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 build detectors for both articulatory features and phones by discriminative training of detector-specific MFCC filterbanks and HMMs. The resulting filterbanks are clearly different from each other and reflect acoustic properties of the corresponding detection classes. For the TIMIT task, our detector-specific features reduce the average detection error rate by 20% compared to standard MFCCs.

Index Terms: Detection, discriminative training, feature extraction, filterbank, articulatory features.

1. Introduction

Detection of phonetic events such as phones and articulatory features (AFs) has applications within phonetic analysis, word spotting, computer aided pronunciation training [1] and specially in detection-based ASR (DBASR) [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 frame. The frequency content variation over classes is modeled by the class-specific MFCC-densities in the HMM-based classifier. The short time dynamic information is modeled by the MFCC time derivatives and the HMM state structure. The long time dynamic information is modeled by language model and lexicon.

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, a logical choice is to use MFCCs optimized for the specific detection problem. In principle, any of the building blocks in Fig. 1 could be improved. However, we have focused on optimizing the FB for two reasons. First, the shape of the standard FB can probably be modified to extract information that is relevant to discriminate class and anti-class. Thus, the resulting FB would reflect some of the typical frequency content of the detection class. Second, the standard FB is based on empirical experiments on human audition and it is not clear that this FB is optimal. In fact, several studies have shown that it can be optimized for ASR by data-driven techniques, e.g. [6].

The main contributions of this paper are as follows. First, we introduce a structure for detection of phonetic events in continuous speech where both the MFCC FB and the HMMs are optimized. To our knowledge this is the first experiment on FB optimization for detection (either separately or jointly with HMMs). Second, we propose a modification of embedded Minimum Classification Error (MCE) for the training of the detector structure that focuses on decreasing the error for the target class.

Previous work in data-driven FB optimization for speech recognizers includes a joint training of FB and a prototype-based distance classifier [6], optimization of robust features [7] [8], FB design using Linear Discriminant Analysis [9], minimum entropic distance [10] or minimum phoneme error [11], etc. Discriminative training of models improved detection performance in [12]. However, they found that standard MCE-training did not guarantee an error decrease for the target class and argued for a modified version of MCE-training for detection. A different approach to optimizing classification performance for a subset of classes was studied in [13].

The paper is organized as follows: Section 2 describes the structure of our detectors and discusses their evaluation. Section 3 describes the discriminative training of the FB and HMMs in each detector. Section 4 presents the detection experiments, results and analysis of the optimized FBs. Finally, Section 5 presents the conclusions and plans for future work.

2. Detectors

We focused on detectors for phonemes and AFs in continuous speech. In both cases each detector was built with a decoder based on phone HMMs. In phone detectors the class was modeled by the HMM of the phone to detect. AF detectors are commonly trained with a database transcribed at the phone level and a mapping to AFs. We chose to model the class in AF detectors with a set of HMMs in parallel: those phone HMMs where the AF feature is active. In both cases, the anti-class was modeled with all the other phone HMMs in parallel. Therefore, any detector can be regarded as a standard HMM-based ASR system optimized to improve the accuracy of a specific phone (or group of phones in the case of AF detectors).

Figure 1: MFCC feature extraction.
2.1. MFCC feature extraction
This module extracts information that provides good discrimination between class and 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 \( z \)). Then a FB performs the linear mapping \( y = H z \), where \( H \) is the FB matrix. Each component \( y_i = h_j^T z \) represents the energy in the band of the \( j \)-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 \( c \) 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 \( c \) and outputs the observation vector \( x \).

2.2. 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 subindices \( c \) 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 \( P = \text{hits}/(\text{hits} + \text{false alarms}) \) and \( R = \text{hits}/(\text{hits} + \text{misses}) \). In addition, F-score is a measure that combines precision and recall:

\[
F = \frac{2PR}{P + R} = \frac{2H_c}{2H_c + S_c + D_c + I_c + S_{ac}} .
\]

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 DBASR it is important not to lose candidates and recall is usually prioritized over precision [5]. In this paper we also prioritize hits over false alarms and, therefore, choose to use the accuracy of the detector as evaluation criterion:

\[
A_c = \frac{H_c - I_c}{N_c} = R - \frac{I_c}{N_c} ,
\]

where \( N_c = H_c + S_c + D_c \). This is the commonly used system accuracy limited to the 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 \( I_c \).

