OPTIMIZING FEATURES AND MODELS USING THE MINIMUM CLASSIFICATION ERROR CRITERION

Alain Biem

IBM T. J. Watson Research Center,
P. O. Box 218, Yorktown Heights, NY 10598, USA
biem@us.ibm.com

ABSTRACT

Discriminative Feature Extraction (DFE) has been proposed as a extension of MCE/GPD for the joint optimization of features and models. This study presents various configurations of this discriminative framework aimed at optimizing filter-bank parameters, using cepstrum and delta cepstrum as features, within an HMM-based system. Features and models are optimized either jointly or separately. Experimental results on the ISOLET database show that the joint optimization of features and models realizes the best performance: more than 13% absolute error rate reduction on the E-set task compared to an MLE-trained system using MFCCs and more than 1.85% absolute error rate reduction compared to an MCE-trained system using MFCCs.

1. INTRODUCTION

A speech recognizer is primarily composed of two modules: a feature extraction module, which maps the input signal into a form suitable for recognition by removing noisy components and enhancing relevant-to-recognition characteristics, and a modeling module, which relies on the statistics of the feature space to create models. Although these two modules constitute an essential part of any recognition system, most systems do not fully integrate them. The feature extraction process and the modeling process are designed separately, each process using its own optimization criterion, meaning that the overall system is sub-optimal.

In previous work [1, 2], we proposed Discriminative Feature Extraction (DFE) as a framework for the joint optimization of features and models using the Minimum Classification Error (MCE) criterion. DFE applies discriminative training to an integrated recognition system, where a unique criterion is used to optimize both modules simultaneously. However, integrated optimization and discrimination of the front-end and the back-end of the system are two separate paradigms that can be applied independently of each other. Integrated optimization, that is, the joint optimization of the front-end system and the back-end system, can be done using any optimizable criterion, and discriminative training can be applied selectively on the front-end system or the back-end system. It has been argued that the integrated optimization scheme may have difficulties in converging while selective optimization, where DFE training is applied first to optimize the front-end and then the models in turn, is more stable in convergence. Indeed since the introduction of the DFE method in [1], various implementations of the algorithm have been carried out. Rathivanelu and Deng [3], Rahim and Lee [4] use DFE on an Hidden Markov Model (HMM)-based integrated system that optimizes feature transformations; De La Torre *et al.* [5] uses DFE to generate feature transformation in a pre-training stage using simple models; B. Mak *et al.* [6] uses DFE to optimize auditory filters and HMMs separately. Although, all these DFE variants have shown great improvement on performance within their relative context, it remains unclear how these DFE paradigms compare to each other.

In this paper, we carried out an exhaustive study of various DFE configurations on the ISOLET database using HMM as data modeling and filter-bank based cepstrum as features. Similar to our previous work in [2], DFE is aimed at optimizing center frequencies, bandwidths, and gains of a filter-bank. In this study, we extend the previous work by using a continuous HMM and embedding dynamic cepstrum in the DFE optimization process on a publicly-available database.

2. DISCRIMINATIVE FEATURE EXTRACTION FOR ISOLATED WORD RECOGNITION

DFE is an extension of MCE [7] that embeds the feature extractor's parameters within the optimizable parameters of the overall recognizer. Let Λ denotes the parameter set of all models and Θ the parameter set of the feature extraction module. The parameter set of the overall recognizer is referred to as $\Phi = \{\Theta, \Lambda\}$.

