Machine learning for speaker identification

Information Technology
Computatational Intelligence
DIAB ELMEHDI
1324316
el.diab@stud.fra-uas.de

Abstract—**Machine Learning (ML) is an intelligence branch that can directly learn from examples, data and experience by computer systems. By allowing computers to intelligently perform certain tasks, machine learning systems can perform complex processes by learning from data instead of following pre-programmed rules. That consist of networks having the ability to learn from the unstructured data. Learning can be supervised or unsupervised. Machine learning has been widely adopted in speech recognition and speaker identification (SID) task over traditional approaches such as those that use Mel-Frequency cepstral coefficients (MFCC) for feature extraction. In this study we will discuss about Conventional speaker identification system use by Gaussian Mixture Model (GMM) and support vector machine (SVM) to model a speaker voice based on the speaker acoustic characteristics. Furthermore, this paper also discusses some experimental result on several samples show the performance of the proposed approach.**

Keywords—Machine learning (ML), Mel-Frequency Cepstral coefficients (MFCC), Gaussian Mixture Model (GMM), Support vector machine (SVM), speaker identification (SID)

I. Introduction

Speaker recognition uses speech to identify a speaker based on its sound representations by matching the speaker 's voice profile with current profiles of specific speakers. It is also classified into closed and open set speaker identification. In the closed set speaker identification task, an unknown utterance will be assigned to the known speaker reference template with the highest level of similarity. The initial presumption, however, is that one of the given speakers has an unexpected utterance and the program makes an unavoidable decision to choose the best matching speaker from the speaker pool. Therefore, if the highest matching scores are smaller than the pre-set threshold, the reference example for an unknown speaker cannot be located in a freezer. Speaker identification task can be further divided into text-dependent and text-independent task [1-2]. Unlike text-independent speaker verification system [3-4], which is a process of verifying the identity without constraint on the speech content, text-dependent speaker verification requires the speaker pronouncing pre-determined pass-phrase [5-6] These pass-phrases may be unique (chosen by user or system), or prompted by the system. The concurrent technology in speaker identification is based on short-time speech signal analysis followed by machine learning based modeling. The most commonly used features for speaker recognition are the Mel frequency cepstral coefficients (MFCCs) [7-8]. In terms of speaker modeling, the Gaussian Mixture Models (GMMs) introduced in the mid-1990s [9] is widely considered to be a benchmark for modern text-independent Speaker Recognition. Discriminative approaches, such as support vector machines (SVMs) have also successfully been used in the task of speaker recognition [10]. As a stand-alone method as well as in combination with GMMs by concatenating the means of the Gaussian components of the GMMs to super-vectors and applying discriminative classification on them [10].

In this paper is organized as follows. A brief Introduction to machine learning is presented in section 2 followed by Section 3 with detailed process of speaker identification along with various phases of SID. Section 4. Introduce An overview of three popular machine learning algorithms with various implementations which attained state if art results. In section 5 an experimental result and analysis. Finally, conclusion and future work is presented as section 6.

II. Machine learning

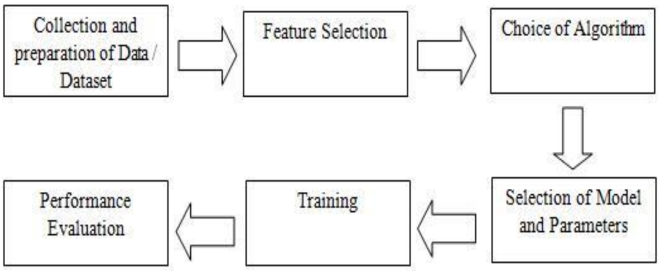
A. ML's general layout

ML is used to solve various problems that require learning on the part of the machine. A learning problem has three features:

- Task classes (The task to be learnt)
- Performance measure to be improved
- The process of gaining experience

The generic model of machine learning consists of six components independent of the algorithm adopted. The following figure 1 depicts these primary components.

Fig 1. Components of a Generic ML model


Collection and preparation of Data / Dataset
Feature Selection
Choice of Algorithm
Selection of Model and Parameters
Training
Performance Evaluation

Each component of the model has a specific task to accomplish as described next.

i. **Collection and Preparation of Data**: The primary task of in the machine learning process is to collect and prepare data in a format that can be given as input to the algorithm. A large amount may be available for any problem. Web data is usually unstructured and contains a lot of noise, i.e. both meaningless data and redundant information. The data must then be cleaned up and pre-processed in a standard format.

ii. **Feature Selection**: The data collected from the above stage can include various features that are not always important for the process of learning. Such features must be eliminated and a subset of main features must be provided.

iii. **Choice of Algorithm**: Not all algorithms for machine learning are for every problem. As discussed in the previous section, other algorithms more appropriate for the class problem. To obtain the best results possible, it is important to pick the best machine learning algorithm for this problem.

iv. **Selection of Models and Parameters**: Many machine learning algorithms require initial treatment to set the right values for different parameters.

v. **Training**: The model has to be programmed with a part of the data set as a training data after selecting the correct algorithm and fitting parameter values.

vi. **Performance Evaluation:** In order to evaluate how far it has been learned through different performance metrics, such as accuracy, precision and recall, the model must be checked before applying the system in real time.

