International Journal of Applied Science & Environmental Engineering – IJASET 2018
Copyright © Institute of Research Engineers and Doctors, SEEK Digital Library
Volume 1 : Issue 1 Publication Date : 28 December, 2018

Performance Evaluation of Machine Learning Algorithm Applied to a Biometric Voice Recognition System

Andrea L Piroddi

Abstract— The article investigates the possibilities of applying machine learning algorithm to identify an individual through biometric voice recognition with the higher possible reliability. The emphasis in the analysis is placed on the possibility of using artificial intelligence approach methods for the purposes of recognizing a person unambiguously, uniquely on the basis of the data contained in his/her vocal spectral information. A large number of routes we can go to parametrically representing the speech signal for the voice recognition system such as Mel-Frequency Cepstrum Coefficients (MFCC). During the authentication phase the input voice signal is recorded and processed comparing it by using MFCC features with a signal that has been previously stored in the database by the same user. The main purpose is to compare some of the main machine learning algorithms to classify them on this particular application

Keywords—Biometric, Mel-Frequency Cepstrum Coefficients (MFCC), Voice Recognition, Weka, Open smile, Praat

I. Introduction

The recognition of an individual through biometric parameters is the basis for many technological applications starting with the protection of personal data systems [3] to get to robotic systems that obey to voice commands to perform the most varied tasks, to voice pathology monitoring [2] using the integration of the IoT and the cloud. A biometric technology is the one which use the user features parameter as the identification discriminant element. Imagine a robot that is interacting with the people and must identify each person, without the support of a cam, to be able to retrive data from its database, in order to respond appropriately to any questions or vice versa ask questions meaningful to each party, a bit like we do when we are on the phone with our friends or our colleagues. So the question is: what is the machine learning algorithm that best suits this purpose? From an engineering point of view there are two types of Automatic Speech Recognition approach: Direct Voice Input (DVI) and Large Vocabulary Continuous Speech Recognition (LVCSR). These systems analyze user's specific voice and use it to fine tune the recognition of that user's speech. Both work use a mechanism based on two actions: the extraction action and the comparison action. The first one educes a small amount of data from the speech signal used to represent uniquely the speaker, the second one compares the extracted features of a utterance[1] with the ones from a set of known users to match which of these is the most suitable sample and then the most likely user. The speech signal and its characteristics can be represented in two different domains which are time and frequency domain.

Andrea L. Piroddi
University of The People
Pasadena – California - USA

II. Preliminary Observations

Like by fingerprint, human being can be recognized by his unique tone, rhythm, frequency and pitch. The average male has a lower voice than the average female but the average range of each person's voice is unique. A lot of studies have been done about this particular issue. One of the most important is the Mel Frequency Analysis [4]. Normally we can represent speech signal (time dependent), as a sequence of spectral vectors with the support of the FFT (Fast Fourier Transform). In a graphical representation we can map spectral amplitude to a grey level (0-255) value, where 0 represents black and 255 represents white. In this way, higher the amplitude, darker the corresponding region, see Figure 1. This is a Time versus Frequency representation of a speech, that is a Spectrogram. Now in Figure 2 we show the spectrogram of the speech signal of a female 16 years old that is pronouncing the phrase "Phone Six". In the Speech Spectrum peaks denote dominant frequency components in the voice signal, so they are referred to as Formants [8]. Formants carry the identity of the sound. Now if we connect all the peaks with a smooth curve we get the Spectral Envelope. (see Figure 3).