Given an input signal S, recorded prior to feature extraction, the discriminative function $g_k(S;\Phi)$ of word W_k is defined as the log likelihood of the Viterbi path. The misclassification measure $d_k(S;\Phi)$, which is positive for correct recognition and negative otherwise, is defined as $d_k(S;\Phi) = -g_k(S;\Phi) + \overline{g}_k(S;\Phi)$ where $\overline{g}_k(S;\Phi)$, the anti-discrimination function, is defined as $\overline{g}_k(S;\Phi) = \log\{\frac{1}{M-1}\sum_{j\neq k}^{M} e^{g_j(S;\Lambda)\eta}\}^{\frac{1}{\eta}}$ with a positive η and M being the number of words in the lexicon. The objective function to be minimized is the expected loss $\mathcal{L}(\Phi) = E_s[\ell(S;\Phi)]$ where the loss $\ell(S;\Phi) = \ell(d_k(S;\Phi))$ is a smooth approximation of the 0-1 cost function, and is typically a sigmoid. The Generalized Probabilistic Descent (GPD) update is applied at each iteration after presentation of each pattern S as

$$\Lambda_{\tau+1} = \Lambda_\tau - \epsilon_\tau \mathbf{U1} \frac{\partial \ell(S;\Phi)}{\partial \Lambda}\bigg|_{\Lambda=\Lambda_\tau} \quad (1)$$

$$\Theta_{\tau+1} = \Theta_\tau - \rho_\tau \mathbf{U2} \frac{\partial \ell(S;\Phi)}{\partial \Theta}\bigg|_{\Theta=\Theta_\tau} \quad (2)$$

where $\mathbf{U1}$ and $\mathbf{U2}$ are positive definite matrices; τ is the training time index; ϵ_τ and ρ_t are the models' and feature extraction module's learning rates; and Λ_τ and Θ_τ indicate the models' and the feature extractor's parameter status at training time τ.

The chain rule of differential calculus is used to adjust the feature extraction module. As argued in [2], the use of the above modular GPD framework enables one to deal with the instability within the DFE training process that may occur due to differences in the type of parameters between the feature extraction module and the models. This is especially the case when DFE is applied to a low-end feature extraction module as in [2] [6]. For $\rho_t = 0$ training is equivalent to classical MCE and for $\epsilon_t = 0$ training optimizes the front-end while models are unchanged.

3. HMM-BASED DFE OPTIMIZATION OF FILTER-BANK PARAMETERS

Here, we only describe derivatives for filter-bank parameters. HMM derivatives can be found in [8] and [9].

3.1. Cepstrum-based Filter-bank modelling of speech

The cepstrum coefficients are computed at the output of the filter-bank, which is simulated in the DFT domain by weighting of the DFT bins with the magnitude frequency response of the filter. For a sequence of speech spectral vectors $\boldsymbol{s}_1^{\mathcal{T}} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_t, \ldots, \boldsymbol{s}_{\mathcal{T}}\}$ in which $\boldsymbol{s}_t$ is the magnitude spectrum of the frame and $s_{t,f}$ is the magnitude at time-frequency (t, f). An I-channel filter-bank model transforms each $\boldsymbol{s}_t$ into a vector of log-energies $\boldsymbol{x}_t$ such that an output feature $x_{t,i}$ is the windowed log-energy of the i-th channel:

$$x_{t,i} = \log_{10}\Big(\sum_{f \in B_i} w_{i,f}\, s_{t,f}\Big) \quad \text{for } i = 1, \ldots, I, \tag{3}$$

where B_i represents the channel interval and $w_{i,f}$ the weighting at frequency f provided by the i-th filter. From the vector of log energies, the cepstrum vector $\mathbf{c}_t$ is computed via an inverse discrete cosine transform (IDCT) as $c_{t,q} = \sum_{i=1}^{I} x_{t,i} u_{i,q}$ for $q = 1, \ldots, Q$, where Q is the number of cepstral coefficients and $u_{i,q} = \frac{q\pi}{I}(i - \frac{1}{2})$. Similar to [2], the magnitude response of the filter $w_{i,f}$ in i-th channel is constrained to a Gaussian-form:

$$w_{i,f} = \alpha_i \exp\left[-\beta_i \left\{p(\gamma_i) - p(f)\right\}^2\right], \quad \text{for } i = 1, \cdots, I,$$

for $i = 1, \ldots, I$, where the trainable parameters $\beta_i > 0$ and γ_i determine bandwidth and center frequency, and α_i is the trainable "gain" parameter in the i-th channel. $p(f)$ maps the linear frequency f onto the perceptual representation, which in this paper is the Mel scale. Here, Θ is the set of α_i, β_i and γ_i.

