A
Dissertation
On

ONLINE SIGNATURE VERIFICATION USING SUPPORT VECTOR MACHINE (SVM)

Submitted in partial fulfilment of the requirements

for the award of the degree of

MASTER OF TECHNOLOGY

IN

SOFTWARE ENGINEERING

By

Anjali

(Roll No. 2K12/SWE/05)

Under the guidance of

Dr. (Mrs.) Daya Gupta
Professor
Department of Computer Engineering
Delhi Technological University

Mr. A. K. Bhateja
Scientist G
DRDO, New Delhi

Department of Computer Engineering

Delhi Technological University, Delhi

2012-2014

Department of Computer Engineering

Delhi Technological University

Delhi-110042

CERTIFICATE

This is to certify that the project report entitled **"ONLINE SIGNATURE VERIFICATION USING SUPPORT VECTOR MACHINE (SVM)"** is a bona fide record of work carried out by Anjali (2K12/SWE/05) under my guidance and supervision, during the academic session 2012-2014 in partial fulfilment of the requirement for the degree of Master of Technology in Software Engineering from Delhi Technological University, Delhi.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Dr. (Mrs.) Daya Gupta
Professor and Former HoD,
Department of Computer Engineering,
Delhi Technological University, Delhi

Mr. A.K.Bhateja
Scientist G ,
DRDO, Delhi

DELHI TECHNOLOGICAL UNIVERSITY

ACKNOWLEDGEMENT

With due regards, I hereby take this opportunity to acknowledge a lot of people who have supported me with their words and deeds in completion of my research work as part of this course of Master of Technology in Software Engineering.

To start with I would like to thank the almighty for being with me in each and every step of my life. Next, I thank my parents and family for their encouragement and persistent support.

I would like to express my deepest sense of gratitude and indebtedness to my guides and motivators, **Prof. Daya Gupta**, Department of Computer Engineering, Delhi Technological University **Mr. A.K. Bhateja,** Scientist G ,DRDO, Delhi for their valuable guidance and support in all the phases from conceptualization to final completion of the project.

I wish to convey my sincere gratitude to all the faculties and PhD. Scholars of Computer Engineering Department, Delhi Technological University who have enlightened me during my project.

I humbly extend my grateful appreciation to my friends whose moral support made this project possible.

Last but not the least, I would like to thank all the people directly and indirectly involved in successfully completion of this project.

Anjali

Roll No. 2K12/SWE/05

ABSTRACT

Selection of features of subject to be identified is very important step before applying any classification technique as different features have different type of significance associated with them. In online signature verification scheme we have to take care of selecting those features that can discriminate forgery and genuine signatures by giving a clear classification boundary. Extracting some other features from existing features by means of some technique can improve importance of that feature. We have combined three approaches together to give an optimal set of features for signature classification. Mean and variance analysis is done to identify Global features having capability to discriminate genuine and forgery one. Some Global features of signature were selected by PCA helps in identifying genuine signature genuine. Converting local features into more reliable FAR reducing feature is done with DTW and extended regression technique. We have designed RBF neural network and used it for classification. Results from all variations in features and classifiers are observed and discussed. A combined feature set obtained from three methods is passed to SVM classifier and results were improved rather than selecting features from individual techniques. We have compared our results and some other related work that reported their results on SVC2004 and it is found that accuracy of our algorithm is 95.375% .

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

The most easy-to-use way for identification of person is Hand written signature verification. Truly secured authentication by signature is becoming more and more crucial because signatures play an important role in legal, financial and commercial transactions. For example financial institutions like banks etc relay on signature for account openings, withdrawals, transaction payments through checks. A signature of a person is considered to be the seal of approval by that person and it is the most preferred acceptance for authentication. On the other hand frauds are increasing day by day hence losses continue to rise dramatically particularly in case of check fraud. the most suited technique static signature verification at the back-office is not good enough to reduce fraud thru payment forms, hence researches proposed a new technology which include dynamic behaviour of a person during signing via online capturing using pressure-sensitive pen-pads called online signature verification system.