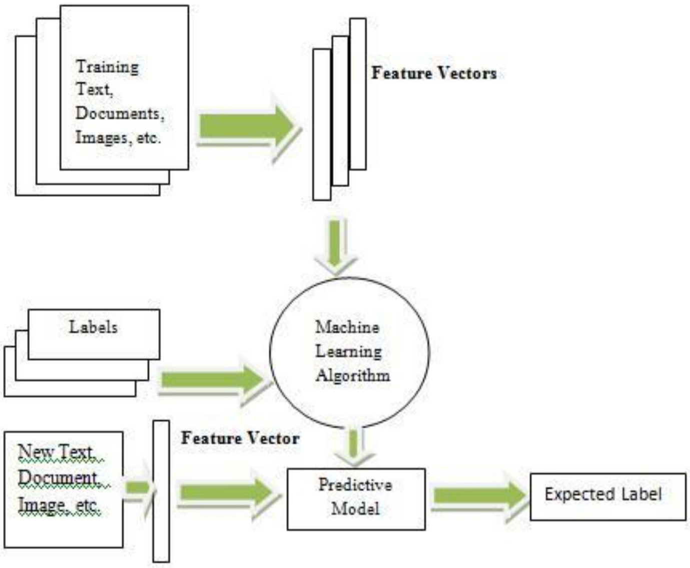
B. Machine learning Paradigms

Depending on how an algorithm is being trained and on the basis of availability of the output while training, machine learning paradigms can be classified into fours categories. These include: supervised learning, semi-supervised learning, unsupervised learning, reinforcement learning, [31,32,33,34]. Each of these paradigms is explained in the following sub-sections.

a) Supervised Learning

Under supervised learning, a set of examples or training modules are provided with the correct outputs and on the basis of these training sets, the algorithm learns to respond more accurately by comparing its output with those that are given as input. Supervised learning is also known as learning via examples or learning from exemplars. The following figure 2 explains the concept.

Fig. 2. Supervised Learning [31]

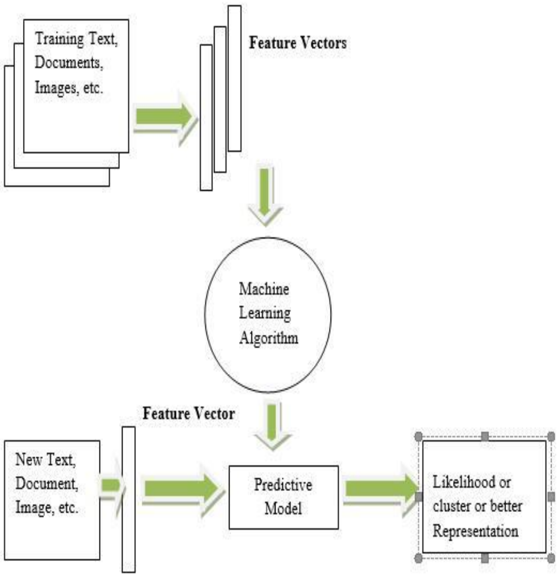

Training Text, Documents, Images, etc.
Feature Vectors
Labels
Machine Learning Algorithm
New Text, Document, Image, etc.
Feature Vector
Predictive Model
Expected Label

Supervised learning finds applications in prediction based on historical data. For example: to predict the Iris species given a set of its flower measurements or a recognition system that determines whether an object is a galaxy, a quasar or a star given a colored image of an object through a telescope, or given an e-commerce surfing history of a person, recommendation of the products by e-commerce websites [31]. Supervised learning tasks can be further categorized as classification tasks and regression tasks. In case of classification, the output labels are discrete whereas they are continuous in case of regression.

b) Unsupervised Learning

c) The unsupervised learning methodology is all about identifying unexplained current patterns from the data in order to draw laws from them. This approach is suitable in a situation where the types of data are uncertain. Education details are not marked here. Uncontrolled learning is known to be a mathematical approach to learning and hence relates to the question of discovering the unlabeled structure. The definition is defined in Figure 3.

Fig 3. Unsupervised Learning


Training Text, Documents, Images, etc.
Feature Vectors
Machine Learning Algorithm
Feature Vector
New Text, Document, Image, etc.
Predictive Model
Likelihood or cluster or better Representation

d) Semi Supervised Learning

These algorithms provide a technique that harnesses the power of both - supervised learning and unsupervised learning. In the previous two types output labels are either provided for all the observations or no labels are provided. There might be situations when some observations are provided with labels but majority of observations are unlabeled due to high cost of labeling and lack of skilled human expertise. In such situations, semi-supervised algorithms are best suited for model building. Semi supervised learning can be used with problems like classification, regression and prediction [31, 34,35].

