Craciun, Alexandra; Bäckström, Tom

Optimizing MFCC Settings for Low-Complexity VAD Systems - a Case Study

Published in: ITG-Fb. 282: Speech Communication

Published: 01/01/2018

Document Version
Peer reviewed version

Please cite the original version:

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.
Optimal temporal dynamics of MFCCs for low-complexity VAD systems – a case study

Alexandra Craciun1, Tom Bäckström2

1XMOS Ltd, UK, 2Aalto University, Department of Signal Processing and Acoustics, Finland
Email: {tom.backstrom}@aalto.fi
Web: www.aalto.fi

Abstract

Recent advances in machine learning strategies for speech classification require increasingly complex classifiers and large numbers of features. For practical application in low-resource systems, such methods use prohibitively large numbers of operations. A better approach involves reducing the features to the fewest, most salient ones, while simplifying the classifier structure to a minimum. The mel-frequency cepstral coefficients (MFCCs) are often used in speech-related classification tasks, which suggests the compressed information therein is highly informative. They are computed by warping the spectral energy to a mel scale, followed by a logarithm and a discrete cosine transformation. To better understand the properties governing such features, we examine different MFCC configurations using a simple neural network classifier for a low-complexity voice activity detector. In particular, we investigate the optimal number of MFCCs, the extent of the required temporal information and the best compression rate for different analysis settings, with varying frequency resolutions.

1 Introduction

The current trend in machine learning techniques is to switch the focus from extracting salient data features towards tuning the optimization problem on the classifier side, often by increasing the complexity of the classifier [1–3]. While such an approach succeeds in further improving the classification performance, it comes at the cost of large memory usage, high computational power or both [4]. In the current paper, we investigate the optimization problem from the perspective of a reduced-complexity feature space and a simple neural network classifier. We thus aim at finding the best tuning for the mel-frequency cepstral coefficients (MFCCs) while maintaining the classification complexity low.

Our analysis focuses on the MFCCs, since they are able to compress the discriminative information inside speech signals very efficiently. This makes them a standard feature in speech classification tasks, such as voice activity detection (VAD) [5, 6], speech recognition (SR) [1] as well as speaker and language recognition [7]. Although many authors use the MFCCs in combination with other features for classification tasks [6, 8], the current paper analyzes the MFCCs alone. This is because we aim to develop a better understanding of the MFCCs, in particular of which MFCC properties are important for each analysis configuration. We exploit such information to perform a better tuning of the MFCCs for a voice activity detector.

Our decision to investigate several analysis configurations was motivated by the fact that many applications have distinct specifications and often require an adjustment of analysis parameters, such as the frame size and overlap. In this paper we chose to restrict the analysis configurations to three commonly-used cases of 16, 32 and 64 ms frame lengths with a standard frame overlap of 50%.

While the focus of the paper lies on the choice of the optimal MFCC configuration, we also had to select a suitable classifier, which was able to perform well for the given problem, yet would keep the complexity of the whole system low. We chose a 100-node single-layer feed-forward neural network because it required a low number of operations, yet could learn elaborate enough relationships from the data and produce reliable speech/non-speech labels.

This paper investigates various aspects of the MFCCs, where the choice for each optimization procedure is based on the best results of the classifier. Often, when several choices are similar, the final decision is made for the configuration which leads to the highest system complexity reduction. The classifier is fixed, so the lowest complexity is usually achieved by reducing the length of the MFCC-based feature vector. For instance, if two MFCC numbers show similar VAD performance, the lowest one is chosen.

We examine three optimization aspects of the MFCCs. The first optimization problem involves determining the minimum number of MFCCs for each frame length which allows for the best VAD performance. The next test explores how much of the temporal information around the current frame leads to a significant VAD improvement. Here, we consider both past and future frames, although our experiments showed that the most useful information can be extracted from past frames alone. The third test involves MFCC feature compression, which is done by skipping sets of k MFCC frames. Such a test shows how the MFCC dynamics vary for different analysis settings and how much redundant information is encapsulated in subsequent frames.

2 MFCCs

The MFCC features describe a compact representation of the spectral envelope of an audio signal. They are computed by first applying a discrete Fourier transform (DFT) to a windowed segment (frame) of the time signal. Following the DFT step, the spectral energy in each frame is transformed to a mel-scale by applying a bank of overlapping band-pass filters, each corresponding to a mel band. The energy in each band is then summed up and the logarithm of the resulting mel energies is taken, followed by a discrete cosine transform (DCT). The zeroth MFCC is discarded since it is a measure of loudness and using it would bias the classifier towards sounds of different loudness, which is undesired for the current application [9].

For a low-complexity system, we need to efficiently implement each of the previously described steps of the MFCC calculation. With a fixed number of mels and known sampling frequency and DFT length, we can save the mel filterbanks into an external table and simply load them when computing the mel filters. The next step, which involves
a non-linear operation (the natural logarithm), can be replaced by a simpler approximation thereof, such as the one described in [10]. The last part of the MFCC computation, which needs to be optimized is the DCT step. We implemented an efficient DCT for multiples of 3 and powers of 2 [11, 12], which limited our choice of the number of MFCCs to 8, 16, 24, 32 and 48.

