Minimum Detection Error Training of Subword Detectors

Alfonso M. Canterla, Magne H. Johnsen

Department of Electronics and Telecommunications, NTNU
NO-7491 Trondheim, Norway
alfonso@iet.ntnu.no
mhj@iet.ntnu.no

***Abstract*—This paper presents methods and results for optimizing subword detectors in continuous speech. Speech detectors are useful within areas like detection-based ASR, pronunciation training, phonetic analysis, word spotting, etc. We propose a new discriminative training criterion for subword unit detectors that is based on the Minimum Phone Error framework. The criterion can optimize the F-score or any other detection performance metric. The method is applied to the optimization of HMMs and MFCC filterbanks in phone detectors. The resulting filterbanks differ from each other and reflect acoustic properties of the corresponding detection classes. For the experiments in TIMIT, the best optimized detectors had a relative accuracy improvement of 31.3% over baseline and 18.2% over our previous MCE-based method.**

***Index Terms*—Detection, discriminative training, MPE, filterbank.**

I. Introduction

Detection of phonetic events such as phones and articulatory features has applications within phonetic analysis, word spotting, computer aided pronunciation training [1] and specially in detection-based ASR [2] [3] [4] [5]. In the latter accurate detectors are decisive for the performance of the system. A detector is a binary classifier that discerns between patterns that share a specific quality (the *class*) and the rest (the *anti-class*). A possible approach to continuous speech detection is to adapt the standard ASR framework for the two-class problem.

A standard ASR system extracts acoustic features from the frequency content and the time dynamics of the speech signal. In Mel-frequency cepstral coefficients (MFCCs), the dominant speech representation in ASR, the short-term spectrum is processed with a filterbank (FB) that imitates two important properties of human audition: critical bands and a logarithmic frequency scale. Note that this feature extraction is common to all classes, i.e. a single MFCC extraction is performed for every time frame. The frequency content variation over classes is modeled by the class-specific MFCC-densities in the HMM-based classifier.

In the detector case, the parameters of the feature extractor and the decoder can be improved for each detector. The HMM state-density parameters are good candidates for detector-specific optimization. In the feature extractor an obvious choice is to keep the MFCC structure, as this has shown to be state-of-the-art preprocessing for many years in ASR. However, the structure parameters should be optimized for each specific detection problem.

Discriminative training techniques have been successfully applied to ASR, e.g. Maximum Mutual Information (MMI) [6], Minimum Phone Error (MPE) [7] and Minimum Classification Error (MCE) [8]. Moreover, in [9] it was shown that these methods are closely related. MCE training of models improved detection performance in [10]. However, they found that the standard method did not guarantee an error decrease for the target class and argued for a modified version of MCE-training for detection.

In [11] we modified standard string-based MCE to train subword detectors. FB and means in a HMM-based structure were optimized with this method for the task of phone and articulatory feature detection. We showed that the shape of the standard FB can be modified to extract information that is relevant to the specific detection task. In addition, we included a brief analysis of the resulting FBs with respect to phonetic knowledge of the detection classes. As far as we know, those are the only published experiments on data-driven FB optimization for detection. However, the performance function that we used in this MCE-based training method was not directly related to the evaluation criterion applied to the detectors in the test phase.

Previous work in data-driven FB optimization for speech recognizers includes a joint training of FB and a prototype-based distance classifier [12], optimization of robust features [13] [14], FB design using MPE [15], Linear Discriminant Analysis [16] or minimum entropic distance [17], etc. In addition, MPE has also been applied to other feature optimization methods, e.g. MPE-trained feature transforms were studied in [18] and [19].

The main contributions of this paper are as follows. First, we propose a novel discriminative training for subword detectors capable of directly optimizing the F-score or any other performance measure for detection. We call this method *Minimum Detection Error* (MDE) training because it is based on the MPE framework. Secondly, we apply MDE to optimize the HMM parameters and FB of the detector structure proposed

in [11]. Finally, we analyze the optimized FBs and compare MDE to our previous training method.

The paper is organized as follows: Section II describes the structure of our detectors and discusses their evaluation. Section III presents MDE-training of the FB and HMMs in the detectors. Section IV describes the phone detection experiments, results and analysis of the optimized FBs. Finally, Section V presents the conclusions and plans for future work.