It may further be categorized as Generative Models, Self-Training and Transudative SVM.

e) Reinforcement

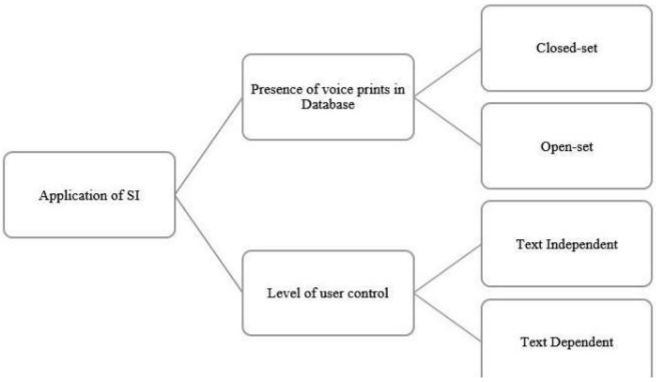
Reinforcement learning is regarded as an intermediate type of learning as the algorithm is only provided with a response that tells whether the output is correct or not. The algorithm has to explore and rule out various possibilities to get the correct output. It is regarded as learning with a Critic as the algorithm doesn't propose any sort of suggestions or solutions to the problem.

III. SPEAKER IDENTIFICATION ARCHITECTURE

Based on the application requirements, SID may be distinguished in different types as shown in figure 4. The first categorization of SID is based on the presence of speaker's voice print [11]. To put it differently, the SID can be a closed-set approach where the speaker is verified with the existing voice prints in the database.

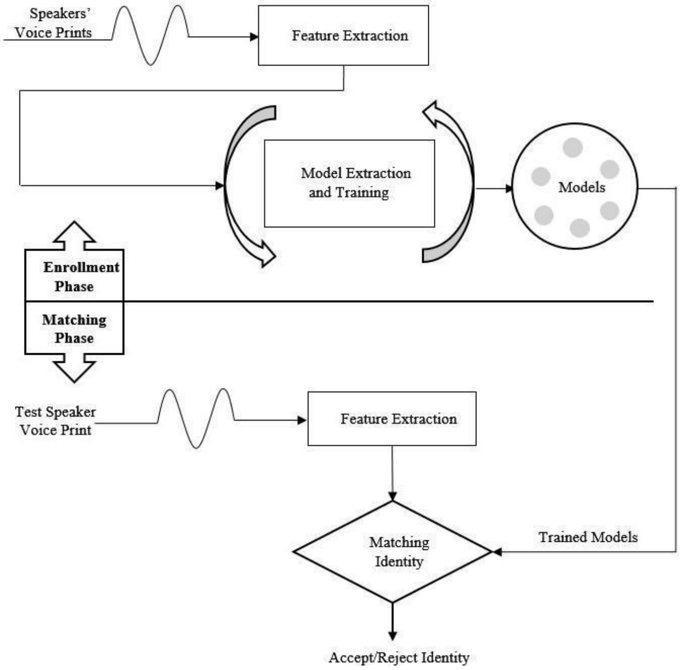
The other approach is an open set approach in which the identification process is done for a new speaker whose voice prints do not exist in the database. The second categorization can be based on the level of user cooperation and control in application. In other words, this category depends on the content of speech. This type is further divided into two types: text dependent approach and text independent approach. In the text dependent approach, the speaker repeats the same text for identification which he / she used during enrolment (i.e. training) [12]. In text independent SID systems, the speaker is identified irrespective of content of the utterance [13]. The identification approach in a text dependent SID can be either the same for all users or user specific.

Figure 4: classification of speaker identification



The SID's key function is to classify the test speaker in the speaker community. The characteristics of the input / test speaker are compared to existing speaker models (data). The SID process takes place in two phases. The first phase of training is known as a training stage and a second stage is known as matching. Figure 5 shows the process. The following is a concise overview of these two processes.

Figure 5: Phases of Speaker Identification



IV. MACHINE LEARNIG FOR SPEAKER IDENTIFICATION

Machine learning is one of the hottest research areas of data mining. It has been widely adopted in speaker recognition task. This section gives a brief overview of four machine learning algorithms related to this study, namely MFCC, GMM and SVM.

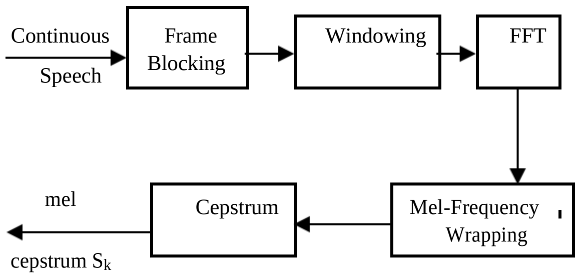
A. Feature Extraction

This module is intended to transform the waveform of the speech into some kind of parametric representation (with much lower information). The expression signal is a signal that changes gradually in time (it is called quasi-stationary). The properties are relatively constant when tested over a fairly short period of time between 5 and 100 ms. But the signal features shift over long stretches (in the 0.2 or more order) to reflect the different speech sounds spoken. And the most common way to describe the speech signal is to perform short spectrum sampling. There are different methods of parametrically representing the speaker recognition signal function, such as Linear Prediction Coding (LPC), MFCC and others. MFCC may well be the most common and well-known feature, which was used in this article. MFCCs are based on the known frequency variability in critical bandwidths of the human body. Two different types of filter, linear filters and logarithmically spaced filters, are used with this MFCC technique. The signal is provided in the Mel frequency scale to detect phonetically significant characteristics of voice. This measure has a linear frequency difference of less than 1000 Hz and a log spacing of more than 1000 Hz. Natural waveforms of speech can differ periodically based on the physical state of vocal cord speakers. Rather than the speech waveforms themselves, MFFCs are less susceptible to the said variations [14,16].

a) *The MFCC processor*

Figure 6 includes a structure block diagram for an MFCC processor. The input is recorded at a 22050Hz sampling rate. This frequency of sampling is selected to minimize the effects of the analog to digital conversion.

