TEXT-INDEPENDENT SPEAKER VERIFICATION

Gintaras Barisevičius
Department of Software Engineering,
Kaunas University of Technology, Kaunas, Lithuania
gintaras.barisevicius@stud.ktu.lt

ABSTRACT

This paper describes one of the biometric systems - text-independent speaker verification. It discusses the different stages of speaker verification in text-independent systems as well mentioning other systems for speaker verification. Each stage has its subparts, so those parts are discussed as well. The methods for the speaker-verification are displayed in the article. Feature extraction from the raw speech data is discussed. Pre-emphasis, windowing and other parts of feature extraction are mentioned. Vector Quantization, Gaussian Mixture Model, Hidden Markov Model, Artificial Neural Network approaches for classification in speaker verification are overviewed.

KEYWORDS

Text-independent, speaker verification, biometrics, security.

1. Introduction

Voice is a part of human biometrics. It is unique for each person. Since it is a natural phenomenon it is very comfortable to use it in daily life, compare to iris scan or fingerprint. The latter one, due to social circumstances is associated with criminals and the iris scan always is though as a dangerous procedure.

Voice biometrics, the same as other biometrics are vulnerable to forgery. One the biometric data is stolen in digital format, the one who possesses whose biometric features, will never be secure anymore, since you can't change your biometrics on demand, as you can change your password or key [3]. Besides due to the same problems in speech recognition, voice biometrics encounters obstacles that have to overcome. First, the environment is never ideal, so it has background noise. The same if you use telephone lines, you get channel noise. Furthermore, the equipment that is used for voice capturing differs, so the characteristics are not always the same when using different microphones and in some cases telephones (if you take compression methodology and etc.). The voice biometrics technology takes a special place among other biometrics, because the human voice is not so static over the life time such as human DNA or fingerprints[6]. It may change due to the trauma, stress, illness or you mental and physical state (e.g. if you just got up from the bed) [1], [6], [7].

Voice biometrics can be split up into two branches: speaker verification and speaker identification. In the speaker-verification systems the person is authenticated if he or she is the one who she or he claims to be. In this type of system you have to compare only one to one data set. In contrary, in speaker-identification systems you have to compare one to many data sets and find the one which matches. Speaker-identification is the process of finding the identity of an unknown voice, i.e. finding the identity of the person voice in the database, where the user was enrolled. Usually to identify the speaker from the voice sample is a more complex task, then to verify the user. Searching through vast databases of voice features takes lots of time, but this is not the main problem. Speaker identification is often used in forensic, where voice can be recorded with hidden microphones. So, the quality of such voice is far from perfect [1].

In this paper we will focus on speaker verification and mainly on text-independent speaker verification. In text-dependent speaker verification systems the user usually has to say a fixed phrase in order to be accepted by the system [5]. Sometimes the system can prompt the text, which user has to read, or even the user selects which phrase to read [4]. The purpose of text-independent speaker verification is to verify the claimed identity of a user and then to come with the decision whether the user is the one who he claims to be, or is he an imposter [2].

It is obvious, that text-independent systems is much harder to implement, that the text-dependent or text-prompted. The applications for text-independent verification systems are vast: starting with telephone services and ending up with handling bank accounts [1].

We can define two types of information while speaking about speaker identification. First, the human for voice recognition uses high level-information such as dialect, accent, talking style and etc. For computerized speaker voice recognition features like tone, frequencies, pitch period, rhythm, spectral magnitude and bandwidths of the voice [6]. The same type of information is valid for speaker verification as well.

When talking about biometrics it is very important to mention the false rejection (FR) and false acceptance (FA) rates. They depend on the thresholds set in the models. Modifying them you can adjust the model to best suit your case. For instance, if you have bank account voice verification, then you probably want the FA rate to

be very low and vice versa if you have internet application where the users just logging in to the web workspace, then voice verification FR rate should be low. The FR is contrary proportional to FA. When the two errors (FA and FR) are equal it is called equal error rate (EER). Usually it is the best trade-off to choose the thresholds in the model to meet EER [12].

2. System classification

The mentioned text-dependent speaker recognition systems can be classified into DTW (dynamic time warping) or HMM (Hidden Markov Model) based methods.[6]

There are two most popular approaches to text-independent speaker verification: GMM (Gaussian mixture models) and vector quantization (VQ). The latter one is a non-parametric method whereas GMM is a parametric method [7]. In vector quantization approach cepstral coefficients are used and if truncated them, it would provide a more stable representation of a speaker‘s utterance from one repetition to another. The GMM is a series of Gaussian distributions over the space of the data. Each Gaussian distribution in this model is characterized by a mean, a covariance matrix and a prior probability [11].