3. Discriminative training of filterbanks
Embedded MCE-training optimizes the parameters of a recognizer to improve the accuracy across all classes. We have extended the framework presented in [14, Sec. V] to optimize both the model means and the FB matrix \( H \). In this section we present first the notation and formulae, and then how the standard MCE framework has been adapted to train detectors. The interested reader is referred to [14] for further details.

There is a set of transcriptions \( \{L_j\}_{j=0}^N \), where \( L_0 \) is ground truth and the rest are given by an N-best algorithm. Each segmentation labels each frame \( x \) at the model and state level with Viterbi alignment. The performance function is a soft count of the correctly classified sentences: \( J = \sum \text{sigm}(d(V_k)) \), where \( \text{sigm} \) is the sigmoid function. The classification measure is \( d(x) = g_0(x) - \ln \left\{ \frac{1}{N} \sum_{j=1}^N \exp [g_j(x) - \eta] \right\} ^\beta \), where \( \eta \) is a positive number, and \( g_j(x) \) are the Viterbi log likelihoods of \( \{L_j\}\). For any model \( s \) and state \( t \) the state pdf \( b_t^s(x) \) is a Gaussian mixture model with mixture weights \( c_{sm} \), means \( \mu_{sm} \) and covariance matrices \( \Sigma_{sm} \). We maximize \( J \) by updating \( H \) and \( \mu_{sm} \) iteratively with the Rprop (resilient backpropagation) algorithm. Applying the chain rule of differential calculus:

\[
\frac{\partial J}{\partial \mu_{sm}} = \frac{\gamma_{ismt} \Sigma_{imt}^{-1} (x_t - \mu_{imt})}{\Sigma_{imt}^{-1} (x_t - \mu_{imt})}.
\]

\[
\frac{\partial J}{\partial \Sigma_{imt}} = \gamma_{ismt} \Sigma_{imt}^{-1} (x_t - \mu_{imt}) (x_t - \mu_{imt})^T.
\]

\[
\gamma_{ismt} = \text{sigm}^c(d(x)) c_{imt} b_{imt}(x)]^T \Sigma_{imt}^{-1} \delta_{jis} \beta_j(X),
\]

\[
\beta_j(X) = \begin{cases} 0 & \text{if } j = 0 \\ \frac{1}{\sum_{j=1}^N \exp (d(X)[\gamma_{ismt}])} & \text{otherwise.} \end{cases}
\]
where the notation of Fig. 1 is followed, “/” is elementwise division, $w_{ti} = D^T \Sigma^{-1}(\mu_c - \varepsilon_i)$, $w_{di} = D^T \Sigma^{-1}(\mu_d - d_i)$, etc. To ensure that the Fcoefficients remain positive, the parameter transformation in [14, Eq. 32] is applied.

This global training is suitable to optimize speech recognizers, i.e. applying Eqs. 3 and 4 iteratively with Rprop leads to an improved overall accuracy of the model units (phones in our case). However, in our detector MCE-based training we focus on maximizing the accuracy for the detection class, that is increasing class hits, $H_c$, and decreasing false alarms, $S_{ac}$, and $I_c$. This strategy can lead to a reduced joint accuracy of the class and anti-class, but the anti-class performance (related to $H_{ac}$, $D_{ac}$ and $I_{ac}$) is of minor importance in class specific detection. Our method is based on optimizing $\mu_{cm}$ and $H$ only with those frames that have most influence in the segmentation of the class. We have assumed that these are the frames classified as the class in any of the $N + 1$ transcriptions. The sum over all frames in Eq. 3 can be separated in two sums: 1) sum over those frames that belong to the class in at least one of the N+1 transcriptions and 2) sum over those frames that belong to the anti-class in all transcriptions. These two sums could be weighted, but we have chosen to compute Eq. 3 only with the first sum. For the same reason, Eq. 4 for phone detectors is limited to $i = c$, where $c$ is the class model, and $i = c_1, c_2, \ldots \$ (all active phone models) for AF detectors. The above results in the following steps for each iteration $n$:

1. MFCC extraction using $(H)^n$
2. N-best segmentation with present phone HMMs: $\hat{\mu}^n$.
3. Phone string alignment of $L_0$ and $L_1$ and mapping to detector labels for evaluation ($A_c$, F-score or other).
4. Viterbi state alignment of the N+1 transcriptions.
5. Compute Eq. 3 and 4 (modified for detectors).
6. Find $H = H_{c+1}^n$ and $\mu = \mu_{c+1}^n$ with Rprop.

