC and S-DFCC), most filters gather in the F2 and F3 regions. In the C-DFCC task, filters of the lower frequency region, has gathered around 3 specific regions, which correspond to regions spanned by the first formant (around 0.5 kHz), the sec-

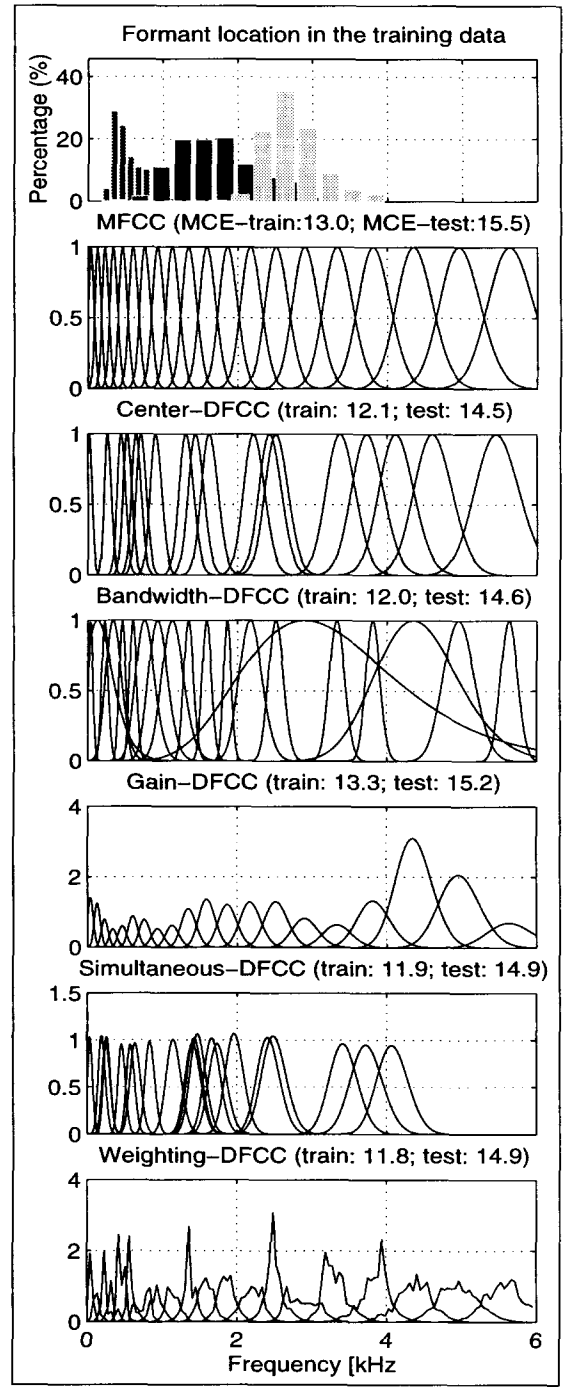

Figure 1. Optimized filter-bank in the vowel recognition task. Number between parentheses are the error rates for the corresponding cepstral features.

ond formant (around 1.5 kHz) and third formant (around 2.5kHz). C-DFCC also displays the best performance on testing data. B-DFCC tends to focus on the F3 region which is noticeable by the increase in bandwidth of the filter in this specific region. The bandwidth values are not anymore a monotonic function of center frequencies. G-DFCC put an emphasis on medium and higher frequency regions. W-DFCC did not generalized well, given the best result achieved in the training data. This may be due to the high number of parameters involved.

It seems that in most cases, the filter-bank puts emphasis on the most spectrally meaningful regions.

5. NAMES RECOGNITION TASK

The method was tested in the design of a system which recognizes Japanese's names and forwards calls to staff members within the ATR laboratory.

Data were automatically collected in office environment by a system that periodically called staff members to repeat 5 randomly selected names. Each name utterance was therefore spoken in isolated word recognition mode. The process resulted into 684 utterances in total from 47 speakers (3/4 of them are male) with the target of recognizing 64 names; We used 570 utterances as training and 114 as a closed speaker testing set (c-test). The c-test features the same speakers as in the training data as well the same vocabulary. Also, 234 utterances were collected during a demonstration of the system, which involved new names and new speakers that were not contained in the design set. Those utterances constitute the open test set (o-test).