Sometimes even more sophisticated approaches are used, such as ANN (artificial neural networks) [15], HMM or covariance modeling [14], or even combined methods are used [16].

3. Enrollment (training phase)

First, the user has to be enrolled to the system. His voice characteristics must be entered to the database in order the system could perform authentication process on that person. This enrollment process sometimes is called training phase. Figure 1 shows the enrollment process, when the user provides the required data (in this case his voice samples) to the system. As you can see the training phase can be spitted up into independent modules, which follow each other [5].

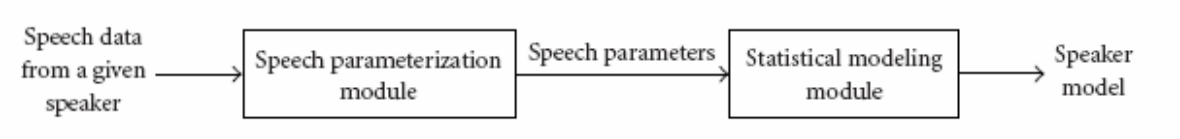

Figure 1 Training phase, when the user is enrolled to the speaker verification system [5]

At first, you need to extract the speech parameters from the sample data that is provided by speaker, in order to obtain the data for statistical modeling [5]. Secondly, you need to build a statistical model from those parameters obtained in the first step [5].

We will discuss about the extraction of parameters from the voice and construction of statistical model later in the paper.

4. Test phase

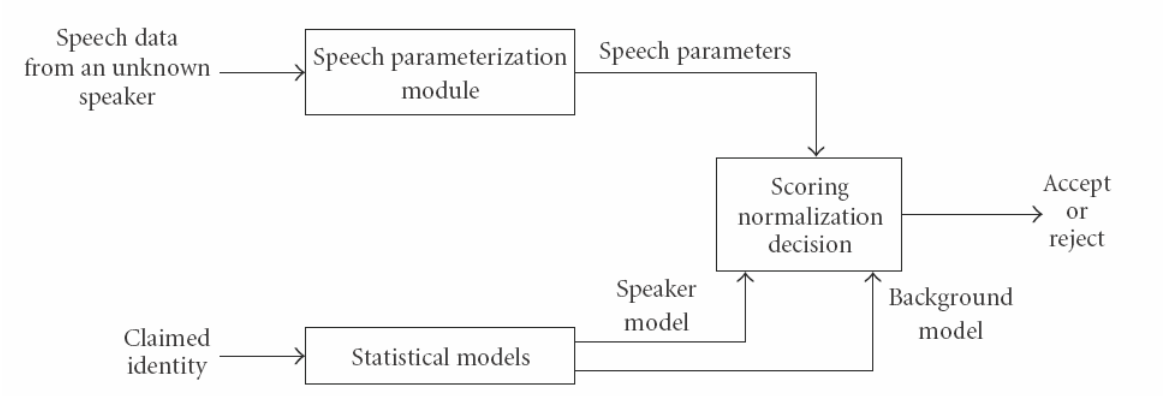

Figure 2 Test phase of speaker verification system [5]

For the speaker verification text phase the same speech parameterization module which is used in training phase is involved. The task of this module is to convert the speaker voice sample data to statistical information which could be processed and passed to the decision module. The speaker must claim his identity as well, so that his statistical model from the database could be retrieved and used in decision making. There are only two possible outputs for the text phase: the speaker is either accepted or rejected. The simple scheme for the test phase is represented in Figure 2.

4.1. Speech parameterization (feature extraction)

You can pose the question why for speech parameterization the low-level acoustic features are used? Why not to use prosodic features or phonetic features, or even lexical and syntactic features? The answer is quite simple. Even though lexical and syntactic features are robust to the channel effect and noise it is text dependent, hard automatically to extract and requires a lot of training data. The models for such features are also complicated. The similar problems lie in phonetic and prosodic features. The low-level acoustic features are easy to extract automatically and small amount of data is sufficient for modeling. The models itself are simple. Besides, it is text-independent, but the problem is that is very sensitive to the channel effects and noise [7].

Later on we will discuss about the feature extraction from the raw speech data.