Figure 6: Block diagram of the MFCC processor



b) *Frame Blocking*

Frame Blocking involved filtering, the filtering mechanism converts the provided voice signal into a computer-appropriate type. Preprocessing distinguishes the speech section from the unfavored section.

c) *Windowing*

Window: It is used to reduce the distortion of the spectrum. We use a hamming window to block the frames at 20-25 meters to achieve a stationary conduct. The hamming window gives continuity in each frame at the beginning and end. It gives a better resolution of frequencies. The window result is shown as
Y(n) = X(n) x w(n).
Where, Y(n) – output signal
X(n) – input signal
w(n) – hamming window

d) *Fast Fourier Transforming (FFT)*

FFT is MFCC's key step in constructing the fast four-string transform of a system, which removes signal components at 10 ms speeds. Fast Fourier translates each sample N numbers from time domain to frequency domain. There are 512, 1024, 2048 FFT sizes. It is used to get frequency response in magnitude.

e) *Mel-frequency wrapping*

Tons of various frequencies are used in the sound. A subjective pitch is determined on the 'Mel' scale for every ton with a true frequency, F, expressed in Hz. The mel-frequency scale is a linear frequency difference less than 1000Hz and a log distance greater than1000Hz. A reference point for a 1KHz tone pitch is specified as 1000 mels, 40 dB above the threshold for perception. We can then measure the melts for a given frequency f in Hz using the following formula:

$$mel(f)= 2595*\log10(1+f/700) \ldots\ldots\ldots. (1)$$

A filter bank, for each desired self-frequency component, is an approach for simulating the subjective spectrum .. The filter bank has a three-pass frequency response and a constant mel frequency interval determines the distance and the bandwidth.

f) *Cepstrum*

The log mel spectrum should be adjusted back into time in the final phase. The result is called the cepstrum coefficients mel frequency (MFCC). For the given frame study, a cepstral interpretation of the speech spectrum gives an outstanding description of the local spectral properties of the signal. As the coefficients of the mel spectrum are true numbers (and their logarithms are likewise), they can be turned into the time domain by using the discreet cosine transform (DCT). The MFCCs may be calculated using the following equation [15] [17]:

$$\tilde{C}_n = \sum_{k=1}^{K} (\log \tilde{S}_k) \left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K} \right] \ldots\ldots\ldots\ldots\ldots\ldots (2) \qquad (1)$$

where n=1, 2, …, K. The number of mel cepstrum coefficients, K, is typically chosen as 20. The first variable, c~0 is not supported by DCT because it refers to the mean input signal value that carries little speaker results. In the implementation of the above procedure, a range of mel-frequency cepstrum coefficients is determined for each spoken frame of approximately 30 ms by overlapping. This parameter set is referred to as an acoustic vector. These acoustic vectors can be used to represent and recognize the voice characteristic of the speaker [18]. Each input utterance is therefore converted into an acoustic vector sequence. The following section describes how these acoustic vectors represent and acknowledge a speaker 's voice.

B. *Feature Matching*

The matching function is the recognition method of two identical repositories. Someone knows the root and the other the goal.

1) *GMM*

The GMM forms the basis for both the training and classification processes. GMM-based classifiers have shown good performance in many applications including speech processing [9]. This is a statistical method that classifies the speaker based on the probability that the test data could have originated from each speaker in the set [9, 20, 21].

a) *Feature Extraction.*

A statistical model for each speaker in the set is developed and denoted by λ. For instance, speaker s in the set of size S can be written as follows:

λs = _wi, μi , σi_ i = 1, . . . ,M; s = 1, . . . , S, (2)

where, w is weight, μ is mean, σ is a diagonal covariance, and M is the number of GMM components.

A diagonal covariance is used rather than a full-covariance matrix for the speaker model in order to simplify the hardware design. However, this means that a greater number of mixture components will need to be used to provide adequate classification performance.

