2013

Speaker Gender Recognition via MFCCs and SVMs

Ernest Fokoue
Rochester Institute of Technology

Zichen Ma
Rochester Institute of Technology

Speaker Gender Recognition via MFCCs and SVMs

Zichen Ma
Center for Quality and Applied Statistics
Rochester Institute of Technology
98 Lomb Memorial Drive,
Rochester, NY 14623, USA
zxm7743@rit.edu

Ernest Fokoué
Center for Quality and Applied Statistics
Rochester Institute of Technology
98 Lomb Memorial Drive,
Rochester, NY 14623, USA
ernest.fokoue@rit.edu

Technical Report - Draft 2

Abstract

An algorithm involving MFCCs and SVMs is provided to perform speaker gender recognition. For each signal, the mean vector of MFCCs matrix is used as an input vector in the SVM algorithm. A sample of 246 signals, containing 124 female voice and 122 male voice, is analyzed based on this algorithm. With only the first 13 MFCCs, the average prediction error is as low as 7% in a cross-validation of size 500. It is shown that this error drops down below 1% as the number of MFCCs increases to 27. Also, the RBF kernel is compared with polynomial kernel and considered as a better kernel function in this gender recognition task.

Keywords: *Speaker Gender Recognition, Feature Extraction, Pattern Recognition, Mel-frequency Cepstral Coefficients (MFCCs), Support Vector Machines (SVMs), Kernel Function, Cross-validation*

I. Introduction

A rather intuitive classification task in signal processing is speaker gender recognition. That is, given an input signal, the task is to classify the gender of the speakers, either female or male. A signal can be analyzed either in the time domain or in the frequency domain. Analysis in frequency domain is preferred since feature extraction, a special method of dimensionality reduction, can be performed. The computation of MFCCs is one algorithm to achieve feature extraction, which can be used together with SVMs to perform pattern recognition. In this paper we show that the combination of MFCCs and SVMs is a powerful tool in speaker gender recognition.

The second and third sections explain the computation of MFCCs and SVMs, respectively, in detail. We describe the data being used in the paper in the fourth section and perform the result and discussion in the fifth section. Finally, we draw a brief conclusion and provide some future tasks in the last section.

II. Feature Extraction via MFCCs

A voice signal in the time domain, which is simply a time series of the amplitude of the voice, is readily resulting in large number of variables. Consider a 5-second signal with a sampling rate of 8 kHz. It contains 40000 entries which transforms into the same amount of variables in order to construct the data matrix being used in pattern recognition. Analysis with large number of variables generally will lead to intense computation and overfitting.

Fortunately, high dimensionality can be reduced through algorithms of feature extraction. In terms of voice signal, such an algorithm should be different from the common algorithms like principal component analysis, since we would like the algorithm not only reduce the dimensionality, but also retain the feature of the unique voice as much as possible. MFCC is a useful algorithm of performing feature extraction for voice signal.

The main idea of MFCC is to transform the signal from time domain to frequency domain and to map the transformed signal in hertz onto Mel-scale due to the fact that 1 kHz is a threshold of humans' hearing ability. Human ears are less sensitive to sound with frequency above that threshold. The calculation of MFCCs includes the following steps:

- **Pre-emphasis filtering**
- **Take the absolute value of the short time Fourier transformation using windowing**
- **Warp to auditory frequency scale (Mel-scale)**
- **Take the discrete cosine transformation of the log-auditory-spectrum**
- **Return the first q MFCCs**

Usually in a voice segment the spectrum has more energy at lower frequencies than at higher frequencies, but the signal to noise ratio (SNR) is lower at low frequencies. Pre-emphasis filtering, a special kind of finite impulse response (FIR), can be used to compensate this problem and provide more information by boosting the energy at higher frequencies. Let $x[n]$ be the raw signal at sample n, and $s[n]$ the signal after the high-pass filtering.

$$s[n] = x[n] - \alpha x[n-1], n = 1, 2, \ldots, N \tag{1}$$

where α is a parameter controlling how much is filtered and is often chosen between 0.95 and 1 in practice. The plots below show the difference between $x[n]$ and $s[n]$ in time domain.