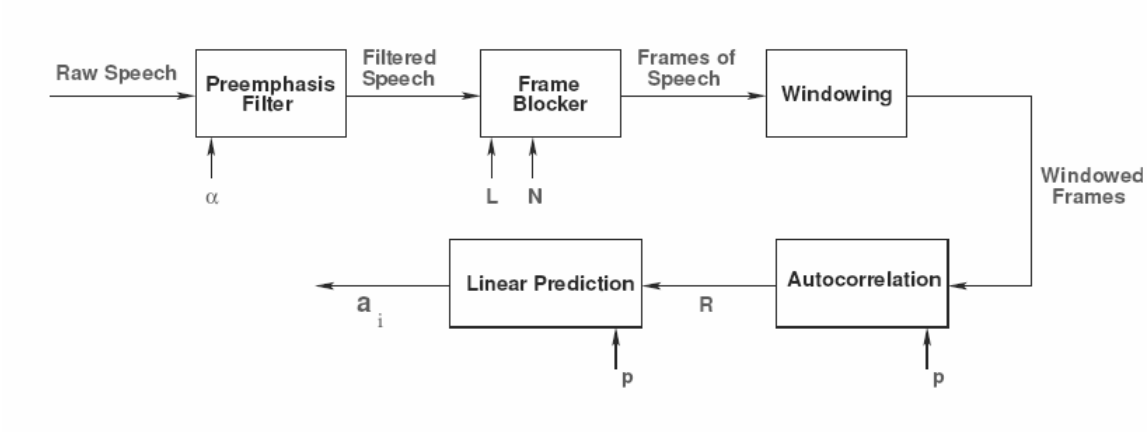

Figure 3 The LP feature extractor [12]

First the speech is captured by analog device and the signal is later on converted to digital signal, by analog digital converter (ADC) at the sampling frequency f_s. Then the digital speech signal is filtered by a first order FIR (Finite Impulse Response) filter which is the first module in Figure 3, where α is the degree of pre-emphasis. It is usually between 0.9 and 1.0 [11], [12].

Frame blocking is used afterwards the pre-emphasis filter. The signal thus is spitted into equal frames of length N. The beginning of the next frame is determined by adding the L offset to the start of previous frame. Thus the beginning of the second frame starts at L, the beginning of the third one starts at 2L and so on. If you draw such signal graphically as it is shown in Figure 4 you would see that if $L \leq N$ then the adjacent frames will overlap and of course, the Linear Prediction (LP) spectral estimates will show a high level of correlation [12]. This process is the second block in the Figure 3.

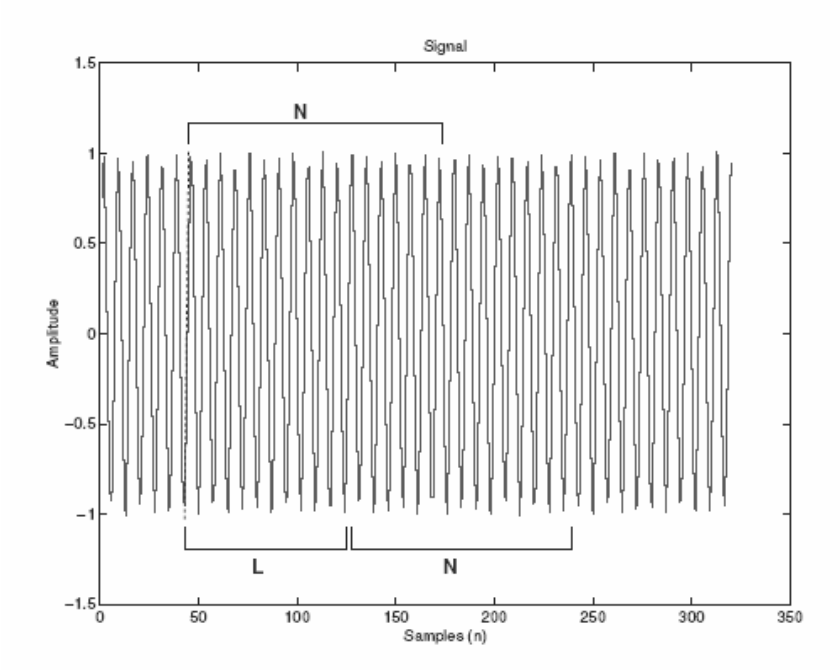

Figure 4 How the parameters N and L are utilized in the frame blocker [12]

According to [12] in the systems where the sampling frequency is 8 kHz, the values of N and N are 80 and 160 respectively. That means that the frame length is 30 ms and an update time is 10 ms.