3.2. Filter-bank optimization

Below, we summarize the filter-bank derivatives of center frequencies, bandwidths and gains. Full details can be found in [8]. Let ϕ be any adjustable filter-bank parameter. The transformation $\overline{\phi} = \log(\phi)$ is used to constrain the filter-bank's parameter to stay positive. The update rule is

$$\phi[\tau+1] \quad = \quad \exp\left(\log(\phi[\tau]) - \rho_\tau \mathbf{U2} \delta\overline{\phi}\right) \tag{4}$$

where $\delta\overline{\phi} = \dfrac{\partial \ell(\boldsymbol{s}_1^{\mathcal{T}}; \Phi)}{\partial \overline{\phi}}$. The chain rule of differential calculus gives

$$\delta\overline{\phi} \quad = \quad \sum_{t=1}^{\mathcal{T}} \sum_{q=1}^{Q} \frac{\partial \ell(\boldsymbol{s}_1^{\mathcal{T}}; \Phi)}{\partial c_{t,q}} \sum_{i=1}^{I} \frac{\partial c_{t,q}}{\partial x_{t,i}} \frac{\partial x_{t,i}}{\partial \overline{\phi}} \tag{5}$$

$$= \quad \sum_{t=1}^{\mathcal{T}} \sum_{q=1}^{Q} \sum_{i=1}^{I} u_{i,q} \mathcal{I}_{t,q} \mathcal{O}_{t,i}. \tag{6}$$

where $\mathcal{I}_{t,q} = \dfrac{\partial \ell(\boldsymbol{s}_1^{\mathcal{T}}; \Phi)}{\partial c_{t,q}}$ and $\mathcal{O}_{t,i} = \dfrac{\partial x_{t,i}}{\partial \overline{\phi}} = \sum_{f=1}^{F} \upsilon_{t,i,f} \xi_{i,f}$ with $\upsilon_{t,i,f} = \dfrac{\partial x_{t,i}}{\partial w_{i,f}} = \frac{s_{t,f}}{\log(10) 10^{x_{t,i}}}$ and $\xi_{i,f} = \dfrac{\partial w_{i,f}}{\partial \overline{\phi}}$.

Let ψ_t^j corresponds to the state occupied by the cepstrum-vector $\mathbf{c}_t$ along the Viterbi path for word W_j; The state has $N_{\psi_t^j}$ mixture components and $\boldsymbol{\mu}_{\psi_t^j, n, q}$ is the q-th component of the means vector $\boldsymbol{\mu}_{\psi_t^j, n}$ of n-th Gaussian component. Then,

$\mathcal{I}_{t,q} = -\sum_{j=1}^{M} \sum_{n=1}^{N_{\psi_t^j}} \delta\boldsymbol{\mu}_{\psi_t^j, n, q}$ where $\delta\boldsymbol{\mu}_{\psi_t^j, n, q} = \dfrac{\partial \ell(d_k(\boldsymbol{s}_1^{\mathcal{T}}; \Phi))}{\partial \boldsymbol{\mu}_{\psi_t^j, n, q}}$ is the derivative of means described elsewhere [9].

$\mathcal{O}_{t,i}$ solely depends on the nature of the parameter $\overline{\phi}$ and is defined below for each parameter type.

3.2.1. Center frequency adjustment

Let $\phi = \Gamma_i = p(\gamma_i)$, where Γ_i represents the center frequency of channel $\hat{\imath}$ in the perceptual domain. For $\overline{\Gamma}_i = \log(\Gamma_i)$, it follows that

$$\frac{\partial w_{i,f}}{\partial \overline{\Gamma}_{\hat{\imath}}} \quad = \quad -2\beta_i p(\gamma_i) \left(p(\gamma_i) - p(f)\right) w_{i,f}\, \boldsymbol{\chi}(i, \hat{\imath}). \tag{7}$$

where $\boldsymbol{\chi}(a, b) = 1$ if a equals b and zero otherwise.