1.1 General Concepts

Handwritten signature has been an accepted authentication technique in our society, in the areas of financial transactions or document authentication. Computer based online and offline signature verification approaches have been developed to extract the identity from signature of a person. Compared to the offline signature approach which uses static handwriting image of signature, online signature verification methods include dynamic features during signing hence has relatively higher classification rate and lower error rate. Online Signature verification system requires primarily a digitizing tablet attached with a special pen connected to the Universal Serial Bus Port (USB port) of a computer. This arrangement is capable of capturing both temporal and spatial information. An individual can sign on the digitizing tablet by use of the special pen regardless of size and positioning of his signature. Now a day a verity of smart phones are available in market which can be used to capture the signature using phone screen and special designed

attached stick. The signature is represented as pen-strokes consisting x y coordinates and thus this obtained data will be stored in the signature database in the form of a text file. This file can also include other dynamic features like time of writing, trajectory, pen velocity, acceleration and pressure which are much harder to imitate and when taken into consideration significantly improve the correctness and hence success rate of signature recognition.

1.2 Motivation

Signature is a type of behavioural biometric and it is not based on physiological properties of the individual, such as fingerprint or face. Signature of a person may change over time and it is not always unique and difficult to forge as iris patterns or fingerprints. For lower-security authentication needs Signature is widespread acceptance by the public and it is more suitable. MasterCard estimates a $450 million loss each year due to credit card fraud, likewise some billions of dollars being lost because of fraudulent encashment of checks. One recent survey finds that "27% of cardholders (debit, credit and prepaid) around the world have experienced fraud in the past five years. Rates of fraud vary across countries but in Mexico and the United States are 44% and 42% of respondents". According to the report from Aite Group and ACI Worldwide survey based on 5,000 consumers of 17 countries, reported that U.S. consumers are heavy card users, its obvious that more card use means a greater chances for card fraud [1].

A proper solution to reduce such losses could be Reliable automatic signature verification since there is involvement of handwritten signatures in the bank checks encashment and credit card transactions.

Using a password or any security PIN there is a chance of losing ,forgetting or sharing of this identification data, but the captured values of the handwritten signature are unique to an individual and in case of online signature which include dynamic behaviour it becomes very difficult to duplicate.

1.3 Related Work

For classification of any entity we need its important features that identify and distinguish it from others. Features of online signature can be broadly categorized in two type global features and local features. Local feature is extracted for each sample point in the input domain, where as global features is extracted for a whole signature, based on all sample points in the input domain. Some authors have done various experiments for selection of features of online signature and given their methodologies like Javier Galbally, Julian Fierrez, Manuel R. Freire, and Javier Ortega-Garcia categorized global features based on four types i) Time based ii) speed and acceleration based iii) direction based and iv) geometry based. They used genetic algorithm for selection of best feature from set of 100 features. With binary encoding they have assumed 0 in gene value as feature not selected and 1 in gene value as feature selected. For curse of dimensionality they used integer encoding in range [1 100] and finally concluded that Time based and geometry based features are the most discriminative when dealing with random forgeries, while speed and acceleration based and direction based parameters are the most appropriate to maximize the recognition rate with skilled forgeries [2]. Lee and Tobby calculated distance between features of genuine and forgery signatures of an individual users on basis of mean and variance and selected those features who had greater distance in maximum of users [3] .For extraction of local features we have to consider signature as a time varying signal hence there is need of unifying the signature signals and Dynamic Time Warping technique is mostly used for this work [4] [5] [6] [7] [8] [9]. But application of this algorithm reduces the discrimination between genuine signature and forgery one [10] so some authors given their own approach for this unification of time varying signal like Christian Gruber, Thiemo Gruber, Sebastian Krinninger and Bernhard Sick propsed a technique that uses a longest common subsequences (LCSS) detection algorithm on time series as kernel function for support vector machines (SVM) to measures the similarity of signature [11]. Marianela Parodi, Juan C. Gomez and Marcus Liwicki proposed their technique to approximate the time functions associated to the signatures using orthogonal polynomials series [12]. Least squares estimation techniques were used to compute the coefficients in these series expansions, which were used as features to model the signatures. M. Saeidi, R. Amirfattahi, A. Amini, M. Sajadi used ant colony for extremum matching of signals and to equalize their time duration [10]. After extracting feature a

classification technique is used for calculating the distance between 2 signatures, which is basically done with known samples to train the machine. There are many classifiers available for classification purpose but SVM is preferred because of its ability to provide optimal separating hyperplane that maximizes the margin of the training data clasess. Saeidi and Amirfattahi combined extended regression and SVM both for classification [10], thay have unified the signatures using ant colony, than calculated similarities using extended regression and extracted some similarity features and passed those features to SVM. Khalid Mokayed and Ono used Neural network and Fuzzy classifier [13], Marzuki Khalid, Rubiyah Yusof and Hamam Mokayed combined local and global features using fuzzy logic classifer [14].