II. Detectors

We focus on HMM-based detectors for phonemes in continuous speech. The class was modeled by the HMM of the phone to detect. The anti-class was modeled with all the other phone HMMs in parallel. Therefore, any detector can be regarded as a standard HMM-based phone recognition system optimized to improve the accuracy of a specific phone. This structure is also suitable to build detectors for articulatory features, as it was shown in [11].

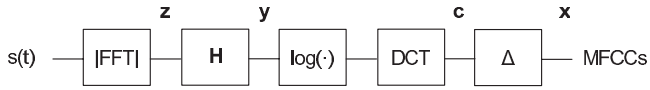


Fig. 1. MFCC feature extraction.

A. MFCC feature extraction

This module should extract information that discriminates between the class and the anti-class. Fig. 1 shows the block diagram for a MFCC preprocessor. First, the input speech signal is windowed and the magnitude of the FFT is computed for each frame (the resulting vector is referred to as $\mathbf{z}$). Then a FB performs the linear mapping $\mathbf{y} = \mathbf{H}\,\mathbf{z}$, where $\mathbf{H}$ is the FB matrix. Each component $y_i = \mathbf{h}_i^T \mathbf{z}$ represents the energy in the band of the i-th filter. This block is different in each detector. The following block is a logarithmic transformation to imitate loudness sensitivity. Then a DCT maps the log-energies to the cepstral domain and decorrelates the components. The cepstral vector $\mathbf{c}$ keeps only the first N_{cep} components, which have most of the discriminative information. The last module adds the first and second order time derivatives to $\mathbf{c}$ and outputs the observation vector $\mathbf{x}$.

B. Evaluation of detectors

In continuous speech recognition the segment sequence is aligned with a reference to find hits (H), substitutions (S), deletions (D) and insertions (I). In detection the relevant outcomes are *hits* (H_c), *misses* (S_c and D_c) and *false alarms* (S_{ac} and I_c), where the subindexes "c" and "ac" refer to the class and the anti-class, respectively. In practice we must accept a trade off between hits, false alarms and misses. Precision (P) and recall (R) are commonly used to score detectors, where $P = hits/(hits + false\_alarms)$ and $R = hits/(hits + misses)$. In addition, F-score is a measure that combines precision and recall:

$$F = \frac{2PR}{P+R} = \frac{2H_c}{2H_c + S_c + D_c + I_c + S_{ac}} \ . \qquad (1)$$

It is important to consider that most patterns belong to the anti-class (unbalanced data problem). This means that S_{ac} can be high compared to the other terms. In addition, in the context of detection-based ASR it is important not to miss candidates and recall is usually prioritized over precision [5]. We proposed in [11] to use the accuracy of the detector as evaluation criterion:

$$A_c = \frac{H_c - I_c}{N_c} = R - \frac{I_c}{N_c} \ , \qquad (2)$$

where $N_c = H_c + S_c + D_c$. This is the commonly used ASR accuracy, however modified for detection as it is limited to include only class segments. Both recall, R, and the detector accuracy, A_c, solve the unbalanced data problem by not considering S_{ac}. However, A_c is more restrictive than R because it counts the inserted class segments I_c.

III. Minimum Detection Error Training

Minimum Phone Error (MPE) optimizes the parameters of a recognizer to improve the overall phone accuracy. In this method, a performance function is built that estimates the expected phone accuracy on the training set as a function of the model parameters. The MPE-optimized parameters are those that maximize the performance function. This section presents Minimum Detection Error (MDE) training, the application of the MPE framework to detection.

The training data is a set of sentences $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_K\}$ and their phone labels. A sentence has a number of frames: $\mathbf{X}_k = \{\mathbf{x}_1, \ldots, \mathbf{x}_{T(k)}\}$. The key idea is that we want to train the detector structure to optimize a detector evaluation measure, see Section II-B. Therefore, the performance function J in MDE estimates the expected detector score on the training set as a function of the detector structure parameters Λ:

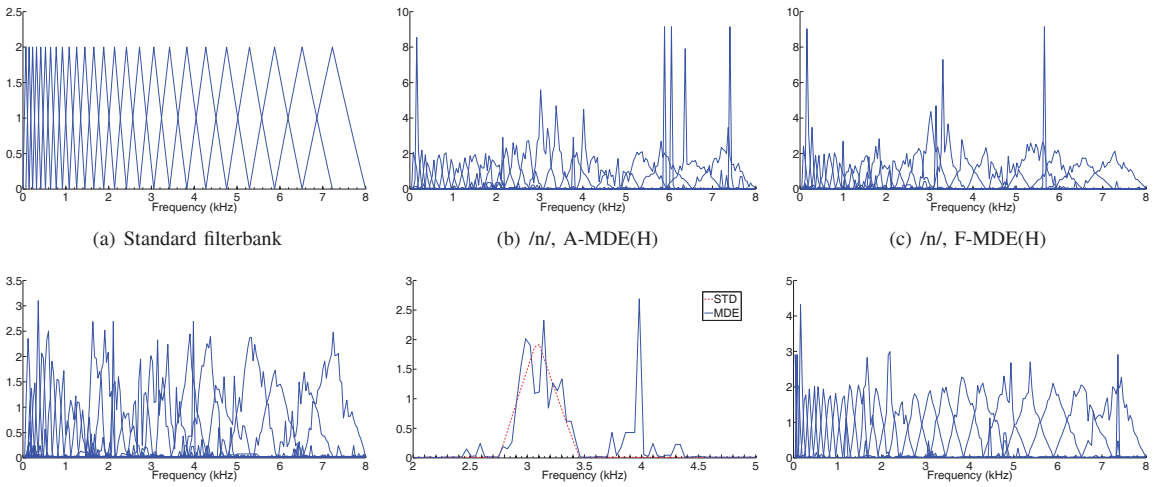
$$J(\Lambda) = \frac{1}{K} \sum_k \bar{S}_k(\Lambda) \ , \qquad (3)$$

where $\bar{S}_k(\Lambda)$ is the expected detection score of sentence k given the current set of parameters Λ and K is the total number of sentences. We estimate this expectation using a set a set of transcriptions $\{L_{kj}\}_{j=0}^{N}$, where L_{k0} is ground truth and the rest are generated with an N-best algorithm. These transcriptions label each frame $\mathbf{x}$ at the model and state level with Viterbi alignment. Then we have that

$$\bar{S}_k(\Lambda) = \sum_j P(L_{kj}/\mathbf{X}_k, \Lambda) S_{kj} \ , \qquad (4)$$

where S_{kj} is the detection score (e.g. F-score or A_c) of transcription L_{kj} and it is independent of Λ. This score is computed as follows: first L_{kj} is string aligned against L_{k0} (ground truth) based on the phone labels. Then these phone labels are mapped to detection labels, i.e. "class" and "anti-class", to compute H_c, D_c, etc. The last step is to evaluate with, e.g. Eq. 1 or 2.

The performance function J is maximized by updating the parameters Λ iteratively with the *Rprop* (resilient backpropagation) algorithm. In the following we discard the constant



(a) Standard filterbank (b) /n/, A-MDE(H) (c) /n/, F-MDE(H)

(d) /ih/, F-MDE(H) (e) /ih/, F-MDE(H), filter 18 (f) /sh/, A-MDE(H)

Fig. 2. Selected examples of filterbanks.

$\frac{1}{K}$ in J for simplicity. Applying the chain rule of differential calculus:

$$\frac{\partial J(\Lambda)}{\partial \Lambda} = \sum_{k,j} S_{kj} \frac{\partial P(L_{kj}/\mathbf{X}_k, \Lambda)}{\partial \Lambda} \ . \tag{5}$$

The posterior $P(L_{kj}/\mathbf{X}_k, \Lambda)$ can be expressed in terms of likelihoods applying Bayes' theorem:

$$P(L_{kj}/\mathbf{X}_k, \Lambda) = \frac{p(\mathbf{X}_k/L_{kj}, \Lambda)P(L_{kj})}{\sum_u p(\mathbf{X}_k/L_{ku}, \Lambda)P(L_{ku})} \ . \tag{6}$$