3.2.2. Bandwidth adjustment

Here ϕ is the parameter β_i of the $\hat{\imath}$-th channel. Let $\overline{\beta_i} = \log(\beta_i)$. It follows that,

$$\frac{\partial w_{i,f}}{\partial \overline{\beta_i}} = -\beta_i \left(p(\gamma_i) - p(f)\right)^2 w_{i,f}\, \boldsymbol{\chi}(i, \hat{\imath}) \tag{8}$$

3.2.3. Gain adjustment

ϕ is the parameter α_i of channel $\hat{\imath}$. For $\overline{\alpha_i} = \log(\alpha_i)$,

$$\frac{\partial w_{i,f}}{\partial \overline{\alpha_i}} = w_{i,f}\, \boldsymbol{\chi}(i, \hat{\imath}). \tag{9}$$

3.3. Embedding dynamic features

The inclusion of dynamic features in the DFE optimization framework is as follows. Let $\Delta\mathbf{c}_t = [\Delta c_{t,1}, \ldots, \Delta c_{t,q}, \ldots, \Delta c_{t,Q}]^{\mathrm{T}}$, where $\Delta c_{t,q}$ is the q-th feature-index (quefrency). The polynomial regression coefficients are defined as $\Delta c_{t,q} = C \sum_{\rho=-R}^{R} \rho c_{t+\rho,q}$ where $C = 1/2 \sum_{\rho=1}^{R} \rho^2$ and R is the number of forward and backward frames used for calculating the regression coefficients. If the feature vector contains delta parameters, optimization of the

filter-bank's parameters $\overline{\phi}$ should take this fact into account which finally gives [8]:

$$\delta\overline{\phi} = \sum_{t=1}^{\mathcal{T}} \sum_{q=1}^{Q} \sum_{i=1}^{I} u_{i,q} \left\{ \mathcal{I}_{t,q} \mathcal{O}_{t,i} + \mathcal{I}_{t,q+Q} \left(C \sum_{\rho=-R}^{R} \rho \mathcal{O}_{t+\rho,i} \right) \right\} \quad (10)$$

4. EXPERIMENTAL EVALUATION

4.1. Database

The task is isolated word recognition from the ISOLET database [10]. The database consists of two examples of each letter of the English alphabet uttered by 150 American English speakers, 75 males and 75 females. The database is divided into 5 portions of 30 speakers. We used the first 4 portions for training (120 speakers, 6240 utterances) and the last portion for testing (30 speakers, 1560 utterances). The ISOLET database is a highly confusable task, with many letters sharing similar vowels. In particular, the E-set subset, the set of nine letters ending with the sound /e/, has been a good framework for testing the performance of various discriminative algorithms. All training scenarios were done targeting the discrimination of all letters.

The speech signal was downsampled to 8 kHz. For all experiments, we used 12 cepstral coefficients from a 24-order filter-bank at a 5ms frame rate. The initial configuration of the filter-bank emulated Mel-based filter-bank cepstrum (MFCC) with Gaussian filters. DFE was carried out optimizing center frequencies, bandwidths, and gain simultaneously; we refer to this feature set as discriminative filter-bank-based cepstrum (DFCC).

4.2. Experimental setup

Each word is modeled by a 5-mixture density left-to-right continuous HMM consisting of 5 states. Only the mean vectors were optimized. The baseline systems use MFCC with either MLE or MCE-trained models. We tested the following DFE configurations.

The first configuration is MLE-estimated models using DFCC as features. DFCC was precomputed through an iterative process that optimizes the features while models are untrained and then re-estimates models by MLE. This iterative process was run twice. This configuration is referred to as MLE/DFCC. The second configuration is classical MCE training of HMM using DFCC as features, starting from models generated by the MLE/DFCC configuration. The third configuration jointly optimizes the features and the HMM within an integrated system by MCE. This configuration is referred to as MCE-I/DFCC-I.