We can see that Spectrum (*logX[k]*) is composed by Spectral Envelope (*logH[k]*) and by Spectral Details (*logE[k]*) see Figure 4. That is

$$\log X[k] = \log H[k] + \log E[k] \qquad (1)$$

Now we would like to sunder the spectral details and the envelope from the spectrum. That is the information we got is the Spectrum *logX[k]* and so we'll try to obtain *logH[k] and logE[k]*. So just applying IFFT to the Spectrum we'll obtain a component at low frequency that correspond to the Spectral Envelop and a component at higher frequency that correspond

[1] Utterance is the vocalization of a word or words that represent a single meaning to computer.

to Spectral details. Roughly the frequency component of the envelope gives a peak at about 10Hz in the pseudo-frequency axis, instead the frequency component of the spectrum details is about 100Hz. So using this mathematical ploy we have obtained that

$$x[k] = h[k] + e[k] \qquad (2)$$

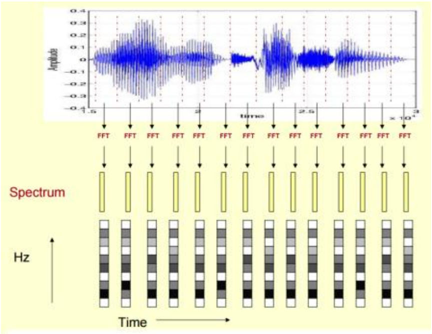


Figure 1 Speech signal represented as a sequence of spectral vectors

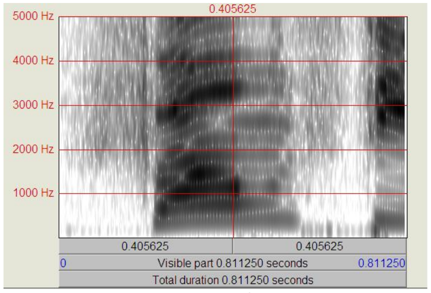


Figure 2 Spectrogram of the voice signal

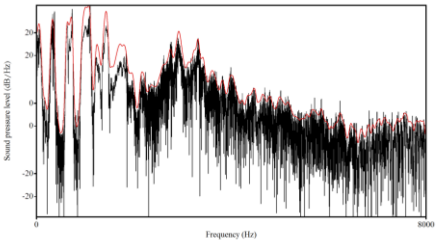


Figure 3 Spectral Envelope (red line)

Mel-Frequency analysis of speech is based on human perception experiments [1]. It has been found that human auditory apparatus is a filter. Just certain frequencies are receipt and are not uniformly distributed on the frequency axis. We know that low frequencies give the most important contribution on the global power, so there will be more filters in the low frequency regions and less in the higher regions. So Mel-Frequency Cepstral Coefficients (MFCC) are the coefficients that better represent the h[k] component. These Cepstral vectors are given to pattern classifiers for speech recognition purpose. Now, the idea is to generate different speech samples from different people so to have the basic material from which elicit the cepstral data and then to test some kind of machine learning approach to classify which is the best performer, if any, on this particular application.

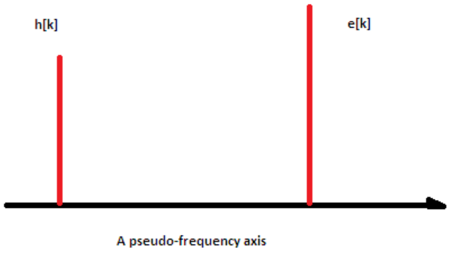


Figure 4 x[k]=h[k]+e[k]

III. Methodology

We put together four people of different age and sex, 2 males, respectively 18 and 45 years old and 2 females, respectively 16 and 45 years old. We have defined just 23 different kind of brief phrases such as “Phone Steve”, “Dial One”, “Call Bob” and we have recorded 23 identical sentences from each one of the candidates. The most important elements are the sampling rate fixed at 16KHz, sample period, window size and the pre-emphasis. These parameters, together, uniquely define the recording interval so that all the samples have the same duration. Once we have collected all the 92 speech samples, we analyze some of those, with the Praat application software [7]. This powerful tool allows us to manage and compare graphically the speech signal both in the time and frequency domain. For example we have compared the same sentence “Phone Bob” uttered by the young lady and the 45 years old male in the time domain, see **Figure 5**, and in the frequency domain, see **Figure 6**. It’s easy to note in **Figure 6** the different spectral components. At this moment we need a software application that allow us to extract in a more readable form the Mel-Frequency Cepstral Coefficients (MFCC) and some other parameters from all 92 tracks we have collected in order to have the possibility to feed different machine learning algorithm and classify the results. Now we have to introduce the Mel Scale. It refers to perceived frequency of a pure tone to its actual measured frequency. As we have seen, human beings have a very refined hearing system that is able to distinguish small changes in pitch at low frequencies than it is at high frequencies. Incorporating this scale makes our features match more closely what humans hear. The relation between frequency and Mel scale is given by eq.(3) :