In addition we can consider that

$$\frac{\partial p(\mathbf{X}_k/L_{kj}, \Lambda)}{\partial \Lambda} = p(\mathbf{X}_k/L_{kj}, \Lambda) \frac{\partial \log p(\mathbf{X}_k/L_{kj}, \Lambda)}{\partial \Lambda} \ , \tag{7}$$

where $\log p(\mathbf{X}_k/L_{kj}, \Lambda)$ is the loglikelihood of sentence k and transcription j. Then it can be verified that

$$\frac{\partial J(\Lambda)}{\partial \Lambda} = \sum_{k,j} (S_{kj} - \bar{S}_k) P(L_{kj}/\mathbf{X}_k, \Lambda) \frac{\partial \log p(\mathbf{X}_k/L_{kj}, \Lambda)}{\partial \Lambda} \ . \tag{8}$$

In this paper we focus on the optimization of means and FB matrices. The Viterbi loglikelihood is given by

$$\log p(\mathbf{X}_k/L_{kj}, \Lambda) = \log a_{kj} + \sum_{t=1}^{T(k)} \log[b_{s_t}^{i_t}(\mathbf{x}_t)] \ , \tag{9}$$

where i_t and s_t are the model and state that segmentation j assigns to frame $\mathbf{x}_t$ in sentence k, a_{kj} is a segmentation dependent constant independent of Λ, and $b_s^i(\mathbf{x})$ is the state probability density function given by a Gaussian mixture model. Then it can verified that the derivatives of the means and the FB matrix are given by

$$\frac{\partial \log p(\mathbf{X}_k/L_{kj}, \Lambda)}{\partial \boldsymbol{\mu}_{sm}^i} = \sum_{t,m} \delta_{jtis} \frac{c_{sm}^i \cdot b_{sm}^i(\mathbf{x}_t)}{b_s^i(\mathbf{x}_t)} {\boldsymbol{\Sigma}_{sm}^i}^{-1} (\mathbf{x}_t - \boldsymbol{\mu}_{sm}^i) \tag{10}$$

and

$$\frac{\partial \log p(\mathbf{X}_k/L_{kj}, \Lambda)}{\partial \mathbf{H}} = \sum_{t,m} \frac{c_{s_t m}^{i_t} \cdot b_{s_t m}^{i_t}(\mathbf{x}_t)}{b_{s_t}^{i_t}(\mathbf{x}_t)} \frac{\partial \mathbf{x}_t}{\partial \mathbf{H}} {\boldsymbol{\Sigma}_{s_t m}^{i_t}}^{-1} (\boldsymbol{\mu}_{s_t m}^{i_t} - \mathbf{x}_t) \ , \tag{11}$$

where δ_{jtis} is an indicator function for the event $\{L_j$ labels $\mathbf{x}_t$ as $(i, s)\}$ and the index k has been omitted for clarity. As in in [11], the term $\frac{\partial \mathbf{x}_t}{\partial \mathbf{H}}$ is computed using the complete observation vector, i.e. static cepstrum and derivatives. Considering $\mathbf{x}_t = [\mathbf{c}_t^T, \mathbf{d}_t^T, \mathbf{a}_t^T]^T$, $\boldsymbol{\mu} = [\boldsymbol{\mu}_c^T, \boldsymbol{\mu}_d^T, \boldsymbol{\mu}_a^T]^T$ and $\boldsymbol{\Sigma}^{-1} = diag(\boldsymbol{\Sigma}_c^{-1}, \boldsymbol{\Sigma}_d^{-1}, \boldsymbol{\Sigma}_a^{-1})$, where model, state and mixture indices are omitted for clarity, $\mathbf{d}_t = \mathbf{c}_{t+2} - \mathbf{c}_{t-2}$, $\mathbf{a}_t = \mathbf{d}_{t+2} - \mathbf{d}_{t-2}$ and the chain rule applied, it can be verified that

$$\begin{aligned} \frac{\partial \mathbf{x}_t}{\partial \mathbf{H}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu} - \mathbf{x}_t) &= (\mathbf{w}_t^c ./ \mathbf{y}_t) \mathbf{z}_t^T + (\mathbf{w}_t^d ./ \mathbf{y}_{t+2}) \mathbf{z}_{t+2}^T \\ &- (\mathbf{w}_t^d ./ \mathbf{y}_{t-2}) \mathbf{z}_{t-2}^T + (\mathbf{w}_t^a ./ \mathbf{y}_{t+4}) \mathbf{z}_{t+4}^T \\ &- (\mathbf{w}_t^a ./ \mathbf{y}_{t-4}) \mathbf{z}_{t-4}^T - 2(\mathbf{w}_t^a ./ \mathbf{y}_t) \mathbf{z}_t^T \ , \end{aligned} \tag{12}$$