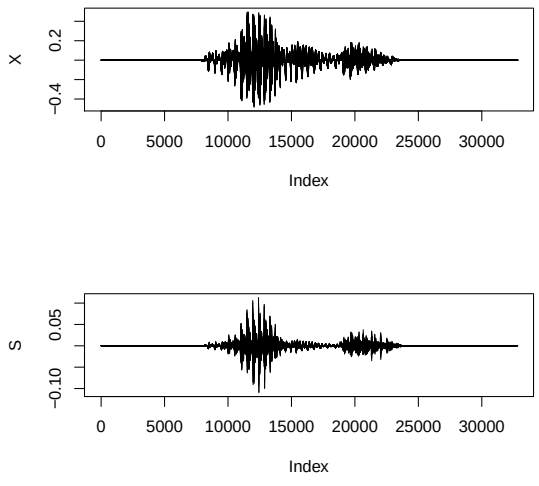

Figure 1: *$x[n]$ and $s[n]$ in time domain*

The next step is to transform the signal from time domain to frequency domain by applying short time Fourier transformation together with a window function. One assumption of Fourier transformation is that the time series is stationary, which usually does not meet the situation when the signal is relatively long. Short time Fourier transformation assumes that the signal over a very short time period is at least nearly stationary thus able to be transformed to frequency domain. This can be done by

$$X_a[k] = \sum_{n=0}^{N-1} s[n] \cdot w_a[n] \cdot e^{-i2\pi kn/N} = \sum_{n=0}^{N-1} s[n] \cdot w_a[n] \cdot e^{-i\omega k}, 0 \leq k < N \tag{2}$$

where $w_a[n]$ is the window function, which is a zero valued function everywhere except inside the window m, and i is the imaginary unit. Usually to keep the frames continuous, a Hamming window

$$w_a[n] = \alpha - \beta \cos(\frac{2\pi n}{N-1}), 0 \leq n < N, \alpha = 0.54, \beta = 1 - \alpha = 0.46 \tag{3}$$

is preferred and the length of each frame is kept between 20 to 40 ms. The plot given below demonstrates the effect of windowing with a frame length of 40 ms.

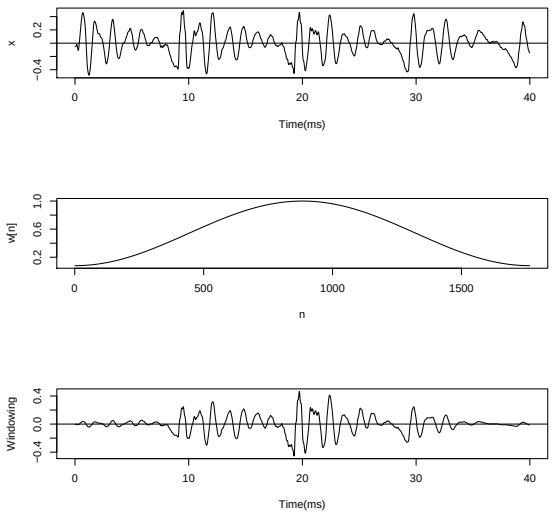

Figure 2: *Effect of a Hamming window*

A fact of human hearing ability is that we are more sensitive to sound between 20 and 1000 Hz. Thus it is less efficient to assign a signal the same scale at high frequencies as at lower frequencies. An adjustment can be made by mapping the data from Hertz-scale onto Mel-scale:

$$mel = \begin{cases} f & f \leq 1000 \\ 2595 \log_{10}\left(1 + \frac{f}{700}\right) & f > 1000 \end{cases} \tag{4}$$

And its inverse is given by

$$f = \begin{cases} mel & mel \leq 1000 \\ 700\left(e^{\frac{mel}{2595}} - 1\right) & mel > 1000 \end{cases} \tag{5}$$

A corresponding plot of this warping function is given below.