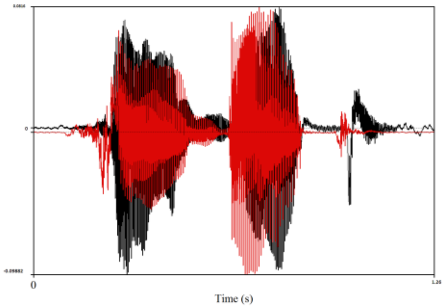


Figure 5 Sentence "Phone Bob" uttered by 16 years old female (black line) and 45 years old male (red line) in time domain

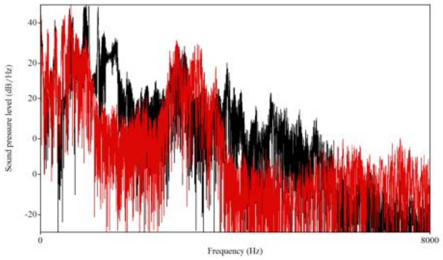


Figure 6 Sentence "Phone Bob" uttered by 16 years old female (black line) and 45 years old male (red line) in freq domain

$$M(f) = 1125\ln\left(1 + \frac{f}{700}\right) \qquad (3)$$

The inverse relation is given by eq.(4):

$$M^{-1}(m) = 700\left(e^{\left(\frac{m}{1125}\right)} - 1\right) \qquad (4)$$

Framing the speech signal into 25 ms frames, the frame length for a 16kHz sampled signal is 0.025*16000 = 400 samples. The next steps are applied to every single frame, one set of 12 MFCC coefficients is extracted for each frame. That is, we call our time domain signal. After framing, we get $s_i(n)$ where n ranges from 0 to 400 (because our frames are 400) and i ranges over the number of frames. Proceeding with calculation of the DFT we get $S_i(k)$ where i identify the number of the frame in the domain of time. Defining $P_i(k)$ the Power Spectrum of the *ith* frame, we have that the DFT is given by eq.(5):

$$S_i(k) = \sum_{n=1}^{N} s_i(n)h(n)e^{\frac{-j2\pi kn}{N}} \qquad 1 \le k \le N \qquad (5)$$

Where h(n) is the hamming window with N sample length, and K is the length of the DTF. So the power spectral evaluation is given by eq.(6):

$$P_i(k) = \frac{1}{N}|S_i(k)|^2 \qquad (6)$$

We take the absolute value of the complex DFT, and square the result. Generally a standard approach performs a 512 points FFT and keep only the first 257 coefficients. Compute the Mel-spaced filterbank. This is a set of 20-40 (26 is standard) triangular filters that is applied to the periodogram power spectral estimate. Just to clarify the filterbanck is obtained in the following way:

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \cup k > f(m+1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \le k \le f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \le k \le f(m+1) \end{cases} \qquad (6)$$

Where M is the number of the filters we want and $f(m)$ is the list of $(M+2)$ Mel-spaced frequencies. Our filterbank comes in the form of 26 vectors of length 257. Each vector is mostly zeros. To calculate filterbank energies each of them is multiplied by the power spectrum, then coefficients are summed up. Once this is performed we are left with 26 numbers that give us an indication of how much energy was in each filterbank. If we calculate the logarithm of each of the 26 energies and process the Discrete Cosine Transform (DCT) of the 26 log filterbank energies we obtain the 26 cepstral coefficients, see eq.(8). For Automatic Speech Recognition, only the lower 12-13 of the 26 coefficients are kept. Because they are the most significant. The DCT in one dimension, for a sequence of length N, is defined as:

$$C(u) = \alpha(u)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{\pi(2x+1)u}{2N}\right] \qquad for\ u = 0,1,\dots,N-1 \cup \alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & if\ u = 0 \\ \sqrt{\frac{2}{N}} & if\ u \neq 0 \end{cases} \qquad (7)$$

