Speaker Identification with VoxCeleb DataSet

Yifan He, Zhang Zhang

In this project, we perform a text independent speaker identification experiment with a newly released data set, VoxCeleb (2017)[1], which consists of celebrity interview audio clips downloaded from Youtube. It's a challenging data set in the sense that there are often multiple vocal sources in the same clip. A MFCC feature vector based Deep Neural Network (DNN) is used as our baseline. It is compared with a support vector machine (SVM) algorithm. We utilzes the k-mean vector quantization technique to optimize our SVM performance, as well as filter out other vocal sources while training the SVM model.

1 Introduction

Automatic speaker recognition has been an interesting and challenging research topic for the past decade. There are two categories under this topic, speaker identification and speaker verification. Speaker verification is a widely used authentication technique and could be applied to areas such as cell phone or bank account authentication. Speaker identification determines the identity of the speaker from a pre-known speaker set. It could be applied to areas such as voice based criminal investigations or fine tuning smart devices' settings according to family member identities.

Speaker recognition algorithms can also be divided into text dependent and text independent methods. Text dependent algorithms usually involves a restricted lexicon (a set of words) and identifies the speaker from his or her lexical content. It's particularly suited for large scale commercial applications. For text independent tasks, there is no constrain on usage of words. It requires very little cooperation by the speaker and thus has a wider range of applications.

This paper will present a text independent experiment on a challenging data set. It is arranged as following: section 2 and 3 describes the data set and the feature vectors we selected. Section 4 briefly introduces our baseline DNN model and it's result will be compared with the SVM method outlined in section 5. Our result and discussion will be presented in section 6, followed possible extension of this work in section 7.

2 The VoxCeleb Data Set

Unlike most traditional data set that are collected under controlled environment, VoxCeleb directly extracts sound clips from youtube videos. This makes the data set more challenging in the sense that there are more complex background sounds and sometimes there are multiple speakers in the same clip. In the paper, authors provides a baseline prediction accuracy using CNN architectures for others to challenge, from which the best scores are top-1 accuracy 80.5 % and top-5 accuracy 92.1%.

Due to computation power limitation, we only use a subset of VoxCeleb which consists of 190 sound clips from 8 different celebrities. Also we will not use the original full clips, which are usually of several minutes length. Instead, we only use a 30 seconds section and see if we could get close to the baseline provided. This is also closer to real world speaker identification and authorization application scenarios, in which algorithms are usually required to provide a response with one or two utterances' input. On the other side, this makes this data set even more challenging. Some of the original clips are hour long interviews. During the 30 second section we selected, which is the 30 second to 1 minute session from the beginning, it may be the interviewer or some movie clips making an introduction for the speaker, who makes few to none utterance. This significantly limits our prediction accuracy and an algorithm that separates the speaker from other vocal sauces may be needed to achieve a great performance.

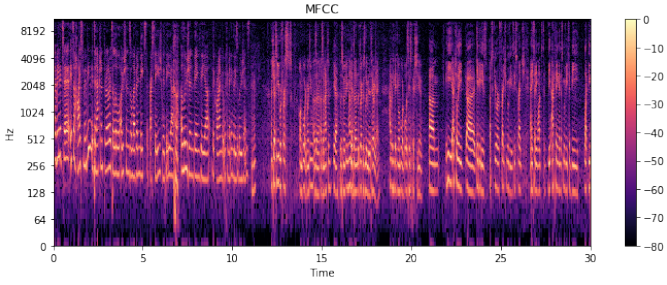

FIG. 1. MFCC vector of a sample clip.

3 MFCC Feature Vectors

The features vector selected are the Mel-frequency Cepstrum coefficients (MFCC), which are widely used features for speech signal processing. The MFCC feature extraction process, starts with frame blocking and hamming windowing of the signal, followed by a FFT transformation. Then, it transforms the frequencies to a mel scale, which has two set of filters: one spaced linearly for frequencies below 1000 Hz and one spaced logarithmically above 1000 Hz. It is based on the fact that human hearing perceptions are not sensitive to frequencies over 1Khz. The following formula can be used to compute the Mels approximately

$$Mel(f) = 2595 \times log_{10}(1 + \frac{f}{700}) \tag{3.1}$$

The final step is to transform this wrapped Mel frequency back to the time domain by a Discrete Cosine Transform (DCT). The process is summarized in Fig.2. For this experiment, we extract 20 MFCC features for each time frames that lasted 93 ms and spaced 23 ms apart.

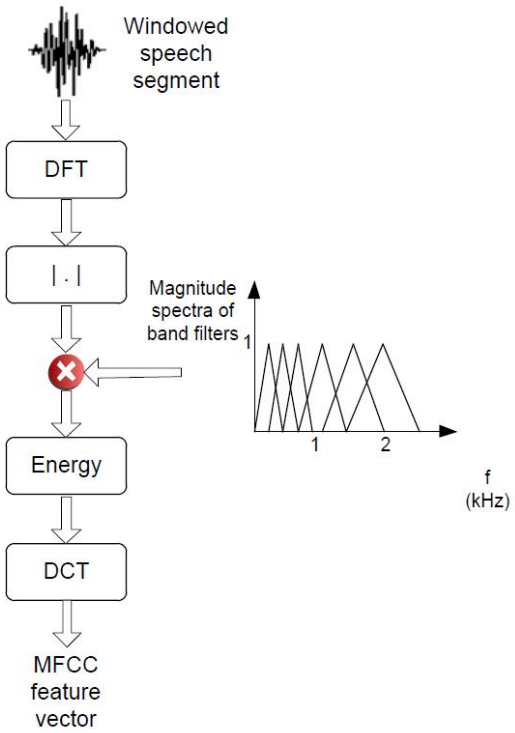

FIG. 2. MFCC compuation process [3]

4 DNN

We build our DNN baseline architecture with python TFLearn package. The model consists of 3 hidden layers of neuron numbers of 32, 32, 16 respectively, with RELU as activation function. A drop out layer of keeping rate 0.5 is added between the hidden layers and the output layer to prevent over fitting.