3 System evaluation

3.1 Evaluation procedure

In the following, we investigate the optimal MFCC configuration for the chosen analysis settings. The first test examines the optimal number of MFCCs for each analysis configuration. Here, only the current frame of features is used. Based on the results of the first test, we investigate how much temporal information is useful by incorporating frames of features around the current frame. This is followed by a data compression test, where we determine the amount of redundant information and eliminate frames at an optimal rate. We evaluate the performance by plotting the false-negative rate (FNR) versus the false-positive rate (FPR) in percentages as a modified ROC curve. The FPR is the ratio of the number of negative samples wrongly classified as positive to the total number of actual negative samples, while the FNR is the ratio of the number of positive samples wrongly classified as negative to the total number of positive samples. The configurations plotted include only the cases before saturation, which denotes the point where the amount of improvement is very small or almost negligible. In the following, we will describe the experimental setup, including database construction, the classifier properties and the choice of analysis parameters.

3.1.1 Database description

The experiments were performed on a set of mixtures with 11500 items, where 9200 items ($\approx 14.1h$) were used for training the classifier, while 2300 items ($\approx 3h$) were used for testing. Each mixture consisted of balanced speech and non-speech content ($\approx 50\%$ speech, $\approx 50\%$ non-speech), with a speech segment in the middle, surrounded by non-speech segments. The speech segment was mixed with either music or noise interferers in ratios of [0, 5, 10, 15, 20] dB SNR. 12 music items of approximately 15s each - half with instrumental music, half with singing voice - were manually generated from a large collection of different music genres, while 11 noise items were collected from the NOISEX-92 available online [13]. For each mixture item, a different random cut of the interferer signal was added to the speech signal in order to increase the variance of the datasets. The speech items used for testing were taken from 4 subsets (2 male and 2 female speakers) in \TEST\DR2, from TIMIT [14], where only the SX files were employed (5 files/subset). For training, we extracted speech files from 16 subsets (8 male and 8 female speakers) in \TRAIN\DR2, where only the SA and SI files were used (5 files/subset). Thus, the main difference between the training and testing datasets was the use of different speech sentences from different speakers and of different segments of the additive interferer.

3.1.2 Classifier

As a classifier, we employed a single-layer 100-nodes neural network (NN), which was implemented with the Keras package [15], using a rectified linear unit (ReLu) activation at the hidden node layer and a sigmoid activation at the output layer. We tested a few different variations of the NN, including adding more layers and both increasing and decreasing the number of nodes. The current configuration was chosen because it proved to be a good trade-off between complexity and performance and did not show signs of overfitting the speech model. The NN used the RMSprop optimizer, which is a robust adaptive learning rate method proposed by Geoff Hinton. It uses the magnitude of recent gradients to normalize the current gradients and tackles the problem of quickly decaying learning rates. As a loss function, we employed the mean squared error and trained the classifier for 5 epochs. In total, we trained 10 different models with the same settings and inputs and averaged their performance for the final result in order to get a more statistically unbiased evaluation measure.

3.1.3 Analysis settings

For the evaluation, suitable analysis settings were selected, which in this case relates to the properties of the segments the MFCCs were extracted from. Most speech applications require a frame size between 20 and 40 ms, since speech can be assumed to be stationary in this interval. We thus chose three frame sizes: 16, 32 and 64 ms, corresponding to 256, 512 and 1024 samples at 16000 Hz sampling frequency. The overlap between frames was 50% in all three cases, and we multiplied each frame by a Hann window of the frame size, followed by a discrete Fourier transform (DFT) of the same size. The MFCCs were normalised to zero mean and unit variance by using the statistics computed over the training dataset, which were also used for normalizing the features in the testing dataset.

3.2 Experiments

3.2.1 Number of MFCCs

For each of the three frame sizes, we compared the VAD performance for 8, 16, 24, 32 and 48 MFCCs extracted from the current frame. These MFCC numbers can be efficiently implemented on a low-power device. The results can be seen in Fig. 1. For all three frame sizes, we notice that an increase in the number of MFCCs results in a performance improvement. For the 16 ms window, two levels of distinct improvement can be observed: one for 16/24 MFCCs and another for 32/48 MFCCs. The performance saturates at 32 MFCCs, so 32 was chosen as the optimal number of MFCCs for the 16 ms window in the subsequent tests.

The first improvement step remains obvious also for the two larger frame sizes, yet the difference between 24 and 32 MFCCs gets smaller as the frequency resolution increases. In addition, while at 16 ms the ROC curves for 32 and 48 MFCCs are overlapping, there is a noticeable distancing between them as the window length increases (see Fig. 2). We think this is due to the fact that for shorter frames, the frequency information is already rather coarse before extracting the MFCCs, so increasing the number of MFCCs to 48 does not help much the classification. If a higher frequency resolution is available, there is more information the classifier can extract from different frequency regions, so a larger number of MFCCs improves the VAD results.