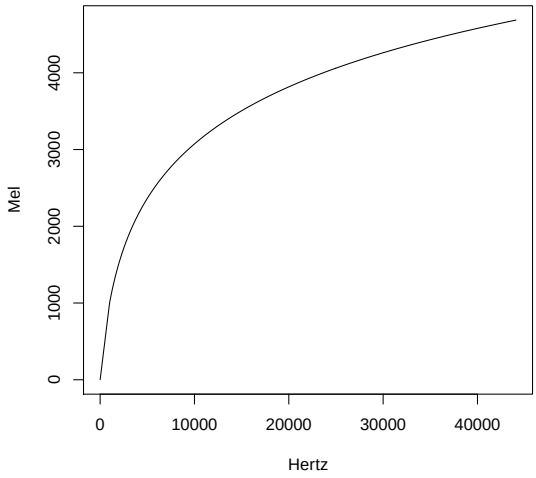

Figure 3: *Relationship between Hertz-scale and Mel-scale*

Notice that when it covers a wide range on Hertz scale at high frequencies, it transforms onto Mel scale a much narrower range.

Given the STFT of a input window frame $x_a[k]$, we define a filterbank with M filters $(m = 1, 2, , M)$ that are linear on Mel scale but nonlinear on Hertz scale, where m is triangular filter given by

$$M_m[k] = 1 - \left| \frac{k - \frac{N-1}{2}}{\frac{N-1}{2}} \right|, 0 \leq k < N \tag{6}$$

where N is the length of the filter. Notice again that these filters are linear on Mel scale and they need to be transformed back to Hertz scale. Thus we can then compute the log-energy of each filter as

$$S[m] = \ln\left[\sum_{k=0}^{N-1} |x_a[k]|^2 M_m[k]\right], 0 < m \leq M \tag{7}$$

The Mel-frequency cepstrum coefficients are then the discrete cosine transform of the M filter outputs:

$$c[q] = \sum_{m=0}^{M-1} \left[S[m] \cos\left(\frac{\pi q (m - \frac{1}{2})}{M} \right) \right], 0 < m \leq M \tag{8}$$

In practice, M is usually chosen between 24 and 40 and the first 13 MFCCs are computed. Also notice that for each signal, the MFCCs actually form a $n \times q$ matrix where n is the number of window frames and q is the number of MFCCs. If we are to pass the MFCC matrices to a vector-based pattern recognition technique, these matrices have to be transformed or summarized to vectors. The simplest way of doing this is to take the mean values of each of the n column vectors.

III. Pattern Recognition via SVMs

The main idea of SVMs is to define a boundary between two classes by maximal separation of the closest observations. In practice, SVMs are powerful algorithm on binary classification tasks. The main idea of SVMs is shown in the graph below.

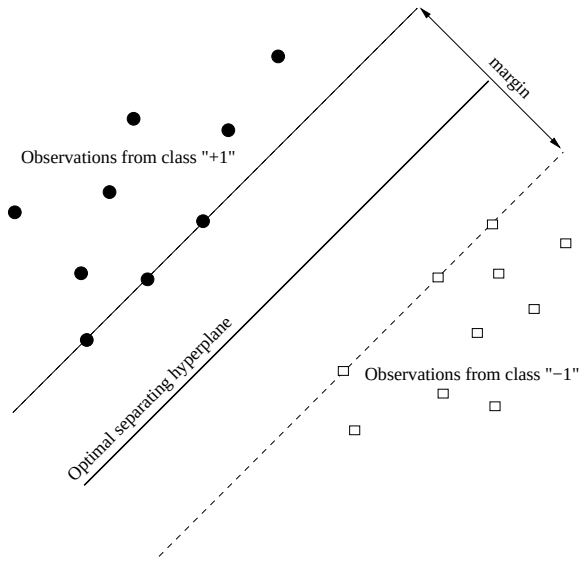

Figure 4: *The goal for SVMs is to maximize the margin.*

Consider a dataset $D = \{(\mathbf{x_i}, y_i), \mathbf{x_i} \in \mathbb{R}^p, y_i \in \{-1, 1\}\}_{i=1}^n$ that is linearly separable. The margin is defined as the shortest perpendicular distance between the hyperplane and the observations, referring to the width of the blank region separating two data clouds. The goal for SVMs is to maximize the margin. Any hyperplane can be written as