The training phase consists of two steps, namely initialization and expectation maximization (EM). The initialization step provides initial estimates of the means for each Gaussian component in the GMM model. The EM algorithm recomputes the means, covariances, and weights of each component in the GMM iteratively. Each iteration of the algorithm provides increased accuracy in the estimates of all three parameters. The EM algorithm formulas [9, 20,21] are the following:

posterior probability:

$$p(i\,|x_i,\lambda) = \frac{P_i b_i(x_i)}{\sum_{k=1}^{m} p_k b_k(x_i)} \quad (3)$$

new estimates of *i*th weight:

$$\bar{w}_i = \frac{1}{T}\sum_{t=1}^{T} P(i|x_i,\lambda) \quad (4)$$

new estimates of mean:

$$\bar{\mu}_i = \frac{\sum_{t=1}^{T} P(i|x_i,\lambda)x_t}{\sum_{t=1}^{T} P(i|x_i,\lambda)} \quad (5)$$

new estimates of diagonal elements of *i*th covariance matrix:

$$\bar{\sigma}_i = \frac{\sum_{t=1}^{T} P(i|x_i,\lambda)(x_t \cdot x_t)}{\sum_{t=1}^{T} P(i|x_i,\lambda)} - \bar{\mu i}^2 \quad (6)$$

b) Classification

In this stage, a series of input vectors are compared, and a decision is made as to which of the speakers in the set is the most likely to have spoken the test data. The input to the classification system is denoted as

$X = \{x_1, x_2, x_3, \ldots , x_T\}.$ (7)

The rule to determine if X has come from speaker S can be stated as

$p\ (\lambda_s\ /\ X) > p\ (\lambda_r\ /\ X)\ r = 1, 2, \ldots , S\ (r \neq s).$ (8)

Therefore, for each speaker S in the speaker set, the classification system needs to compute and find the value of S that maximizes $p(\lambda_S \mid X)$ according to :

$$p(\lambda_s|X) = \frac{p(X\lambda_s)P(\lambda_s)}{P(X)} \quad (9)$$

The classification is based on a comparison between the probabilities for each speaker. If it can be assumed that the prior probability of each speaker is equal, then the term of $p(\lambda_s)$ can be ignored. The term $p(X)$ can also be ignored as this value is the same for each speaker [1], so

$$p(\lambda_s|X) = p(X|\,\lambda_s) \quad (10)$$

Where:

$$p(X|\lambda_s) = \prod_{t=1}^{T} P(x_t|\lambda_s) \quad (11)$$

Practically, the individual probabilities, $p(x_t\ /\ \lambda_s)$, are typically, in the range 10-3 to 10-8. There are 1000 test vectors for a test input of 10 seconds. When 10-8 is multiplied by itself 1000 times on a standard computer. the result will underflow and the probability for all speakers will be calculated as zero. Thus, $p(X\ /\ \lambda_s)$ is computed in the log domain in order to avoid this problem. The likelihood of any speaker having spoken the test data is then referred to as the log likelihood and is represented by the symbol L. The formula for the loglikelihood Function is. [9]

$$L(\lambda_s) = \sum_{t=1}^{T} \ln\left(p(x_t|\lambda_s)\right) \quad (12)$$

The speaker of the test data is statistically chosen by

$$\text{Speaker} = \max_{s\,=\,1}^{S} L(\lambda_s) \quad (13)$$

2) SVM

The support vector machine (SVM) is a linear machine pioneered by Vapnik [22]. SVM is a method for the kernel machine to make its decisions by building a hyperplane which separates two classes optimally. The hyperplane is defined by x · w + b = 0 where w is the normal to the plane.

For linearly separable data presented by $\{x_i, y_i\}$, $x_i \in \Re^d$, $y_i \in \{-1,1\}$,i = 1...N. Due to the ultimate criteria the optimal hyperplane is calculated. This is achieved by minimizing

$$\|w\|_2^2 \ \textit{subject to}\ (x_j \cdot w + D)y_i \geq 1, \forall i \quad (14)$$

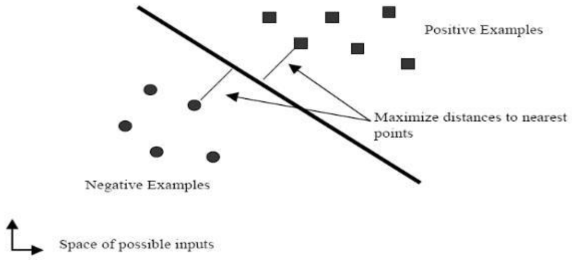
The solution for the optimal hyperplane w_0 is a linear combination of a small subset of data x_s , s ∈{1...N}, known as the support vectors that satisfy $(x_s \cdot w_0 + b)y_s = 1$.

There is no hyperplane for which all aspects that satisfy the above inequalities are not linearly separable data. To overcome this problem, ζ_i are introduced and the object is then achieved by minimizing

$$\frac{1}{2}\|w\|_2^2 + C\sum_i L(\zeta_i) \text{ Subject to } (x_j \cdot w + D)y_i \geq 1 - \zeta_i \qquad (15)$$

In cases where L is loss function, C is the hyperparameter for determining the impacts of empirical risk reduction, the limit maximization and the empirical risk of margins or erroneous points, as defined on RHS. This is the term used for L is the loss function. According to Burges [16], the dual formulation, which is more conveniently solved, of (3) with $L(\zeta_i) = \zeta_i$ is (16)