5 SVM

Support Vector Machines (SVM) is a commonly used supervised learning classifier due to its simplicity and efficiency. It can map its input features into higher dimension with kernel function and derive a hyperplane boundary that maximizes the margin between different classes.

For multi-class classification problem, there are two popular strategies to extend the binary SVM. The first one is one-against-all (OAA), which build k SVM classifiers for the k classes, each classifier is positive if the sample belong to that specific class and negative if not. This method is simple and fast, but asymmetric, since the negative examples set tends to be much larger than the positive one. The second method is one-against-one (OAO). It trains k(k-1)/2 classifiers, with each one distinguishes a pair of classes. The final decision will be based on a majority voting of all these classifiers.

5.1 K-mean Vector Quantization

One short coming of SVM is that its training time scales super-linearly with number of samples and it became even worse when a non-linear kernel function is applied. Standard library like *sklearn* takes extremely long time to train after the sample size exceeds several thousands. For our experiment, each sample audio clip lasts 30 s and with a 23 ms MFCC framing, each clip produces about 1000 MFCC feature vectors. This makes it difficult to train an SVM classifier directly on our MFCC features.

To solve this problem, we reduces our training sample size by performing a k-mean vector quantization (VQ) procedure first. The VQ process started with gathering all the MFCC feature vectors belonged to the same classification label in the train set. Then a k-mean clustering algorithm with $k = 150$ is performed on these vec-

Algorithm	Accuracy(%)	error(%)
DNN	43	5
SMV OAA	58	7
SMV OAO	52	8

TABLE I.

tors. Only these cluster centers will be used to train the SVM model. With this locality sensitive hashing method, we significantly reduces the train sample size for each class label from typically several tens of thousands to just 150.

In the prediction phase, all the MFCC vectors are used for each audio sample clip. The final label is decided by majority voting of all these vectors. This majority voting process together with the smoothing of the training data introduced by the VQ process should reduce some of the errors introduced by the interviewer and other vocal sources mentioned in section 2.

6 Experiment Result and Discussion

Our sample set consists of 190 audio clips from 8 different celebrities. We uses 70% of this samples as our training set and the rest as test set. 30% of the training set is further divided out as the development set to optimize the parameters and architectures. After the parameters are fixed, we perform 6 experiments for each model with different division of testing and training data to report the mean accuracy and error. The results are summarized in table I.

The DNN model resulting prediction accuracy is $43\% \pm 5\%$ on the test sets. On the training set, the accuracy is $45\% \pm 6\%$, which is comparable with the test set result. We also applied the same DNN model to sample clips of 1 minute length, which doubles the sample size of the MFCC vectors. It does not generate a detectable difference from the 30 seconds samples. We think these proves that the DNN model does not suffer over-fitting.

The SVM OAA method gives an average accuracy of 58% with a standard deviation of 7%, where as the OAO gives a result of $52\% \pm 8\%$. These accuracies are considered relatively good due to the complexity of the data set as mentioned in section 2. The prediction accuracies on training sets are about 13% higher for both methods. We think there is over-fitting but not significant. First because the difference is still within two standard error. Also, the k-mean VQ procedure should have smoothed our data in a way that reduces over-fitting. The OAA method seems to out perform the OAO by a small margin, but it's still within one standard error. When the sample size increases, the OAO method may still surpass the OAA, due to the asymmetry mentioned in section 5. The gaussian kernel function was also tested, but the performance is not as strong as the polynomial kernel.

The SVM methods out performs the DNN by a good margin. We think this demonstrates that our locality sensitive hashing process does not introduce a significant lost of information. The fact that the DNN model trained on the whole set is worse than the SVM on a reduced set should be due to the VQ process. It smoothed out the noise introduced by other sources such as the interviewer and other speakers. Therefore, VQ maybe a helpful procedure for training on multi speaker audio samples.

7 Future Work

We wish to apply the Total Variability Model (TVM) on our MFCC features and calculate the so called i-vectors. This model separates the variability from the speakers to that from the sessions and we think it particularly suits our data set and hopefully can reduce some of the errors due to other vocal sources from the environment.

We also wish to extend our sample size. We used a test sample set of only 8 different classes, which is only a very small portion of the VoxCeleb data set, due to the computation speed limit of the SVM model. With the i-vector based algorithms, hopefully we can make better use of this huge data set.

8 Contributions from Group Members

Yifan He is mainly in charge of optimizing the DNN baseline model, where as Zhang Zhang computes the SVM model and in charge of the final write up.

[1] Arsha. N., Joon S. C., Andrew Z. (2017) VoxCeleb: a large-scale speaker identification dataset. arXiv:1706.08612.

[2] Lantian Li., et al (2017) Deep Speaker Feature Learning for Text-independent Speaker Verification. arXiv:1705.03670v1.

[3] M.Azhari, C.Ergun (2011) Fast Universal Background Model Training on GPUs using Compute Unified Device Architecture. International Journal of Electrical & Computer Sciences IJECS-IJENS Vol: 11 No: 04