$$\mathbf{w} \cdot \mathbf{x} - b = 0$$

where w is a coefficient vector and b is constant.When the data are linearly separable, there exist two hyperplanes that can separate the data completely and no points fall in between. Such hyperplanes can be defined as

$$\mathbf{w} \cdot \mathbf{x} - b = -1$$

and

$$\mathbf{w} \cdot \mathbf{x} - b = 1$$

The region in between is the margin. It can be shown that maximizing the margin is equivalent to minimizing $\| \mathbf{w} \|$. Finally, the classification can be achieved by

$$class(\mathbf{x_i}) = \begin{cases} 1 & \mathbf{w} \cdot \mathbf{x_i} - b > 0 \\ -1 & \mathbf{w} \cdot \mathbf{x_i} - b \leq 0 \end{cases} \tag{9}$$

When the data is not linearly separable, kernel functions play an important role by linearizing the data. That is, when the data is not linearly separable, they can be transformed by a kernel function $K(\mathbf{x}, \mathbf{y})$ into the inner product space in which it is feasible to separate them linearly. Common kernel functions are the radial basis function kernel (RBF kernel)

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\| \mathbf{x} - \mathbf{y} \|_2^2}{2\sigma^2}\right) \tag{10}$$

and the polynomial kernel

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + c)^d . \tag{11}$$

IV. Data Description

A total of 246 signal data were collected from the voice of 8 speakers, 4 female and 4 male, from an internet resource. Multiple two-syllable words were assigned to each voice and a demographic summary is given by the contingency table below.

Table 1: *A demographic summary of speakers*

Accent	Gender		
	Female	Male	Total
British	65	57	122
American	59	65	124
Total	124	122	246

Though the sound tracks have lengths of only around 1 second, with a sampling rate of 44100 Hz, each sound track vector on the time domain has more than 30000 entries. The response is given by

$$y_i = \begin{cases} 0 & \textbf{if male,} \\ 1 & \textbf{if female.} \end{cases} \tag{12}$$

V. Result and Discussion

Before any formal analysis was done, two spectrograms were created, one for a male voice and the other for a female voice, to visualize the difference between genders.

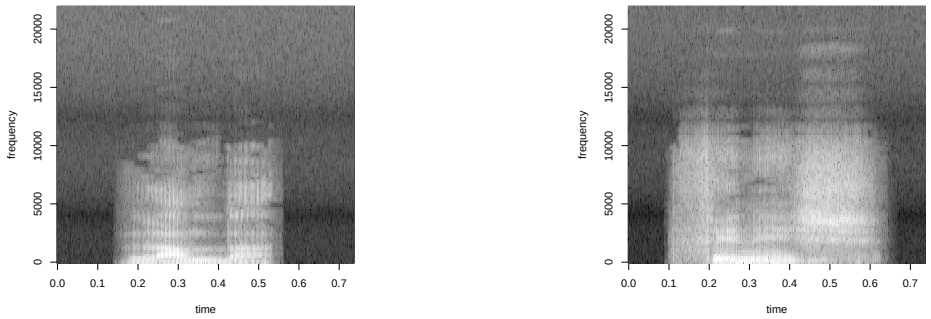

Figure 5: *Spectrograms for male voice (left) and female voice (right)*

Though it is relatively difficult to distinguish the difference directly from the spectrogram, the plots still give an idea that female voice exhibits more at higher frequencies than male voice.