4. Experiments

4.1. Task

Detectors with optimized FBs and model models were applied to the task of phone and AF detection on the TIMIT acoustic-phonetic continuous speech corpus. This choice TIMIT 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 with phone detectors were based on a set of 39 phonemes. The manual TIMT labeling consists of 61 acoustic-phonetic symbols. We merged plosive closures and bursts and applied the standard mapping to 39 phones. The AF detection experiments were based on a set of 20 features similar to Sound Pattern of English (SPE), e.g. vocalic, strident, nasal, syllabic, etc. The AF were derived by a mapping from a set of 56 phonemes where closures were preserved, /jh/ and /ch/ were mapped to /zh/ and /sh/ respectively, and the diphthongs were divided into a vowel and a glide.

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.

Table 1: Average detector accuracy.

<table>
<thead>
<tr>
<th>DET</th>
<th>BL</th>
<th>$(H, \mu)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH39</td>
<td>60.7</td>
<td>67.0</td>
</tr>
<tr>
<td>AF20</td>
<td>81.2</td>
<td>87.0</td>
</tr>
</tbody>
</table>

Table 2: Average F-scores.

<table>
<thead>
<tr>
<th>DET</th>
<th>BL</th>
<th>$(H, \mu)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH39</td>
<td>67.0</td>
<td>68.4</td>
</tr>
<tr>
<td>AF20</td>
<td>86.7</td>
<td>87.7</td>
</tr>
</tbody>
</table>

4.2. Experimental settings

Baseline (BL) phone detectors were built using the standard FB (see Fig. 2(a)) and a set of 39 maximum likelihood (ML) trained HMMs. For the AF detectors we used a set of 56 HMMs. In each case, 3 states and 10 mixtures monophones HMMs were trained using the phonemic transcription of TIMIT.

In the optimization, BL detectors were used as the starting point. We tried two training implementations: 1) only means were MCE-trained, referred to as $(\mu)$, 2) both means and $H$ were MCE-trained, referred to as $(H, \mu)$. In each detector, we selected the iteration corresponding to the best accuracy, $A_c$, in the development set.

In our MCE implementation the number of competing hypotheses $N$ was set to 10 and $\eta$ to 15. The HTK Toolkit was used for standard ML embedded training of HMMs. The standard FB had 26 triangular shaped filters with 201 points, while the center frequencies and bandwidths were uniformly spaced according to the Mel-scale. The grammar was a phone loop.

4.3. Results and discussion

This section presents the accuracy and FBs for the detectors. First, we discuss the accuracy test results in Table 1 for phone (PH39) and AF (AF20) detectors. The large number of detectors prevents us from presenting a thorough analysis of each case; instead we present the average detector accuracy $\bar{A}_c = \sum (N_c \cdot A_c) / \sum N_c$. In addition, Table 2 includes as a reference the average F-scores of these detectors. Note that $A_c$, and not F-score, was used as cross validation stop criterion. Second, we give general comments for the optimized FBs and focus on six detectors, whose accuracy, precision and recall are presented in Table 3 and FBs in Fig. 2. Figs. 2(b) and 2(c) show only the filters in the low frequencies to display a better resolution.

As expected, discriminative training improved the accuracy of the detectors. The $(H, \mu)$-trained detectors for PH39 resulted in a 33% relative error rate reduction relatively to BL. For AF20, the corresponding error rate was reduced by 43%. In both cases, MCE-training of means with features optimized for detection, $(H, \mu)$-training, reduced the error rate by 20% with respect to MCE-training of means with standard MFCCs, $(\mu)$-training. In addition, these improvements are superior to those in [13, Table 2], but in their method they limited the performance deterioration for the anti-class. Some phone detectors for infrequent classes did not improve their performance in the development set after MCE-training, e.g. /ng/, /uh/ and /y/ with $(H, \mu)$. We assume this was a problem of availability of training data. For those detectors, the initial (baseline) $A_c$ was considered when computing $A_c$.
Table 3: Performance of selected detectors.