5.1. Experimental settings

The speech signal, coming from the telephone transmitter, was digitized at 8 kHz sampling rate and at 16 bits. A Hamming window of 21 ms was shifted every 5 ms over an input speech utterance, thus producing 128 FFT-based power spectrum ($F = 128$), as input to the filter-bank. The same feature extraction framework as in the vowel recognition was used with the filter-bank spanning the 0-4 kHz frequency range.

For classification, we used 26 context-independent phoneme models, which correspond to 5 Japanese vowels, 20 consonants and silence. A finite state grammar was used to constraind the search.

5.2. Results

The initial prototypes of the classifier module produced by ML-based segmental K-means provided an estimated segmentation of each utterance. This ML-produced baseline system was further trained by one of the five types of DFE training. For comparison purposes, we also ran classical MCE/GPD training using MFCC. The results for string-level and frame level training are shown in Table 1.

features	String-level			Frame-level		
	train	c-test	o-test	train	c-test	o-test
MFCC (ML)	67.0	54.3	35.9	-	-	-
MFCC (MCE)	97.3	91.2	57.9	97.3	94.7	59.9
C-DFCC	96.9	92.9	60.0	98.2	92.9	59.5
B-DFCC	97.0	92.9	57.8	98.5	94.7	56.8
G-DFCC	97.5	94.7	59.0	99.1	94.7	60.3
S-DFCC	96.4	92.9	55.3	98.7	95.6	64.5
W-DFCC	97.8	92.9	61.6	98.8	93.8	56.9

Table 1. Experimental results of ATR names recognition task using string-level training and frame-level training for various cepstral features.

The results (in terms of recognition rates), show that all MCE-based trainings outperform ML training in both testing sets. The frame-based optimization shows better performance in average than string-level training for both MCE (only classifier adjustment) and DFE (joint optimization), on the closed test set.

For string-level training, the best performance on the closed test set is realized by G-DFCC and on the open test set by W-DFCC (61.6% compared to 57.8% for MFCC). Also, C-DFCC appears to be relatively robust considering its relative performance across the two testing sets.

For frame-level optimization, the best result is achieved by S-DFCC on both testing sets. In particular, the result in the open test set is far ahead of other cepstral features (64.5%, compared to 59.9% for MFCC). This is in contrast to its rather limited performance achieved within the string-level optimization.

For closed speaker test set, MFCC and DFCC provide relatively close performance for both levels of training, although in most cases, DFCCs display better recognition rates. For open speaker/vocabulary test set, DFCCs appear to be more robust than MFCC for both level of training. However, the best type of DFCC depends on both the task and the level of training.

6. CONCLUSION

The Discriminative Feature Extraction application for cepstrum optimization was formalized. Cepstrum optimization consisted in adjusting various filter-bank parameters such as center frequency, bandwidth, gain and weighting within a cepstral distance measure. The system was first applied to vowel fragment recognition task, where it was shown that the filters tend to move towards formant regions after DFE training. Secondly, the system was tested in a telephone-based names recognition task. In this framework, a frame-level training and a string-level training was investigated. DFE and MCE training seem more efficient in average when training at the frame-level for test data close to design data. Using an open data test set, the best result was achieved by simultaneously adjusting center frequency, bandwidth and gain using the frame-level optimization.

7. ACKNOWLEDGMENT

The authors would like to specially thank Mr Erik McDermott for insight helpful comments and for providing the PBMEC software. Thank also to Mr Rick Woudenberg for collecting the ATR names data.

REFERENCES

[1] S. Katagiri, B.-H. Juang, and A. Biem. Discriminative feature extraction. In R. J. Mammone, editor, *Artificial Neural Networks For Speech and Vision.* Chapman and Hall, 1993.

[2] A. Biem, E. McDermott, and S. Katagiri. Discriminative filter bank model for speech recognition. In *Eurospeech 95*, volume 1, pages 545–548, Oct 1995.

[3] C. Rathinavelu and L. Deng. HMM-based speech recognition using state-dependent, linear transforms on mel-warped dft features. In ICASSP'96, editor, *Proc. of ICASSP'96*, volume 1, pages 9–12, May 1996.

[4] E. McDermott and S. Katagiri. Prototype-based minimum classification error/generalized probabilistic descent for various speech units. *Computer Speech and Language*, 8(8):351–368, Oct. 1994.

[5] J.-K. Chen and F. Soong. An N-best candidates-based discriminative training for speech recognition applications. *IEEE Transactions on Speech and Audio Processing*, Jan 1994.