Figure 7: A linear support vector machine



a) . SVM training process

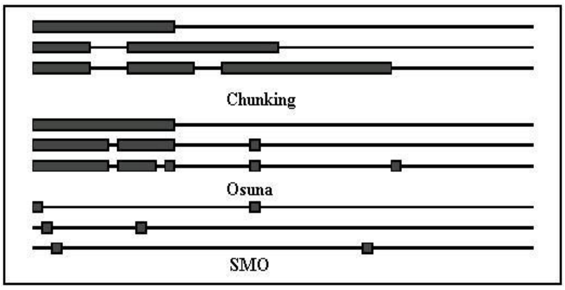
Although a quadratic optimization solution is guaranteed based on the conditions of Karush-kuhn tucker (kkt), depending on the separacy of the knowledge and the number of training data points, the amount of calculations needed can be very high. Because of the complexity, SVMs cannot address a quadratic optimization problem efficiently using generic QP methods. The quadratic form consists of a matrix with a number of elements equal to the number of exemplifiers. This matrix cannot be fit into 128 Megabytes if there are more than 4000 training examples. Vapnik [24] The chunking algorithm is based on the assumption that if you subtract the matrix columns and linear rows that correspond to the zero Lagrange Multipliers, it is the same value as that for a quadratic SVM problem, which is called the "chunking" ("chunking"). So it can be divided up into a series of simpler, quadratic problems with the overall goal of finding all non-zero Lagrange multipliers and discarding all negative Lagrange multipliers. At every step, chunking solves a quadratic optimization problem that consists of the following examples: every non-zero Lagrange multiplier from the last step, and the M worst examples that violate the KKT conditions [16], To a certain M meaning. When fewer than M examples violate KKT conditions in one step, all examples of breach are removed. The results of the previous sub-problem in each quadratic sub-problem are initialized. The entire series of non-zero Lagrange multipliers was defined in the last point. Therefore, the main question solves the last step. Chunking greatly decreases the matrix size from the number of squared test iterations to the non-null multiplier Lagrange squared. But chunking cannot even cope with large-scale training issues, because even this simplified matrix cannot fit into the brain.

Osuna et al.[25] demonstrated a theorem in 1997 which proposes a completely new series of quadratic algorithms for SVMs. Theory shows that a sery of smaller quadratic sub-problems can be grouped into the large quadratic optimization problem. So long as the explanations for the previous sub-problem contain at least one scenario that contradicts the KKT conditions, any move should minimize the total objective function and preserve a feasible point that is compatible with all of the limitations. Therefore, as series of quadratic sub-problems that always add at least one violator would be assumed to converge. Please note that the chunking algorithm obeys the theorem conditions and hence converges.

Osuna, et al. suggests keeping a constant size matrix for every quadratic sub-problem, which implies adding and deleting the same number of examples at every step [25]. Using a constant-size matrix will allow the training on arbitrarily sized data sets. The algorithm given in Osuna's paper [25] suggests adding one example and subtracting one example every step. Clearly this would be inefficient, because it would use an entire numerical quadratic optimization step to cause one training example to obey the KKT conditions. In practice, researchers add and subtract multiple examples according to unpublished heuristics [25]. In any case, a numerical QP solver is necessary for both of these methods. Computational quadratic optimization is extremely difficult to get right; there are multiple computational precision problems that need to be discussed.

For each procedure, three measures are outlined. The horizontal thin line at every step represents the training collection, while the thick boxes reflect the Lagrange multipliers being configured at that stage. For chunking, a set number of examples are added per step, while the zero Lagrange multipliers are discarded at every step. Therefore, the number of examples trained per stage continues to rise. For Osuna 's algorithm, a fixed number of examples are modified per step: the same number of examples is added to and discarded from the problem at every step. For SMO, only two examples are analytically optimized at any step, such that each step is very quick.

Figure 8: Three alternative methods for training SVMs: Chunking, Osuna's algorithm, and SMO.



b) Sequential minimal optimization (SMO)

Sequential marginal improvement (SMO) could be a straightforward algorithmic rule which will quickly solve the SVM drawback with none further matrix storage and while not victimization numerical quadratic improvement steps the least bit. SMO decomposes the general quadratic drawback into quadratic sub-problems, victimization Osuna's theorem to make sure convergence.

Unlike the previous strategies, SMO chooses to resolve the tiniest doable optimization drawback at each step. For the quality SVM quadratic drawback, the tiniest doable optimization drawback involves 2 Lagrange multipliers, as a result of the Lagrange multipliers should adapt a linear Equality constraint. At each step, SMO chooses 2 Lagrange multipliers to together optimize, finds the optimum values for these multipliers, and updates the SVM to mirror the new optimum values. The advantage of SMO lies within the proven fact that resolution for 2 Lagrange multipliers are often done analytically. Thus, numerical quadratic optimization is avoided entirely. The inner loop of the rule is often expressed during a short quantity of C code, instead of invoking a complete quadratic function. even supposing additional optimization sub-problems square measure resolved within the course of the rule, every sub-problem is thus quick that the quadratic drawback is resolved quickly.