Then the MFCCs were computed for each sound track and the mean vectors were passed to support vector machines. Both RBF and 2nd degree polynomial kernels were used and then were compared to each other. In order to approximate the true prediction ability of the model, a hold-out cross-validation of size 500 was done based on stratified random sampling. The precision for each prediction is simply the ratio between the correct prediction, which is the summation of true positive (TP) and true negative (TN), and the total number of sound tracks (N). And the overall prediction error (ERR) is the average prediction error of the cross-validation of size 500.

$$ERR = \frac{\sum\{1 - \frac{TP+TN}{N}\}}{500} \tag{13}$$

A summary result is given in the table below.

Table 2: *A summary of the results*

# MFCCs	ERR(PBF)	ERR(Poly)
12	0.0748	0.0549
19	0.0232	0.0418
26	0.0072	0.0131
34	0.0023	0.0051
40	0.0019	0.0063

Notice that when the number of MFCCs increases from 12 to 26, the average prediction error drops from 7.5% **to** 0.7% **for RBF kernel, but not as much from 26 to 40. It is not of surprising since the feature information provided by MFCCs is richer as the number of MFCCs increases.**

But as the dimensionality keeps increasing, it would have an effect on the complexity of computation. Also notice that the model with 2nd degree polynomial kernel is very close to the one with RBF kernel. Balancing the prediction accuracy and the computational complexity, it seems that SVM with RBF kernel and 26 MFCCs is the best one for prediction, though the other models also generate results with high accuracy.

VI. Conclusion

We have demonstrated in this paper that the combination of MFCCs and SVMs is a powerful tool in recognizing speaker gender. Both RBF kernel and polynomial kernel provided decent results in cross-validation. Due to computational complexity, that is, the computation of more MFCCs require more computational time, we chose $26(= 27 - 1)$ MFCCs in this case, but the number of MFCCs can also be determined by the task and the expected or acceptable accuracy. Usually it is not necessary to use large number of coefficients.

In this paper we only considered the mean vectors of MFCCs matrices for simplicity, but alternative methods can be taken into account to generate the input vectors in SVMs. For instance, the standard deviations of each MFCCs can be used together with the mean values, or each coefficient can be modelled as a Gaussian mixture. Also, feature extraction via MFCCs is not as powerful when the signal contains significant noise since MFCCs provide detailed information of the raw signal. In such cases, alternative algorithms, usually much more complex, should be preferred rather than MFCCs.

References

Chen, S.-H. and Y.-R. Luo (2009). Speaker verification using mfcc and support vector machine. *Proceedings of the International MultiConference of Engineers and Computer Scientists 1*.

Clarke, B., E. Fokoue, and H. Zhang (2009). *Principles and Theory for Data Mining and Machine Learning*. Springer.

Clarkson, P. and P. Moreno (2009). On the use of support vector machines for phonetic classification. *Proc. ICASSP*, 585–588.

Dhanalakshmi, P., S. Palanivel, and V. Ramalingam (2008). Classification of audio signals using svm and rbfnn. *Expert Systems with Applications*.

Gaikwad, S., B. Gawali, and S. Mehrotra (2012). Gender identification using svm with combination of mfcc. *Advances in Computational Research 4*.

Huang, X., A. Acero, and H.-W. Hon (2001). *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Upper Saddle River, New Jersey: Prentice Hall PTR.

Ittichaichareon, C., S. Suksri, and T. Yingthawornsuk (2012). Speech recognition using mfcc. *International Conference on Computer Graphics, Simulation and Modeling*.

Khan, A., M. Farhan, and A. Ali (2011). Speech recognition: Increasing efficiency of support vector machines. *International Journal of Computer Applications 35*(7).

Rabiner, L. and B.-H. Juang (1993). *Fundamental of Speech Recognition*. Englewood Cliffs, N.J.: Prentice-Hall.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. New York: Springer-Verlag.

Zheng, F., G. Zhang, and Z. Song (2001, 11). Comparison of different implementations of mfcc. *Journal of Computer Science and Technology 16***(6).**