The very next step is called windowing. There are different types of windowing, but the simplest is rectangular one. The simplicity not always means the best one. The rectangular window causes distortion in spectral analysis. The reason of that lies in the abrupt discontinuity at the beginning and end of the frame. The distortion caused by rectangular window can be reduced using continuous window function w(n). There is a wide spectrum of window functions. Some of them are know as Rectangular, Bartlett, Blackman, Hamming, Hanning, Kaiser, Lanczos and Tukey window functions. Their mathematical expressions can be found at [12]. Hamming window can be used for windowing the frames [8][9].

The next block in the Figure 3 is called autocorrelation. Autocorrelation term is known from the DSP (Digital Signal Processing). In the autocorrelation process the signal is correlated with itself, thus the most important harmonic and formant properties are extracted from the speech. The variable p in the Figure 3 is the LP analysis order. The p variable typically ranges from 8 to 16 as it is stated in [12]. R in the Figure 3 represents autocorrelation coefficients. Using autocorrelation silent frames can be discarded as well.

The last block in the Figure 3 is linear prediction (LP) module. In this part of parameterization process the correlation coefficients are converted into LP coefficients [12]. In the linear prediction coding (LPC) analysis it is assumed that the speech production model is a linear. The model that is usually used in LPC is regressive moving average (ARMA) model, which simplified variant is auto regressive (AR) model. For more details on these models see [13].

The four parts of the speech apparatus can be represented as separate filters: low-pass glottal filter, AR vocal tract filter, ARMA nasal tract filter and MA (moving average) lips filter [5]. Those all four filters can be interchanged with ARMA filter. Furthermore, the ARMA filter can be simplified to AR filter [5]. When you have a windowed signal, you can estimate the LPC coefficients (a.k.a. predictive coefficients) [5]. Those coefficients can be used as a parameter vector and later on the spectrum envelope can be calculated for the current window using predictive coefficients as it is shown in Figure 5 [5]. When the autocorrelation is very high (e.g. 100 in Figure 5 the coefficients representing values are more characterized.

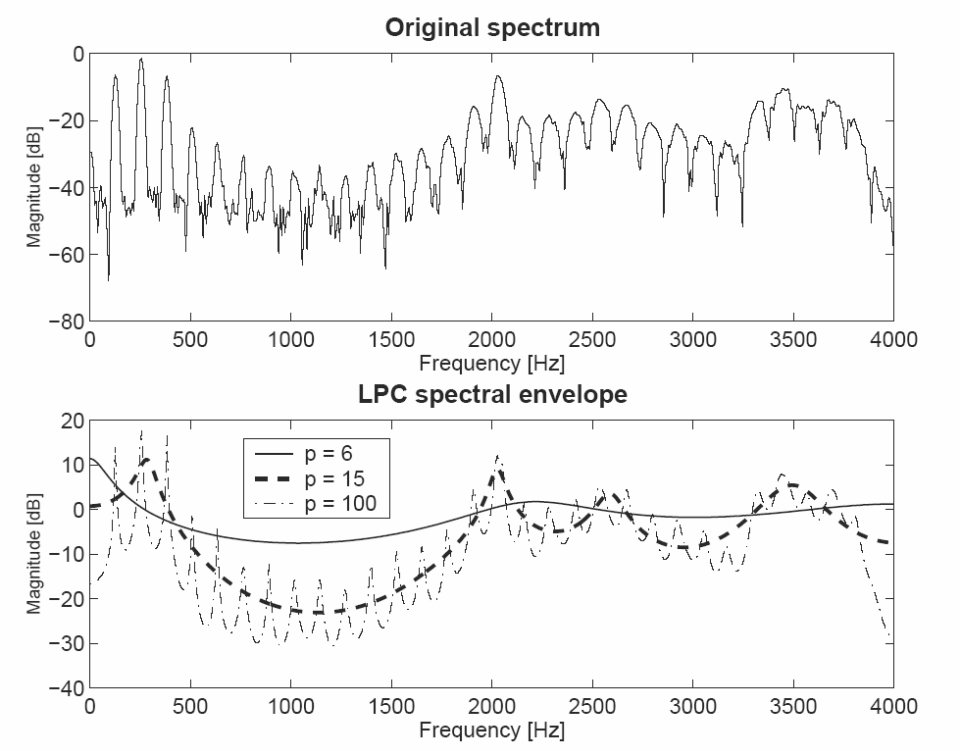

Figure 5 Estimation of the spectral envelope by LPC using different order predictors (p=6, 15, 100) [7]

One of the LPC coefficients estimation algorithms is called Levinson-Durbin algorithm which solves Yule-Walker equations using Toeplitz matrix of autocorrelation coefficients [12].