Note that the distance increases between the 32 and 48 MFCCs as the frame size increases. While the ROC lines
overlap for the 16 ms frame, they grow further apart from each other as the frame size increases. For both 32 and 48 ms frame lengths, the optimal number of MFCC is the largest available: 48.

Using the number of MFCCs identified as optimal in the first test, we analyze how much temporal information around the frame of interest is required to improve the classification performance. Since for a shorter frame the frequency resolution is coarser, we assume that the classifier will try to learn more from the available temporal information in this case and benefits from a larger number of time frames here. Indeed, Fig. 4 shows that the classifier performance requires more time frames (15) to reach saturation for shorter frames, yet it saturates already at 5 frames for the 64 ms frames. In all scenarios there is a slight improvement when using future frames. This is more noticeable for configuration 15(5) (10 past and 5 future frames) for the 16 ms and 32 ms frame lengths and for configuration 10(5) (5 past, 5 future frames) for the 64 ms frame length. We thus chose these as the optimal configurations in terms of temporal information range. Although the number of frames is the largest for the shortest frame length, the time interval they

3.2.2 Range of temporal information
Using the number of MFCCs identified as optimal in the first test, we analyze how much temporal information around

Fig. 3 shows how much improvement can be achieved by increasing the number of MFCCs from 8 to 48 for all three frame lengths. The improvement is considerable in all cases, which suggests that 8 MFCCs compresses too much the spectral information. Note that 48 MFCCs for 16 ms frames has a very similar performance as 8 MFCCs for 64 ms frames. That is, depending on the application, if the system allows for 64 ms long frames, a large reduction in the number of MFCC and thus, in the length of the feature vector, can be achieved.

Figure 1: FNR vs FPR [%] for (a) 16 [ms]; (b) 32 [ms]; (c) 64 [ms] frames; 8, 16, 24, 32, 48 MFCCs; left: zoom out; right: zoom in.

Figure 3: FNR vs FPR [%] for 8 and 48 MFCCs for all frame sizes; left: zoom out; right: zoom in.

Figure 4: FNR vs FPR [%] for (a) 16 [ms], 32 MFCCs; (b) 32 [ms], 48 MFCCs; (c) 64 [ms] frames, 48 MFCCs; multiple frames – the number in parenthesis shows the number of future frames, while the other shows the total number of frames; left: zoom out; right: zoom in.
span (128 ms) is shorter than the 64 ms frame length (352 ms).

Note there is little difference when switching from 10 to 15 and from 5 to 10 frames for the 32, respectively 64 ms frame lengths, which reduces the number of features by 33% and 50%, respectively. This, in turn, considerably scales down the complexity of the system since the feature vector is shortened and fewer multiplications and additions need to be carried out at the classifier. Also, the improvement is noticeable only for FNR rates < 5%, so arguably, if the application does not require such low FNR rates, it can benefit from reducing the number of frames.

3.2.3 MFCC compression

While there exists a lot of information in the previous frames of features, we believe that there is also a certain level of redundancy. For instance, if the classifier learns to extract speech dynamics features, i.e., features which describe variation in speech characteristics, these will change at different rates according to the frequency resolution and the length of the frames. With this in mind, we tried the simplest form of information compression, that is, we completely eliminated frames at certain intervals for each frame length. Fig. 5 shows the performance degradation due to skipping different numbers of frames. The largest amount of frames can be skipped for the shortest frame length (16 ms). Here, skipping 3 frames, i.e., using every fourth frame in all 15 available frames, shows a slight decrease in performance, but reduces the feature vector length from 480 to 60, since we are now using only 4 frames out of the total of 15. At the same time, for the 32 ms frame length, we can skip every 3 frames and use a total of only 3 out of the available 15. For the 64 ms frame length, we can only skip 1 frame, which results in using 3 frames out of 10. We thus notice a trend in each feature dynamic. The MFCC variation for the short frame can be tracked each fourth frame (every 32 ms of new data), the same holds for the 32 ms frame (every 60 ms of new data) and for the 64 ms frame we have to track every second frame (every 64 ms of new data). This suggests the need to properly tune the MFCCs since the rate of variation can have a major impact on reducing the complexity of the classifier. If we now compare the performance for the optimal setting of each frame length (Fig. 6), we observe that the best performance is obtained for the 64 ms frame. The difference between this and the other two shorter windows is however relatively small.

4 Conclusions

This paper presents an analysis of the optimal MFCC settings, in particular of the MFCC dynamics, for an efficient low-complexity VAD. As a classifier, we employed a simple feed-forward NN with 100 nodes. The input mixtures included a balanced content of speech and non-speech, with several music and noise interferers of SNRs between 0 and 20 dB. We observed that for different analysis settings, the ideal number of MFCCs which can be efficiently extracted decreases for shorter frames of data. At the same time, we showed that including a certain amount of feature frames (temporal dynamics) around the current frame significantly improves the performance of the classifier, especially for short frames with low frequency resolution. Lastly, but not least, feature compression could be achieved by simply discarding feature frames, with a high drop-out rate noticeable especially for shorter windows, for which the total number of features can be considerably reduced.

5 Acknowledgements

This work was supported by the Academy of Finland research project 312490.
References