where the notation of Fig. 1 is followed, "./" is elementwise division, $\mathbf{w}_t^c = \mathbf{D}^T \boldsymbol{\Sigma}_c^{-1} (\boldsymbol{\mu}_c - \mathbf{c}_t)$, $\mathbf{w}_t^d = \mathbf{D}^T \boldsymbol{\Sigma}_d^{-1} (\boldsymbol{\mu}_d - \mathbf{d}_t)$ and $\mathbf{w}_t^a = \mathbf{D}^T \boldsymbol{\Sigma}_a^{-1} (\boldsymbol{\mu}_a - \mathbf{a}_t)$. To ensure that the FB coefficients remain positive, the parameter transformation in [8, Eq. 32] is applied.

Each iteration n of our MDE algorithm can be summarized in the following steps:

1) Extract MFCCs using $\mathbf{H}^n$
2) Generate N-best transcriptions with present phone HMMs: $\boldsymbol{\mu}^n$.
3) Compute detection scores S_{kj}, $\forall k, j$.
4) Generate Viterbi state alignment of the N+1 transcriptions.
5) Compute Eq. 8 for $\mathbf{H}$ and $\boldsymbol{\mu}$ (use Eq. 10 and 11).
6) Find $\mathbf{H}^{n+1}$ and $\boldsymbol{\mu}^{n+1}$ with Rprop.

IV. Experiments

To evaluate the MDE training algorithm experiments have been performed on the TIMIT acoustic-phonetic continuous speech corpus. The basic setup is the same as used for the MCE-based experiments reported in [11].

A. Task

Detectors with optimized FBs and models were applied to the task of phone detection on TIMIT. We chose this database because it is a well-known standard reference and it is labeled at the phoneme level. We used the designated training set of 462 speakers (3696 sentences), that is excluding SA-sentences. A development set of 50 speakers (400 sentences) was used for intermediate experiments. Results are reported on the NIST defined core test set of 24 speakers (192 sentences).

The experiments were based on a set of 39 phonemes. The manual TIMIT labeling consists of 61 acoustic-phonetic symbols. We merged plosive closures and bursts and applied the standard mapping to 39 phones. The acoustic parameterization consisted of 13 static MFCCs (including C_0) with their first and second order derivatives. The sampling frequency was 16 kHz, and frames were extracted every 10 ms with 25 ms Hamming window. The FB was specific to each optimized detector.

B. Experimental settings

Baseline (BL) phone detectors were built using the standard FB (see Fig. 2(a)) and a set of 39 maximum likelihood trained HMMs. The phonemic transcription of TIMIT was used to train monophone HMMs with 3 states and 10 mixtures. In the MDE optimization, BL detectors were used as the starting point.

Two MDE training experiments were performed: F-score and class accuracy optimization. In each experiment two implementations of MDE were tested: 1) only means were trained, referred to as (μ) and 2) only **H** was trained, referred to as (H). We can refer to these experiments as *score-MDE(Implementation)*, e.g. *A-MDE(H)* denotes class accuracy optimization through MDE-training of the FB matrix **H** in the detector.

In our MDE implementation the number of competing hypothesis N was set to 10. In each detector, we selected the iteration corresponding to the best score in the development set. The HTK Toolkit was used for standard maximum likelihood embedded training of HMMs. The standard FB had 26 triangular shaped filters with 201 points, where the center frequencies and bandwidths were uniformly spaced according to the Mel-scale. The grammar was an unconstrained phone loop and the language model used under training and testing was a 0-gram (uniform model).