<table>
<thead>
<tr>
<th>DET</th>
<th>BL</th>
<th>(H, μ)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ac</td>
<td>P</td>
</tr>
<tr>
<td>ih</td>
<td>45.0</td>
<td>77.9</td>
</tr>
<tr>
<td>n</td>
<td>57.7</td>
<td>84.7</td>
</tr>
<tr>
<td>sh</td>
<td>76.6</td>
<td>75.0</td>
</tr>
<tr>
<td>VOC</td>
<td>81.8</td>
<td>89.1</td>
</tr>
<tr>
<td>STR</td>
<td>88.8</td>
<td>91.4</td>
</tr>
</tbody>
</table>

The average F-score shows only modest improvements over BL for (μ)-training. Further, (H, μ)-training even results in an F-score reduction compared to BL. This is reasonable as increased recall usually comes at the expense of decreased precision. Note that our detector training focuses both on increasing class hits and reducing false alarms. Therefore, it is possible to use F-score in the cross-validation if we want detectors with maximal F-score.

A significant part of the increase in performance brought by the new features can probably be explained by the fact that the FB in each detector was successfully modified to extract discriminative information for the specific detection task. Note that the FBs are clearly different; this is especially noticeable for classes with different acoustical properties, e.g., see FBs of the AF detectors for vocalic and strident in Figs. 2(d) and 2(f). Most of the changes in the FBs are due to scaling of the filter amplitudes and partly also different filter shapes. The filters were, however, only rarely shifted in frequency. We currently do not have a good explanation for this.

Analyzing the changes in the frequency form of the FBs resulted in some logical conclusions. We found that some of the significant changes often occurred at relevant frequencies, e.g., formant frequencies, for both the class and main competitors. In addition, the amplitude of a filter relative to the amplitudes of the surrounding filters seemed to carry discriminative information. FBs in vowel detectors showed some similarities. First, the main changes occurred in the lower frequencies (0-4kHz), where the formants are located, and filters in the area (4-8kHz) were all attenuated. This is reasonable because it is known that formants are important for vowel classification. Second, in all vowel detectors the amplitude $A_i$ of the first five filters followed 1) $A_1 < A_2 < A_3 < A_4 > A_5$, and 2) $A_4$ was the highest amplitude in the filterbank. This behavior in the lower filters is shown in the FBs of /ih/ and the AF vocalic (respectively in Figs. 2(c) and 2(d)) and it was probably related to the position of the first formant ($F_1$). Nasals have $F_1$ in the range 250 – 300Hz, which is lower than for vowels. This was reflected in a relatively higher $A_2$ than in vowel FBs, e.g., see the FB of /n/ in Fig. 2(b). In addition, it was noticeable that the vowels with lowest $F_1$, /iy/ and /uw/, had a relatively larger $A_2$ than other vowels. Filters in the higher frequencies in nasal FBs behaved as in the vowels, i.e., they were all attenuated. The FB for /sh/ is shown in Fig. 2(e). In this case the shape of the filters in the higher frequencies changed significantly, which did not happen for vowels and nasals. This agrees with the fact that high frequencies are important to discriminate fricatives. Moreover, /sh/ is one of the phones where the AF strident is present and Fig. 2(f) shows the same effect in the high frequencies.

5. Conclusions and future work

In this paper we described a method to design phonetic event detectors with optimized feature extractor and models. We discussed the evaluation of detectors and described our detector MCE-training of MFCC filterbank and HMMs. In the experiments we built detectors of phones and articulatory features in continuous speech. We found that the optimized filterbanks were clearly different and reflected acoustic properties, e.g., formant positions, of the class to detect. In addition, MCE-training of detector-specific features and HMMs reduced the average detector error rate by 20% compared to MCE-training of HMMs using standard MFCCs, and more than 30% compared to the baseline detectors.

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.

6. References