After the LPC coefficient estimation another step is needed to calculate cepstral coefficients. The cepstral coefficients are a better alternative to LP coefficients for speaker recognition. It is possible to calculate cepstral coefficients directly from LP coefficients (those coefficients are known as LP cepstral coefficients) or you can use Mel filter-bank analysis to extract them from frame blocked signal [12]. More about Mel filter-bank

cepstral coefficients (MFCCs) you can find in [7], [12]. Simply put MFCCs are calculated by taking the short-term power spectrum of the signal.

After the estimation of cepstral coefficients, they can be centered. It is done by subtracting cepstral mean vector from each cepstral vector. This procedure is called cepstral mean subtraction (CMS). It is quite common to use it in speaker verification. By doing CMS we remove slowly varying convolutive noises from the cepstrum [5]. The cepstral vectors can also be reduced [5].

Once the cepstral coefficients have been calculated, coefficient vectors have been centered and possibly reduced, we need in our estimations to incorporate some dynamic information, how these our calculated vectors vary in time, since the speech is not always the same and vary even in consequent utterances. In most of the applications it is done by using Δ and ΔΔ parameters [5]. The Δ parameter is the first derivative of delta parameter of the i^{th} cepstral coefficient as defined in:

$$\Delta f_k[i] = f_{k+M}[i] - f_{k-M}[i]. \qquad [12]$$

The M is the number of frames, which usually vary from 2-3. The Δ parameter is calculated for each frame separately, thus you get a vector of Δ parameters for each cepstral coefficient [12].

When you finished calculating the vectors, next step is to eliminate useless vectors. Those vectors which correspond to silence or background noise are discarded. One way of accomplishing that task is calculating bi-Gaussian model of feature vector distribution. When the Gaussian distribution has the "lowest" mean, that means it represents silence or background noise and can be discarded, in opposite, the Gaussians which has the "highest" mean represents speech portions [5].

4.2. Classification and pattern matching

This part of the paper introduces the classification and pattern matching in speech data. This part is the clumsy one, so we don't go into details, but rather will overview the classification and pattern matching in very abstract level.

Once you have produced the feature vectors the next task is to check the utterance of the corresponding feature vectors in the database, in other words you need to determine if the person who produced the speech and his claiming identity is the positive [12].

There exist a lot of model for classification. Template models are considered to be the simplest ones. It includes Dynamic Time Warping (DTW) and Vector Quantization (VQ) models [12].

Another group of classifiers is stochastic models. It includes Gaussian mixture model (GMM), Hidden Markov Model (HMM) and Artificial Neural Network (ANN) [5], [12].

4.3. Template methods

4.3.1. Dynamic Time Warping

DTW model is applicable to text-dependent speaker verification so I won't discuss it in details. The main idea of this approach is that if you have training template T consisting of N_T frames and test utterance R consisting of N_R frames, the Dynamic Time Warping model is able to find the function $m = \omega(n)$, which maps the time axis n of T to time axis m of R. Thus the system makes the comparison between the test and training data of the speaker evaluating the distance between them and makes the decision whether in favor of the user accepting him or the opposite - rejecting him. More about it can be found at [12].

4.3.2. Vector Quantization

A vector quantiser maps k-dimensional vectors in a vector space Rk into a finite set of vectors. Each vector from that space is called a code vector also known as codeword. The set of such codewords is called a codebook [19].

Entire space is partitioned into Varonoi regions, and each region is associated with the codeword. The space is divides so that the union of all regions would compose the entire space, but the regions do not intersect, so the intersection of all regions is an empty set [19].

When we have an input vector we can determine Euclidean distance between that input vector and the closest codeword. Euclidean distance can be calculated as follows:

$$d(x, y_i) = \sqrt{\sum_{j=1}^{k} (x_j - y_{ij})^2}, \qquad [19]$$

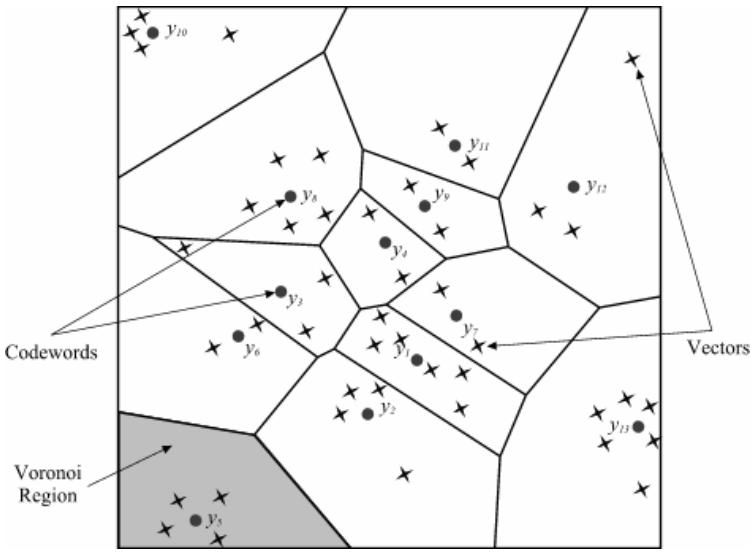

Figure 6 Codewords in 2-dimensional space. Input vectors are marked as a star, codewords are marked with circles and y_n vector. Voronoi regions are separated with boundary lines. [19]

In Figure 6 we display the two dimensional case. Each codeword resides in its Varonoi region. More about the codebooks and vector quantization you can read in [12], [19].

4.4. Statistical methods

Since we have only one speaker detection in our case we can formulate two hypotheses:

$$H_0 : Y \; is \; from \, the \, hypothesized \; speaker \; S,$$

$$H_1 : Y \; is \; not \; from \, the \, hypothesized \; speaker \; S.$$

Using likelihood ratio (LR) we can decide between our two hypotheses:

$$\frac{p(Y \mid H_0)}{p(Y \mid H_1)} \begin{cases} > \theta, accept \; H_0 \\ < \theta, accept \; H_1 \end{cases}, \qquad [5]$$

where the $p(Y \mid H_0)$ is the probability density function (PDF) for the hypothesis H0 evaluated for the observer speech segment Y. The PDF for H1 is $p(Y \mid H_1)$. The θ parameter is the threshold whether to accept or reject the hypothesis H0. The main task is to compute the values for $p(Y \mid H_0)$ and $p(Y \mid H_1)$ likelihoods [5].

The model which represent the hypothesis H0 is λ_{hyp} and the hypothesis H1 representing model is $\lambda_{\overline{hyp}}$. So the likelihood functions are then $p(Y \mid \lambda_{hyp})$ and $p(Y \mid \lambda_{\overline{hyp}})$. The problem is finding that model λ_{hyp} and $\lambda_{\overline{hyp}}$.

4.4.1. Gaussian mixture model

In contrary to HMM, the Gaussian Mixture Model (GMM) based methods is applicable for text-independent speaker recognition. It is claimed that GMM gains a high level of accuracy in such systems [12].

Three parameters are calculated for each speaker model in GMM: Gaussian densities $p_i(\vec{x})$, mean vector $\vec{\mu_i}$ and covariance matrix $\sum i$. Using those three parameters feature vector probability $p(x \mid \lambda)$ is calculated. The process of calculating those parameters is described in [5], [12].

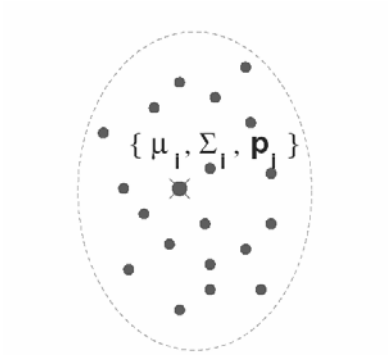

Figure 7 One component of GMM speaker model [12].

The example of one component of GMM speaker model is displayed in Figure 7. Entire speaker model consists of M components, where every of them contains those three parameters.

When using the training data, the speaker model parameters have to be estimated using expectation-maximization (EM) algorithm, which is discussed in detail [12], [20].

4.4.2. Hidden Markov model

The Hidden Markov Model is used for speech recognition, thus it is useful only for text-dependent speaker recognition. HMM is a stochastic model. The HMM can be viewed as a finite state machine. Each state (node) in it has an associated probability density function (PDF) for the feature vector. Moving from one state to another the probability of that transition is defined (the same as moving through the regular graph, since HMM is a graph). Only the first and the last states are not-emitting states, since the first is always where we start and the last one is the one where we always end our transitions, i.e. there are no incoming transitions into the start state and there are no output transitions from the end state. Every emitting state has a set of outgoing transitions and the sum of the probabilities for those transitions is equal to one, since the transition from non-final state always must occur [12].

For each n^{th} frame probabilities are calculated for produced feature vector at the visited state. This probability can be calculated using forward-backward algorithm [17], [18]. Thus the decision can be made about the user acceptance.