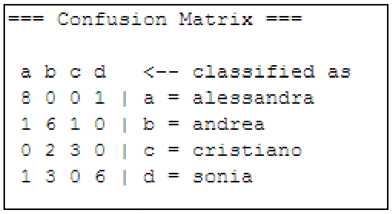
To obtain the MFCC coefficients and some other important parameters we used the application OPENSMILE (ver. 2.1.0 for Linux platform). To facilitate interoperability, OpenSMILE supports reading and writing of various data formats commonly used in the field of data mining and machine learning. These formats include PCM WAVE for audio files, CSV (Comma Separated Value, spreadsheet format) and ARFF (Weka Data Mining) for text-based data files, HTK (Hidden-Markov Toolkit) parameter files, and a simple binary oat matrix format for binary feature data [5].

The next step was to take all the 92 ".wav" files and give them to feed to Opensmile, by bringing together the various parameters extracted in a single file with ".arff" extension (in our case "*voicerecogn.arff*"), which is the typical format accepted by artificial intelligence software WEKA . Each parameters extraction has been labeled with the name of the corresponding volunteer. Opensmile also has a configuration file (in our case *IS09_pir.conf*, customized for this particular research) in which you tell the software the main parameters that you want to extract from the audio file. The result of this extraction is a file with extension ".arff" in which we can find the list of all the name of the attributes OpenSmile extracted from the ".wav" files, and to follow in rows, all the data corresponding to those attributes for each wav file labeled with the name of the volunteer. Since openSMILE is used by the openEAR project for emotion recognition, various standard feature sets for emotion recognition are available as openSMILE configuration files. The INTERSPEECH 2009 Emotion Challenge feature set is represented by the configuration file config/IS09.conf. The IS09_pir.conf is a customization of the latter for this specific research. It contains 384 features as statistical functionals applied to low-level descriptor contours. So far, we have collected a large number of attributes for each speech sample. Now we want to look for a way to identify if there is a way to classify this dataset in order to uniquely pick out the owner of the voice sample by using some artificial intelligence algorithm. To achieve our purpose we propose the use of WEKA [6]. Weka is a machine learning/data mining software written in Java (distributed under the GNU Public License) used for research. Furthermore Weka only deals with “flat” files like arff format. That’s why we used Opensmile to obtain the final file “*voicerecogn.arff*”. The voice recognition in Machine Learning approach is divided into two phases which are training phase and testing phase. So the first thing we did was to provide the file “*voicerecogn*.arff” to Weka Tool applying the unsupervised filter on attribute for removing type. This way Weka is able to group the information provided by the input file according to the class each data belong to. The following step was the final one, that is using of the classifier, that is a models for predicting nominal or numeric quantities, to understand if there is a machine learning algorithm suitable to identify unique attributes that allows the speaker to be identified.

IV. Result

Different experiments were carried out to analyze the performance of the voice recognition system. All the experiments provided for the division of samples into 66 percent for the training phase and 34 percent for the testing phase. The first classifier we proposed is Random Tree. A random tree is a tree that is formed by a stochastic process. Once the classifier, test options and class have all been set, the learning process begins. At the end of the learning process, a lot of data are available, the most important are those enclosed in the summary reported by the tool. In the summary report we can find a list of statistics summarizing how accurately the classifier was able to predict the true class of the instances under the chosen test mode. The kappa statistic measures the agreement of prediction with the true class –1.0 signifies complete agreement.