C. Results and discussion

This section presents the performance and FBs for the detectors. First, we discuss the test results for the F-score and accuracy optimization experiments, which are presented in Tables I(a) and I(b), respectively. The large number of detectors prevents us from presenting a thorough analysis of each case; instead we present a weighted average detector performance. For the F-score we computed $\bar{F} = \sum_c (N_c \cdot F_c) / \sum_c N_c$, and similarly for the class accuracy. In each experiment, we present the averages for both the F-score and the class accuracy of the optimized detectors. However, we present also some specific examples of detectors: Table II shows the class accuracy, F-score, precision and recall for the detectors of /ih/, /n/ and /sh/. We chose these detectors because they were used as specific examples in [11]. Secondly, we make a brief analysis of the optimized FBs, focusing on those shown in Fig. 2.

TABLE I
Class averaged performance after optimization

(a) F-scores optimization.

Score	BL	F-MDE(H)	F-MDE(μ)
$\bar{F}$	67.0	69.1	71.9
$\bar{A}_c$	60.7	63.9	66.6

(b) Class accuracy optimization.

Score	BL	A-MDE(H)	A-MDE(μ)
$\bar{F}$	67.0	67.2	69.9
$\bar{A}_c$	60.7	70.2	73.0

As expected, discriminative training improved the average detector performance in both experiments. In the first experiment, the relative improvements with respect to BL were 6.4% for *F-MDE(H)* and 14.8% for *F-MDE*(μ). In the second experiment, the corresponding improvements were 24.2% for *A-MDE(H)* and 31.3% for *A-MDE*(μ). The average F-score in *F-MDE* experiments was higher than the corresponding F-score in *A-MDE* experiments, and vice versa. This is reasonable because increased recall usually comes at the expense of decreased precision. In addition, this shows that MDE-training focused on the optimization of the chosen performance criterion, F-score in the first experiment and class accuracy in the second. In both experiments, the average performance of *MDE*(μ) was higher than that of *MDE*(H). This can probably be explained by the fact that the total number of parameters in the HMM means was much higher than in the FB matrix (45630 vs. 5226). However, in some detectors we found the opposite, e.g. Table II(a) shows that *A-MDE*(H) improved the result of *A-MDE*(μ) for the /ih/ detector.

We are also interested in comparing MDE with our previous MCE-based training method. In principle, an advantage of MDE is that it can be used to optimize *directly* the chosen detection performance measure. In our previous method the performance function J was not directly related to any detection evaluation criterion. Our approach was then to use the chosen performance metric in the cross validation, but this was at best suboptimal. In addition, while recall is prioritized in detection-based ASR, other applications could prefer an optimized F-score. Therefore, MDE is also more flexible than our previous method because it can optimize detectors

focusing on the specific figure-or-merit of the application. The cross validation stop criterion for the experiment presented in [11] was class accuracy. Following the notation introduced earlier, it can be referred to as *A-MCE*. For convenience, we have adapted and reproduced the results from [11, Tables 1, 2 and 3] in Table III. The results in Table III(a) can be compared with those in Table I(b). However, we have to consider that the previous best results were with *A-MDE*(H, μ), where the FB matrix and the model means where optimized *jointly*.

First, the average performance of *A-MDE(H)* is higher than that of *A-MCE*(μ) (70.2% vs 67.0%) even if the number of optimized parameters was much smaller. In addition, in some cases detectors optimized with *A-MDE(H)* performed even better than those trained with *A-MCE*(H, μ), e.g. comparing Table II and III(b), we find 90.5% vs 72.9% for /ih/ and 74.8% vs 73.5% for /n/. Second, the average performance of *A-MDE*(μ) is higher than that of *A-MCE*(μ) (73.0% vs 67.0%, a relative improvement of 18.1%), and close to *A-MCE*(H, μ) (73.0% vs 73.7%). Also in this case some detectors optimized with *A-MDE*(μ) performed better than those trained with *A-MCE*(H, μ), e.g. /ih/ and /n/ as well. All this seems to indicate that MDE training is more powerful than our previous MCE-based training for detectors. We have not implemented MDE(H, μ) yet, but it is reasonable to expect improvements over MDE(μ), as we saw in A-MCE(H, μ) vs. A-MCE(μ).