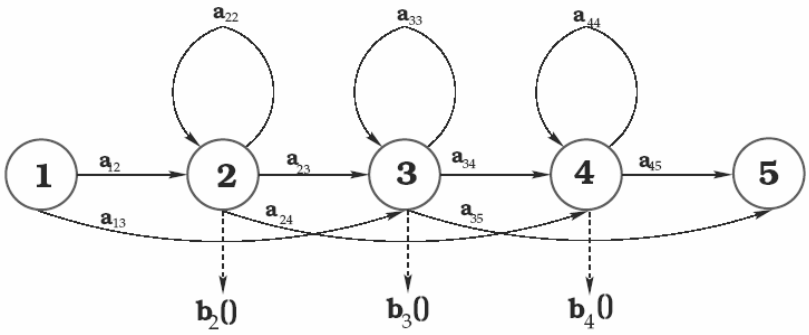

Figure 8 A simple left to righ HMM [12]

In Figure 8 the numbers are the states. 1^{st} is the start state and 5^{th} is the end state. The b_i are the PDFs. The a_{ij} are the probabilities.

4.4.3. Artificial Neural Network

The advantage of ANN is that they can be used in both text-dependent and text-independent speaker identification and speaker verification systems. We won't go deep into ANN types, just will mention that there exist many types of it, few of them are multi-layer perceptron (MLP), the radial bias function (RBF) and learning vector quantiser (LVQ) [12].

The MLP consists of three layers, which include input, hidden and output layer. You can create single MLP neural network for all speakers, trained with N output neurons, where N is the number of trained speakers, or you create a separate MLP neural network for each speaker, just containing two output neurons corresponding to the trained speaker and the rest of the world [12], [16].

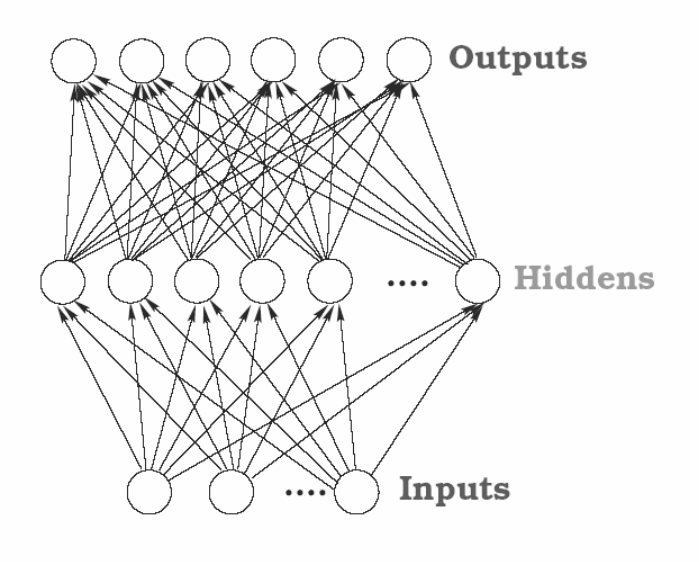

Figure 9 A two layered ANN [12]

In Figure 9 an example of two layered ANN is displayed. It is a two layered case of multi-layered perceptron. As you can see we have a layer of inputs, hidden nodes and a layer of outputs.

As in almost every ordinary artificial neural network, MLP is trained using the error back-propagation. The training is an iterative process. Initial values for each neuron in MLP are set randomly from the range of -0.5 to 0.5. For the each iteration the weights for the nodes are refined. The algorithm is performed in two passes -forward and backward passes. Using the training data the neural network finds the overall error in forward pass and in backward pass it minimizes the error adjusting neuron weights [12].

After the training phase is complete for the neural network, the test phase can take place, where the inputs are given for the neural network and the output is the probability of the claimed identity of the speaker [16].

5. Conclusions

General introduction to biometrics was discussed. General speaker verification problems were outlined and the metrics for the biometrics was displayed in this paper. It was briefly discussed about the classification of speaker verification systems. There various implementations and approaches for the speaker verification systems. The scheme of the general speaker verification in text-independent systems was presented. Each block in the scheme was discussed separately. Besides, the stages for the speaker verification were introduced in the paper: training phase and test phase.

The methods for feature extraction were discussed in the paper. We found out how the raw speech data is transformed to the digital data and then from that data the representing feature vectors are extracted. The classification and pattern matching methods were briefly overviewed. Vector quantization method was outlined and an example of such classification was displayed. Since it is not statistical method, so it is easier to understand it. Later in the paper, were discussed and statistical methods: GMM and HMM. It is obvious, that it is not so trivial to implement those methods. A deep knowledge of statistical mathematics is required. However, even more complicated methods known as Artificial Neural Network was shortly discussed. This methods is quite elegant for speaker verification, but hard to implement.