4.3. Dependency on the classifier structure

The first experiment examines whether DFCCs optimized on a particular structure of the recognizer can be used on a different structure in the context of the MLE/DFCC configuration. DFCCs were generated using a 5-state left-to-right HMM with 5 Gaussian mixtures per state. We simulate different recognizer structures by varying the number of mixture components per state.

Table 1 summarizes the results on the ISOLET database and its E-set sub-task. As expected, DFE training shows the best performance within the configuration for which it has been optimized on both the ISOLET task and the E-set task. On the E-set task, DFE realizes more than 2.4% absolute improvement in recognition rate from the MLE/MFCC configuration. Also, the MLE/DFCC system of 5-mixture density HMM produces similar performance to bigger MLE/MFCC systems with a higher number of mixtures, showing that DFE can realize a smaller and more efficient recognizer.

Table 1. Recognition rate on ISOLET and E-set using MLE with various mixture components for MFCC and DFCC. The DFCC has been generated using the 5-mixture density HMM.

# of mixtures	ISOLET		E-set	
	MFCC	DFCC	MFCC	DFCC
3	77.05	76.08	53.88	50.37
5	77.82	78.78	52.59	55.00
8	79.16	78.26	55.74	54.62
12	78.20	76.98	55.37	53.88

Table 2. Recognition rate on ISOLET and E-set using MLE, classical MCE, and various configurations of DFE.

Criterion		Task	
models	features	ISOLET	E-set
MLE	MFCC	77.82	52.59
MLE	DFCC	78.78	55.00
MCE	MFCC	83.84	66.85
MCE	DFCC	83.84	68.14
MCE-I	DFCC-I	84.35	68.70

4.4. DFE configurations

Table 2 shows the results of the various DFE configurations on the ISOLET task and its E-set sub-task, using a 5-mixture density HMM. The results of the MLE optimization are also shown. From this table, it is quite obvious that all MCE-based approaches outperform MLE-based ones across all feature sets.

The first row of results in the table shows the baseline performance of MLE-derived models using MFCCs. The second row shows the result of the MLE/DFCC configuration, in which models are not discriminatively trained but the features are. This configuration exhibits a 0.96% absolute reduction in error rate on the ISOLET task and a 2.41% absolute reduction in error rate on the E-set task compared to the MLE/MFCC configuration, confirming the well-known observation that the using MLE on discriminative features leads to improved performance.

The third row shows results of the classical use of MCE training using MFCC. In this context, models are discriminatively trained and features are not. This MCE/MFCC configuration clearly outperforms all MLE-based configurations. The fourth row on the table displays the results of the MCE/DFCC configuration, in which both models and features are discriminatively trained but not integrated. Although, the MCE/DFCC configuration gives similar performance to the MCE/MFCC's one on the ISOLET task, it is more

efficient in the context of acoustically similar words: 1.29% absolute error rate reduction on the E-set from the MCE/MFCC configuration. The last row displays the results of the MCE-I/DFCC-I configuration, where both models and feature are discriminatively trained and integrated. This configuration exhibits the best result on both tasks. Compared to the MCE/DFCC configuration, the integrated system is more efficient.

4.5. Using Dynamic Features

In this section we performed the same experiment similar to the previous section, this time, including dynamic features in the feature set. Within the DFE framework, dynamic parameters can be included in two ways. One technique is to embed dynamic features within the DFE training process as was described in section 3.3; these features are referred to as ΔDFCC-E. Another method is to compute the linear regression from the DFE-trained static features; these features are referred to as ΔDFCC-R. This later approach may be an attractive solution when the front-end is a complex process that may lead to instability in the training process. We implemented two methods and tested them against the baseline delta MFCCs.

Table 3. Recognition rate on ISOLET and E-set for MLE, classical MCE and DFE, using dynamic features. ΔDFCC-R refers to calculating delta parameters using the regression formula on previously optimized static DFCC. ΔDFCC-E refers to delta parameters, optimized within the DFE training.