In addition, SMO needs no further matrix storage in any respect. Thus, terribly giant SVM coaching drawback scan match inside the memory of a normal laptop computer or digital computer. as a result of no matrix algorithms square measure utilized in SMO, it's less prone to numerical exactitude issues. There square measure 2 parts to SMO associate analytic technique for resolution for the 2 Lagrange multipliers, and a heuristic for selecting that multipliers to optimize.

3) Hidden Markov Model (HMM)

The standard or reference answer and also the sub-band-based marker use HMM recognizers. The left-to-right or bark models ar the HMM kind that's used. This pattern is primarily wont to perceive voice or speaker. The chance of HMM speech feature vectors is decided with the chance of amendment between states and also the chance of the prevalence of feature vectors during a bound setting. There are three centrals HMM problems in finding the probability of speech feature vectors generated from an HMM [26]. Firstly, evaluation, that finds the chance that a sequence of visible states was generated by the model M and this, is solved by the Forward and Viterbi algorithms [26]. Secondly, secret writing finds state sequence that maximizes chance of observation sequence victimization Viterbi algorithmic program. Lastly, coaching that adjusts model parameters to maximize chance of discovered sequence. This last step is simply a problem of determining the reference speaker model for all speakers. The Baum-Welch re-estimation procedures or Forward-Backwards Algorithm are used for this case as presented in [26].

There are two known models of the HMM which are: Continuous HMM (CHMM) and Discreet HMM (DHMM). Research reveals that data is lost when modelling using DHMM [26], Since the probabilities of performance are determined using a quantized codebook, and only CHMM will be analyzed for this purpose.

V. EXPIREMENTAL RESULT AND ANALYSIS

We present our speech identification studies in this section. Next, we will explain specifics of the application of speech recognition studies. The data set details used and features for this study are included in the implementation details.
We then present the accuracy of the speaker identification obtained with the GMM classification and SVM-based classification devices.

a) Data-set and Features Used

We performed experiments on the 2002 and 2003 NIST speaker recognition (SRE) corpora [27, 28]. We have seen 122 male speakers typical to NIST SRE in 2002 and 2003. A minimum of around three minutes of speech time in the training period of 2002 and 2003 NIST SRE corporation provides a range of preparation details for a speaker. The 2003 NIST SRE corpus research results were used to check the speaker recognition schemes. There are roughly 30 s each of the check utterances. After elimination of pause sections of the voice, we break each utterance into fragments of roughly 5 s of training and experiments. Each section of speech is treated as a case. This results in a total of 3,617 teaching examples with 30 examples per speaker class. A total of 3,044 examples are listed. For function extraction from an example 's speech signals a frame size of 20 ms is used and a change of ten ms. -- frame is seen using a 39-dimensional vector with 12 Mel cepstral frequency coefficients (MFCC) log energy and delta and their accelerating coefficients. A selection of about 500 local function vectors represents any one of the training and test instances. The exactness of classification obtained for 3,044 examples is the identification of the speaker described in this section. The accuracy of the classification gives the percentage of test examples properly classified by the classification system. The classification accuracy indicates the speaker identification rate in the context of the speaker identification. Classification accuracy along with the 95 % confidence interval was provided to assess the statistical significance of the test. A simple asymptotic method (Wald method) [29]

TABLE I. Comparison of classification accuracy (in %), estimated at 95% confidence intervals, given by the GMM-based system and the adapted GMM-based system for speaker identification task [36]

Model	***Number of components (Q)***	***Classification accuracy (in %)***
GMM	32	75.81±1.52
	64	76.50±1.51
	128	71.26±1.61
Adapted GMM	1.024	83.08±1.33

The reliability intervals of classification accuracy are estimated at 95 percent. Classification accuracy confidence interval (CI) is measured as

$$CI = z\sqrt{\frac{a(1-a)}{L_{test}}} \quad (17)\ [36]$$

This is the exactness in decimals and the number of test cases is the number of Ltest. The z of the standard Normal distribution (1 − α/2) for a two-tailed α probability is. For an interval of 95 percent of confidence, z is worth 1.96. Within this analysis, the accuracy of speaker recognition systems based on GMMs, modified GMMs and dynamic SVMs based on the kernel is compared.

VI. Conclusion

This paper gives a comprehensive overview of how machine learning can be used for speaker identification through a different algorithm such as feature extraction MFCC. We have identified GMM approaches and SVM approaches in this chapter to recognize the speaker. The distressing preparation has a great deal to say The GMM parameter estimation method is expected to improve performance Method with optimum likelihood. Even if the number of groups is in the wide range GMMs, the optimization problem is computationally solved Quite serious. Really quiet. Creation of SVM classifiers for biased training the construction of the correct dynamic kernel for speaker recognition for different patterns of length represented by feature vector sets. Some current challenges are discussed which is hot topic of discussion: Hybrid Approach GMM/SVM.

References

[1] J. Campbell Jr., Speaker recognition: a tutorial. Proc, IEEE 85(9),, 1437–1462 (1997).

[2] F. e. a. Bimbot, A tutorial on text-independent speaker verification, EURASIP J. Appl.Signal Process. 1, 430–451 (2004).