In general the paper outlines the field of speaker verification for text-independent systems and not going deeply into mathematics. The examples for classification and pattern matching were displayed as well as a very basic conceptual scheme for speaker verification.

References

[1] Judith A. Markowitz. Voice Biometrics. Article. http://portal.acm.org/ft_gateway.cfm?id=348995&type=pdf

[2] Johnny Mariethoz. 2004. Some Insight on Text Independent Speaker Verification Systems. Presentation slides. http://www.idiap.ch/~marietho/presentation/mariethoz-tam-2004.pdf

[3] Bruce Schneier. The Uses and Abuses of Biometrics. Article. http://portal.acm.org/ft_gateway.cfm?id=310988&type=pdf

[4] Alexandros Xafopoulos. 2001. Speaker Verification (an overview). Presentation slides. http://sigwww.cs.tut.fi/TICSP/PRESENTATIONS/2001%20Fulltexts/SpeakerVerif.pdf

[5] Frederic Bimbot, Jean-Francois Bonastre and others. 2004. A Tutorial on Text-Independent Speaker. Overview article. Verification. http://www.hindawi.co.uk/open-access/asp/volume-2004/S1110865704310024.pdf

[6] Gerik Alexander von Graevenitz. About Speaker Recognition Techology. Overview article. http://www.bergdata.com/downloads/Introduction%20to%20Speaker%20Recognition%20Technology.pdf

[7] Tomi Kinnunen. 2003. Spectral Features for Automatic Text-Independent Speaker Recognition. Licentiate's Thesis. http://www.cs.joensuu.fi/pages/pums/public_results/2004_PhLic_Kinnunen_Tomi.pdf

[8] L. Rabiner and B.-H. Juang. 1993. Fundamentals of Speech Recognition.

[9] Y. S. Moon, C. C. Leung. Fixed-point GMM-based Speaker Verification over Mobile Embedded System. http://portal.acm.org/ft_gateway.cfm?id=982517&type=pdf

[10] Douglas A. Reynolds, Larry P. Heck. 2000. Automatic Speaker Recognition Recent Progress, Current Applications, and Future Trends. Presentation slides. http://www.ll.mit.edu/IST/pubs/aaas00-dar-pres.pdf

[11] Svetoslav Marinov. 2003. Text Dependent and Text Independent Speaker Verification Systems. Technology and Applications. Overview article. http://www.speech.kth.se/~rolf/gslt_papers/SvetoslavMarinov.pdf

[12] Brett Richard Wildermoth. 2001. Text-Independent Speaker Recognition Using Source Based Features. Master of Philosophy Thesis. http://www4.gu.edu.au:8080/adt-root/uploads/approved/adt-QGU20040831.115646/public/01Front.pdf

[13] G. Fant. 1970. Acoustic Theory of Speech Production.

[14] Ran D. Zilca. 2001. Text-Independent Speaker Verification Using Covariance Modeling. IEEE Transactions On Speech And Audio Processing, Vol. 10, No. 6, September 2002 http://www.research.ibm.com/CBG/papers/zilca_sap2002.pdf

[15] Todor Ganchev, Nikos Fakotakis. Text-Independent Speaker Verification: The WLC-1 System. Research article. http://slt.wcl.ee.upatras.gr/papers/ganchev8.pdf

[16] Johan Olsson. 2002. Text Dependent Speaker Verification with a Hybrid HMM/ANN System. Thesis Project in Speech Technology. http://www.speech.kth.se/ctt/publications/exjobb/exjobb_jolsson.pdf

[17] L. R. Rabiner and B. H. Juang. 1986. An Introduction to Hidden Markov Models. IEEE Acoust. Speech Signal Proc. Magazine, Vol. 3, pp. 4-16, Jan. 1986. 25

[18] L. R. Rabiner. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Research article. http://seattleweb.intel-research.net/projects/guide/readinglist/papers/rabiner.pdf

[19] M. Qasem. Vector Quantization. Introductory article. http://www.geocities.com/mohamedqasem/vectorquantization/vq.html

[20] Jeff A. Bilmes. 1998. A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Research article. http://citeseer.ist.psu.edu/rd/96244207%2C436300%2C1%2C0.25%2CDownload/http://citeseer.ist.psu.edu/cache/papers/cs/21980/http:zSzzSzcrow.ee.washington.eduzSzpeoplezSzbulykozSzpaperszSzem.pdf/bilmes98gentle.pdf