Some phone detectors for infrequent classes did not improve their performance in the development set after MDE-training, specially with MDE(H). For those detectors, the initial BL score was considered when computing the average score. We assume this was a problem of training data availability. In addition, some detectors improved their performance with respect to BL both for the training and development sets, while the corresponding test performances decreased, e.g. this was the case for the detector of /sh/ trained with $A{-}MDE(\mu)$. It is possible that this lack of generalization could be explained by the low number of class segments in the test set for the affected detectors. For example, the number of class segments in the training, development and test sets for /sh/ are, respectively, 1466, 153 and 77. In fact, the test size of /sh/ is probably too small for the test results to be statistically significant at conventional levels. This issues were also found in [11]

In the FB optimization experiments, F-MDE(H) and A-MDE(H), the only changes in the detection structure with respect to BL was the FB matrix **H**. Therefore, the increase in performance brought by the new features can be explained by the fact that the FB in each detector was modified to extract discriminative information for the specific detection task. The FBs were clearly different from each other, specially for classes with different acoustical properties, e.g. see FBs for /ih/ and /sh/ in Figs. 2(d) and 2(f).

In our previous work we analyzed FBs optimized with A-MCE(H, μ). Most of the changes in the FBs were due to scaling of the filter amplitudes and partly also different filter shapes. However, in MDE(H) detectors we found that optimized filters had often expanded in new frequencies. Since this cannot be appreciated when all FBs are plotted together,

TABLE II
PERFORMANCE OF SELECTED DETECTORS

(a) /ih/

OPT	A_c	F	P	R
BL	45.0	57.6	77.9	45.7
F-MDE(H)	60.7	63.9	64.8	63.1
A-MDE(H)	90.5	47.6	31.5	97.8
F-MDE(μ)	64.1	69.0	70.3	67.8
A-MDE(μ)	84.1	62.1	47.4	89.8

(b) /n/

OPT	A_c	F	P	R
BL	57.7	69.5	84.7	59.0
F-MDE(H)	70.4	73.6	73.7	73.5
A-MDE(H)	74.8	70.6	63.9	79.0
F-MDE(μ)	73.5	76.4	77.5	75.3
A-MDE(μ)	80.0	72.8	64.0	84.4

(c) /sh/

OPT	A_c	F	P	R
BL	76.6	76.4	75.0	77.9
F-MDE(H)	68.8	77.5	84.6	71.4
A-MDE(H)	74.0	78.7	80.8	76.6
F-MDE(μ)	74.0	81.9	88.1	76.6
A-MDE(μ)	74.0	81.1	84.5	77.9

Fig. 2(e) isolates the 18th filter from Fig. 2(d) as an example, and the initial filter is displayed as well as a reference. We can see 1) modified shape in the initial area $(2.8, 3.5)$ kHz, 2) smaller new values near 2.5 kHz, 3) significant new shape in the interval $(3.8, 4.4)$ kHz, with maximum near 4 kHz. This means that, in contrast to standard filters, this optimized filter outputs energy information from different critical bands.

Analyzing the changes in the frequency shape of the FBs resulted in some logical conclusions. The resulting FBs for the same detector in MDE(H) had some similarities in shape when optimizing for accuracy or F-score, e.g. compare Figs. 2(b) and 2(c). Some FBs reflected properties that were found in FBs optimized with A-MCE(H, μ), e.g. the FB in /n/ had a high amplitude in the second filter probably because nasals have a low first formant, see Figs. 2(b) and 2(c). In addition, some of the changes in vowels seemed to be related to the position of the formants as well. However, some of the properties that we found previously in A-MCE(H, μ) were not present in FBs optimized with MDE(H), e.g. the shape of filters in the high frequencies in some vowel detectors was clearly different from the corresponding standard filters, see Fig. 2(d). These differences can probably be explained by 1) FBs were trained keeping the BL models, while in our previous method both FBs and models were optimized and 2) MDE differs from MCE both with respect to algorithm and performance.

V. CONCLUSIONS AND FUTURE WORK

In this paper we described a novel approach to design of subword detectors with optimized feature extractor and mod-

TABLE III
A-MCE RESULTS FROM [11]

(a) Class averaged performance after optimization.

Score	BL	A-MCE(μ)	A-MCE(H, μ)
A_c	60.7	67.0	73.7
F	67.0	68.4	66.1

(b) Performance of selected detectors.