Furthermore, Weka tool provides the “Detailed Accuracy By Class” that is a more detailed per-class break down of the classifier’s prediction accuracy. The True Positive (TP) rate is the proportion of examples which were classified as class **a**, among all examples which truly have class **a**, i.e. how much part of the class was captured. It is equivalent to Recall. In the confusion matrix, this is the diagonal element divided by the sum over the relevant row, i.e. 8/(8+1)=0.889 for class yes in our experiment. The False Positive (FP) rate is the proportion of examples which were classified as class **a**, but belong to a different class, among all examples which are not of class **a**. In the matrix, this is the column sum of class **a** minus the diagonal element, divided by the rows sums of all other classes; i.e. (10-8)/(8+5+10)=0.087 for class yes. Finally, the tool includes the Confusion Matrix, that shows how many instances have been assigned to each class. Elements show the number of test examples whose actual class is the row and whose predicted class is the column, see **Figura 7**.



```
=== Confusion Matrix ===

 a b c d   <-- classified as
 8 0 0 1 | a = alessandra
 1 6 1 0 | b = andrea
 0 2 3 0 | c = cristiano
 1 3 0 6 | d = sonia
```

Figura 7 Confusion Matrix Random Tree

The confusion matrix is also known as Contingency Table. In our experiment we have four classes, the four speakers, and therefore a 4x4 confusion matrix. The number of correctly classified instances is the sum of diagonals in the matrix; all others are incorrectly classified, for example class ”**a**” gets misclassified as ”**b**” exactly 0, and class ”**b**” gets misclassified as ”**a**” one time). The second classifier we tested was the J48 Pruned Tree. The J48 Decision tree classifier primarily needs to make a decision tree founded on the values of the attribute of the training data. So, when it faces the training set it searches for the attribute that better discriminates the various instances. This feature helps to easily classify the data instances and so it is said to have the highest information gain. One of the most important aspect of J48 is that, if there is any value for which there is no ambiguity, that is, for which the data instances falling within its category have the same value for the target variable, then the algorithm terminates that branch and assigns to it the target value that it has obtained. With this algorithm we have slightly improved the performance, in fact we have passed from 71.875% of Correctly Classified Instances to an 81.25%. Next algorithm we tried was Random Forest. Random Forest is a method of combining multiple Random Trees (thus - Forest) into one big classifier using even more randomization.

With Random Forest we have reached a very important target, that is a 96.875% of Correctly Classified Instances, with just an increment of the time taken to build the model that is passed from 0.08 seconds of the Random Tree to 0.81 seconds of Random Forest. Obviously this latter value is small in the absolute sense but the percentage growth compared to the time used by other systems is 1 order of magnitude greater, this fact becomes critical when the number of samples set becomes very large. So far, we have used only tree based algorithm, so the idea was to try a different approach based on Logistic function. The Logistic Function is a Probability Function and is easy to deal with mathematically. It is used widely in neural networks because it can be generalized to handle multiple continuous parents by taking a linear combination of the parent values producing a soft threshold, see eq.(9).

$$P(y|c) = \frac{1}{1+e^{-2\frac{-c+\mu}{\sigma}}} \qquad (9)$$

This approach is able to muster 100% Correctly Classified Instances. This is an excellent result except for the high value of time taken to build the model that is about 23.6 seconds, i.e. three order of magnitude greater than Random tree. Obviously we must say that we did not introduce noise in this study since the goal was to identify the most suitable algorithm for recognition. But it is true that when a user pronounces a word unless he moves into an anecoic room, the input signal to our system will include environmental noise, which justifies the reason why in this experiment we can reach 100% of recognized samples. It's interesting to note that the resulting confusion matrix is a diagonal matrix. So to verify if exists an algorithm with the same classification capability but with a model building time more similar to that of Random Tree, we tried the Naïve Bayesian. The Bayesian Classifier is well suited when a single cause directly influences a number of effects, all of which are conditionally independent, given the cause. The distribution can be written as showed in eq.(10).

$$P(Cause, Effect_1, \ldots.., Effect_n) = P(Cause) \prod_i P(Effect_i | Cause) \quad (10)$$

With this algorithm we have reached the 100% of Correctly Classified Instances and the time taken to build the model is about 0.15 seconds, that is the quite similar to those of the J48 Pruned Tree. The last algorithm we tried is the Multilayer Perceptron. In [9] is showed that a single threshold unit would not solve all kind of problem. It is demonstrated that such a unit can represent the basic Boolean functions AND, OR, and NOT and then it derives that connecting large number of units into networks of arbitrarily depth we can obtain any desired functionality. With this algorithm we have reached the 100% of Correctly Classified Instances but the time taken to build the model is about 536 seconds, that is about 9 minutes.