[3] D. Q. T. D. R. Reynolds, Speaker verification using adapted gaussian, ISSN 1051–2004, 10(1–3), 19–41 (2000).

[4] S. Safavi, Speaker characterization using adult and children’s speech. Ph. D. dissertation,, 2015.

[5] S. G. H. M. I. S. R. Safavi, Fraud detection in voice-based identity authentication applications and services, In: Proceedings of ICDM, 2016.

[6] A. L. K. M. B. L. H. Larcher, Text-dependent speaker verification: classifiers,, ISSN 0167–6393, 60, 56–77 (2014),.

[7] S. M. P. Davis, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences., 28(4), 357–366 (1980): IEEE Trans. Acoust. Speech Signal Process.

[8] S. Furui, Cepstral analysis technique for automatic speaker verification., Speech Signal Process éd., (1981): IEEE Trans..

[9] D. R. R. Reynolds, Robust text-independent speaker identification using Gaussian mixture speaker models, Speech Audio Process éd., IEEE Trans, (1995).

[10] W. S. D. R. D. Campbell, Support vector machines using GMM supervectors for speaker verification, IEEE Signal Process, (2006).

[11] D. Reynolds, An overview of automatic speaker recognition In Proceedings of the International Conference on Acoustics, (ICASSP), 2002.

[12] A. A. a. M. D. H. Kekre, Speaker identification using row mean vector of spectrogram. In Proceedings of the International Conference & Workshop on Emerging Trends in Technology, 2011, p. pages 171–174. ACM.

[13] G. K. Verma, Multi-feature fusion for closed set text independent speaker identification, In International Conference on Information Intelligence, Systems, Technology and Management, Springer, 2011, p. pages 170–179..

[14] L. R. a. B.-H. Juang, Fundamental of Speech Recognition, Prentice-Hall,, 1993.

[15] E. R. B. J. a. L. R. A. V. F. Soong, «Quantization Approach to Speaker Recognition,» *AT&T Technical Journal*, pp. vol. 66,pp.14-26, March/April 1987..

[16] C. Frequently, «Asked Questions,» [En ligne]. Available: http://svr-www.eng.cam.ac.uk/comp.speech.

[17] J. D. H. J. a. P. J. Jr., «Discrete-Time Processing of Speech Signals,» *IEEE Press,New York,* n° %1second ed., 2000.

[18] C. J. C. A. Burges, «Tutorial on Support Vector Machines for Pattern Recognition, submitted to Data Mining and Knowledge Discovery,,» 1998. [En ligne]. Available: http://svm.research.bell-labs.com/SVMdoc.html,.

[19] R. Auckenthaler, "Test-independent speaker identification with limited resources,," University ofWales, 2001.

[20] . N. Holmes and W. J. Holmes, Speech Synthesis and Recognition,, 2nd edition, éd., London,: Taylor & Francis, 2002.

[21] S. Haykin, Neural Networks a Comprehensive Foundation,, Hamilton, Ontario, , McMaster University.

[22] [En ligne]. Available: ftp://ftp.ics.uci.edu/pub/machine-learningdatabases/.

[23] V. Vapnik, «Estimation of Dependences Based on Empirical Data,,» Springer-Verlag,1982.

[24] E. F. R. G. F. Osuna, «Improved Training Algorithm

for Support Vector Machines,» NNSP ’97, 1997.

[25] L. R. Rabiner, “A Tutorial On Hidden Markov Models And Selected Applications In Speech Recognition”, vol. 77(2), Proceedings of the IEEE, 1989.

[26] «The NIST,» speaker recognition evaluation plan, 2002. [En ligne]. Available: http://www.itl.nist.gov/iad/mig/.

[27] «THE NIST,» speaker recognition evaluation plan., 2003. [En ligne]. Available: http://www.itl.nist.gov/iad/mig/.

[28] L. C.-J. Chang C-C, «LIBSVM: a library for support vector machines. Software available,» 2001. [En ligne]. Available: http://www.csie.ntu.edu.tw/cjlin/libsvm.

[29] Q. T. D. R. Reynolds DA, «Speaker verification using adapted Gaussian mixture models.,» p. Digit Signal Process 10:19–41, 2000.

[30] N. RG, Two-sided confidence intervals for the single proportion: comparison of seven methods, 1998.

[31] C. K. R. Sandhya N. dhage, «“A review on Machine Learning Techniques”,» vol. Volume 4 Issue 3, pp. ISSN: 2321-8169, PP: 395 – 399, March 16.

[32] AyonDey, «Machine Learning Algorithms: A Review”».

[33] A. r. b. R. Society, «“Machine learning: the power and promise of computers that learn by example “,» vol. Vol. 7 (3), April 2017.

[34] Y. P. M. S. K. G. T. K. Al-Jarrah OY, «Efficient machine learning for big data : A review. Big Data Res.,» vol. 2(3):87–93., 2015.

[35] Z. X, «Semi-supervised learning literature survey,» University of Wisconsin-Madison, 2006.

[36] P. C. C. S. M. Tech. A. D. Dileep, Forensic Speaker Recognition, New York: Springer New York.