	BL			A-MCE(H, μ)		
DET	A_c	P	R	A_c	P	R
ih	45.0	77.9	45.7	72.9	52.8	79.5
n	57.7	84.7	59.0	73.5	62.3	79.0
sh	76.6	75.0	77.9	85.7	52.6	90.9

els. We discussed the evaluation of detectors and described our Minimum Detection Error (MDE) training of MFCC filterbank and HMMs. In the experiments we built detectors of phones in continuous speech. We found that MDE-training succeeded in optimizing detectors for the chosen evaluation criteria; detectors optimized for F-score had a relative improvement of 14.8% over baseline, and accuracy optimization led to a relative improvement of 31.3% over baseline and 18.2% over an equivalent implementation with our previous MCE-based method. The optimized filterbanks were significantly different than the standard filterbank and reflected acoustic properties, e.g. formant positions, of the class to detect.

For future work, we want to study HMM-state specific filterbanks, which would model some of the short time dynamic of the signal that is relevant for the detection task. In addition, we want to design filterbanks based on phonetic knowledge of the target classes, optimize them and compare the results. Further, we will use our detectors in a pronunciation training system that focuses on vowel quality, plosive confusion, etc.

REFERENCES

[1] J. Xu, J. Liu *et al.*, "Design of the pronunciation dictionary for an English CAPT system," in *Proc. ICCDA*, vol. 4, 2010, pp. 9 –13.

[2] S. M. Siniscalchi *et al.*, "Towards bottom-up continuous phone recognition," in *Proc. ASRU*, 2007, pp. 566 –569.

[3] I. Bromberg *et al.*, "Detection-based ASR in the automatic speech attribute transcription project," in *Proc. ASRU*, 2007, p. 1829.

[4] S. Siniscalchi *et al.*, "A phonetic feature based lattice rescoring approach to LVCSR," in *Proc. ICASSP*, 2009, pp. 3865 –3868.

[5] C. Ma, "A detection-based pattern recognition framework and its applications," Ph.D. dissertation, Georgia Tech, April 2010.

[6] L. Bahl, Brown *et al.*, "Maximum mutual information estimation of hidden markov model parameters for speech recognition," in *Proc. ICASSP*, vol. 11, apr 1986, pp. 49–52.

[7] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Cambridge University Engineering Dept, 2003.

[8] B.-H. Juang, W. Hou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech and Audio Process.*, vol. 5, no. 3, pp. 257–265, 1997.

[9] X. He, L. Deng, and W. Chou, "Discriminative learning in sequential pattern recognition," *Signal Processing Magazine, IEEE*, vol. 25, no. 5, pp. 14 –36, 2008.

[10] J. Li and C.-H. Lee, "On designing and evaluating speech event detectors," in *Proc. Interspeech*, 2005, pp. 3365–3368.

[11] A. M. Canterla and M. H. Johnsen, "Optimized feature extraction and HMMs in subword detectors," accepted in Interspeech 2011.

[12] A. Biem *et al.*, "An application of discriminative feature extraction to filter-bank-based speech recognition," *IEEE Trans. Speech and Audio Proc.*, vol. 9, no. 2, pp. 96–110, feb 2001.

[13] B. Mak, Y.-C. Tam *et al.*, "Discriminative training of auditory filters of different shapes for robust speech recognition," in *Proc. ICASSP*, vol. 2, 2003, pp. 45–8.

[14] H. Bořil *et al.*, "Data-driven design of front-end filter bank for Lombard speech recognition," in *Proc. of ICSLP*, 2006, p. 381.

[15] H. Huang and J. Zhu, "Minimum phoneme error based filter bank analysis for speech recognition," in *Proc. ICME*, 2006, p. 1081.

[16] L. Burget and H. Hermansky, "Data driven design of filter bank for speech recognition," in *Proc. TSD*, 2001, pp. 299–304.

[17] Y. Suh and H. Kim, "Data-driven filter-bank-based feature extraction for speech recognition," in *Proc. SPECOM*, 2004, p. 154.

[18] D. Povey, B. Kingsbury *et al.*, "fmpe: Discriminatively trained features for speech recognition," in *Proc. ICASSP*, vol. 1, 2005, pp. 961–964.

[19] B. Zhang, S. Matsoukas *et al.*, "Discriminatively trained region dependent feature transforms for speech recognition," in *Proc. ICASSP*, vol. 1, 2006, p. I.