Table 1 shows the result for accuracy of voice recognition system. The best performer on all the most important parameters is the Naïve Bayesian classifier. This means that the Naïve Bayesian is able to recognize the speaker with an accuracy of 100% with a relative absolute error of 0,255% and a model building time of about 0.15 seconds.

Classifier	Correctly Classified Instances	Relative Absolute Error	Time Taken to build the model in sec
Random Tree	71,8750%	37,2137%	0,08
J48 Pruned Tree	81,2500%	28,0066%	0,27
Random Forest	96,8750%	54,3321%	0,81
Logistic function	100,0000%	1,6612%	23,6
Naïve Bayesian	**100,0000%**	**0,2550%**	**0,15**
Multilayer Perceptron	100,0000%	7,1524%	535,89

Table 1 Classifier Performance Comparison

V. Future Work

Next target is to investigate how the performance of algorithms tested in this study changes, as the number of samples, that is, the number of "users to recognize" grows significantly. Another important step is to investigate which subsets of attributes (pcm_RMSenergy_sma_amean, F0_sma_linregerrQ, …) are the most predictive ones. Using Weka panel for attribute selection we can explore this field.

References

[1] Prahallad K., Speech Technology: A Practical Introduction. Carnegie Mellon University & International Institute of Information Technology Hyderabad- 2011; p.3-49

[2] Muhammad G., Mizanur Rahaman S.M., Lelaiwi A., and Alamri A., Smart Health Solution Integrating IoT and Cloud: A Case Study of Voice Pathology Monitoring, IEEE Communications Magazine January 2017, p.69-73

[3] Mohd. Shah H.N., Ab Rashid M.Z., Fairus Abdollah M., Kamaraudin M.N., Kok Lin C., and Kamis Z., Biometric Voice Recognition in Security System, Indian Journal of Science and Technology, February 2014, Vol 7(2), p.104-112

[4] Schafer R., Rabiner L., Theory and Applications of Digital Speech Processing ASIN/ISBN: 978-0136034285A comprehensive guide to anything you want to know about speech processing

[5] Eyben F., Woellmer M., Schuller B., Opensmile the Munich open Speech and Music Interpretation by Large Space Extraction toolkit, Version 1.0.1, May 23rd 2010

[6] Bouckaert R.R., Frank E., Kirkbly R., Reutemann P., Seewald A., Scuse D., WEKA Manual for Version 3-7-13, The University of Waikato

[7] Gut U., Praat Manual, University of Augsburg – Linguistic Laboratory, January 2008

[8] Kawahara T., Lee A., Multipurpose Large Vocabulary Continuous Speech Recognition Engine Julius rev. 3.2, Kyoto University

[9] McCulloch W.S., Pitts, W., Bulletin of Mathematical Biophysics (1943) pp. 5-115

About Author (s):

Andrea Piroddi received his M.S. degree in electronic engineering from University of Rome - Tor Vergata and Ph.D. degree in Electronic Enigineering from Politechnic of Turin, in 1996 and 1999, respectively. He is currently an Adjunct Professor with the University of The People, Pasadena, California, USA, and a manager in Vodafone Group plc – EMEA Region.
He has authored more than 15 publications in refereed journals, book chapters, and conferences. His areas of interest are artificial intelligence, biometrics, image processing, machine learning, and information fusion.
He is a senior member of IEEE, member of the Universal Association of Computer and Electronics Engineers, member of Association on Advancement of Artificial Intelligence and the Association for Computing Machinery. He is also a Guest Editor of the IEEE ICT Express, an Area Chair of the ICT (Elsevier) journal.