Criterion		Task	
models	features	ISOLET	E-SET
MLE	MFCC+ΔMFCC	85.32	68.88
MLE	DFCC + ΔDFCC-R	86.31	68.70
MLE	DFCC + ΔDFCC-E	85.57	67.77
MCE	MFCC+ΔMFCC	90.25	80.37
MCE	DFCC+ΔDFCC-R	87.69	84.25
MCE	DFCC + ΔDFCC-E	89.55	81.66
MCE-I	DFCC-I + ΔDFCC-E	90.44	81.85

Table 3 displays the performance of various optimization methodologies when using dynamic parameters. The use of discriminative dynamic parameters does not show a clear-cut improvement. Embedded optimization of delta parameters may have generated a system highly sensitive to training parameters. Few points, however, are worth mentioning: the MCE/DFCC+ΔDFCC-R, where the features and the models are both discriminatively trained but not integrated, displays disappointing results on the ISOLET task while outperforming all configurations on the E-set task. The MCE-I/DFCC-I + ΔDFCC-E, where models and features are discriminatively trained within an integrated system shows the best performance on the ISOLET task.

5. CONCLUSION

We described a study of the Discriminative Feature Extraction (DFE) method to filter-bank optimization in the context of HMM-based isolated word recognition. The study applied various configurations of the algorithm, where discriminative training was selectively used to optimize features and models either jointly or separately. Having a level of discriminative training applied to either the front-end or the models leads to improved performance. The best performance is obtained when discriminative training is applied to both the front-end and the model simultaneously in an integrated fashion.

6. ACKNOWLEDGEMENTS

The author would like to thank his former colleagues at ATR HIP, Shigeru Katagiri and Erik McDermott, for their contribution in this work. Sincere thanks to John Pitrelli and Jane Snowdon of IBM T. J. Watson Center for reviewing the manuscript.

7. REFERENCES

[1] A. Biem and S. Katagiri, "Feature Extraction Based on Minimum Classification Error/Generalized Probabilistic Descent Method," in *Proc. of ICASSP*, vol. 2, pp. 275–278, 1993.

[2] A. Biem, S. Katagiri, E. McDermott, and B.-H Juang, "An Application of Discriminative Feature Extraction to Filter-bank-based Speech Recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 9, no. 2, pp. 96–110, Feb. 2001.

[3] C. Rathinavelu and L. Deng, "HMM-based Speech Recognition Using State-Dependent, Discriminately Derived Transforms on Mel-Warped DFT Features," *IEEE Trans. on Speech and Audio Processing*, vol. 5, no. 3, pp. 243–256, May 1997.

[4] M. G. Rahim and C. H. Lee, "Simultaneous ANN Feature and HMM Recognizer Design Using String-based Minimum Classification Error (MCE) Training," in *Proc. of ICSLP*, pp. 1824–1827,1996,

[5] A. De la Torre, A. M. Peinado, A. J. Rubio, and V. Sanchez, "An Application of Minimum Classification Error to Feature Space Transformation for Speech Recognition," *Speech Communication*, vol. 20, pp. 273–290, 1996.

[6] B. Mak, Y. C. Tam, and Q. Li, "Discriminative Auditory Features for Robust Speech Recognition," in *Proc. of ICASSP*, vol. I, pp. 381–384, 2002,

[7] B.-H. Juang and S. Katagiri, "Discriminative Learning for Minimum Error Classification," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 40, no. 12, pp. 3043–3054, Dec. 1992.

[8] A. Biem, *Discriminative Feature Extraction Applied to Speech Recognition*, Ph.D. thesis, Université Paris 6, 1997.

[9] Juang B.-H, Wu Hou, and C.-H. Lee, "Minimum Classification Error Rate Methods for Speech Recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 5, no. 2, pp. 257–265, 1997.

[10] R. Cole, Y. Muthusamy, and M. Fanty, "The ISOLET Spoken Letter Database," Tech. Report 90-004, Inst. Beaverton, 1990.