1.4 Problem statement

For the forgoing section it can be concluded that there are various approaches for online signature verification using different kind of features. Some have worked on local features and some on global features and others have combined both. Most of them are not very reliable as their developed system may reject the genuine signature also because of use of alignment of time varying signature signal which reduces the gap between genuine and forgery signatures. Our aim in this thesis is to develop a reliable and accurate authentication system based on online signatures that can verify identity of an individual. It shall use optimized set of local as well as global features extracted from signature database that maintain the proper gap between genuine and forgery signatures and then intelligent classifier so as to complete the verification process. Hence problem can be stated as :

“To develop a reliable online signature verification system that uses optimized feature set and intelligent classifier”.

1.5 scope of work

Here we are proposing an online signature verification system based on combination of local and global features. We have modified mean and variance analysis proposed by L.L.Lee L, T. Berger, E. Avicze [3] to select global features that will be used for

discrimination of genuine and forgery signatures based on their distances. In addition to this another set of global features will be extracted using PCA that helps in identifying genuine signatures. Local features like x, y coordinates and velocities will be used for extracting the similarity features that will help in finding how similar is a signatures from stored genuine signature. Combined set of features from three methods will be passed to SVM classifier and RBF neural network classifier. We have used skilled forgeries signatures for training these classifiers so as to develop system is resistant to frauds. Accuracy will be verified by comparing our results with other techniques.

Hence sub problem of the thesis are the following:

1. To adapt existing mean variance analysis to select global feature set that maintain discrimination of genuine and forgery signature.
2. To use PCA to select genuine signature identifying features.
3. To use RBF neural network and SVM classifier to verify the signatures and to reduce error rate.
4. Establishment of our system's reliability by comparison of results.

1.6 Organization of thesis

The rest of this thesis is organized as follows:
Chapter 2 will provides basic overview of biometrics and then online signatures . It also tells how online signature is preferred over offline signature. Than it provide basic applications of online signature verification system and after this it gives basic system overview and all the component of the system like feature extraction, pre processing, enrolment matching etc are explained. At end Performance Evaluation criteria of system is explained.

Chapter 3 is giving detailed explanations of all the techniques being used in my proposed system such as dynamic time warping (DTW), regression, extended regression, support vector machine (SVM) and radial basis neural network (RBF).

Chapter 4 describes our proposed methodology. First it tells you about general system over view of proposed methodology. After this various kinds of features and ways to

extract them from database is described. Feature extraction process follows selection of best features to be used by our proposed system is explained next. At the end training and classification approach is explained.

Chapter 5 includes implementation details and experimental results from all the techniques proposed in chapter 4.

Chapter 6 summarizes the conclusion driven from experimental results.

Chapter 2

ONLINE VERIFICATION SYSTEM

Humans usually recognize each other based on their various characteristics for ages. We recognize others by their face when we meet them and by their voice as we speak to them. These characteristics are their identity. To achieve more reliable verification or identification we should use something that really recognizes the given person. The way a signature is handwritten creates information that is unique to each individual. Signature verification is the process used to recognize identity of an individual by use of his handwritten signature.

2.1 Biometrics

The term "biometrics" is derived from the Greek words bio (life) and metric (to measure). Biometrics means “the automatic identification of a person based on his/her physiological or behavioural characteristics”. Accuracy and case sensitiveness of biometric verification is very high hence it is preferred over traditional methods involving passwords and security PIN. A biometric system is a type of pattern recognition system by which a personal identification is done by determining the authenticity of a specific physiological or behavioural characteristic possessed by the user. These characteristics are unique and measurable. These characteristics need not be duplicable. How an individual is identified is an important issue in designing an identification system. Depending on the context, a biometric system can either be a verification or authentication system or an identification system. Figure 2.1 shows a general biometric system.

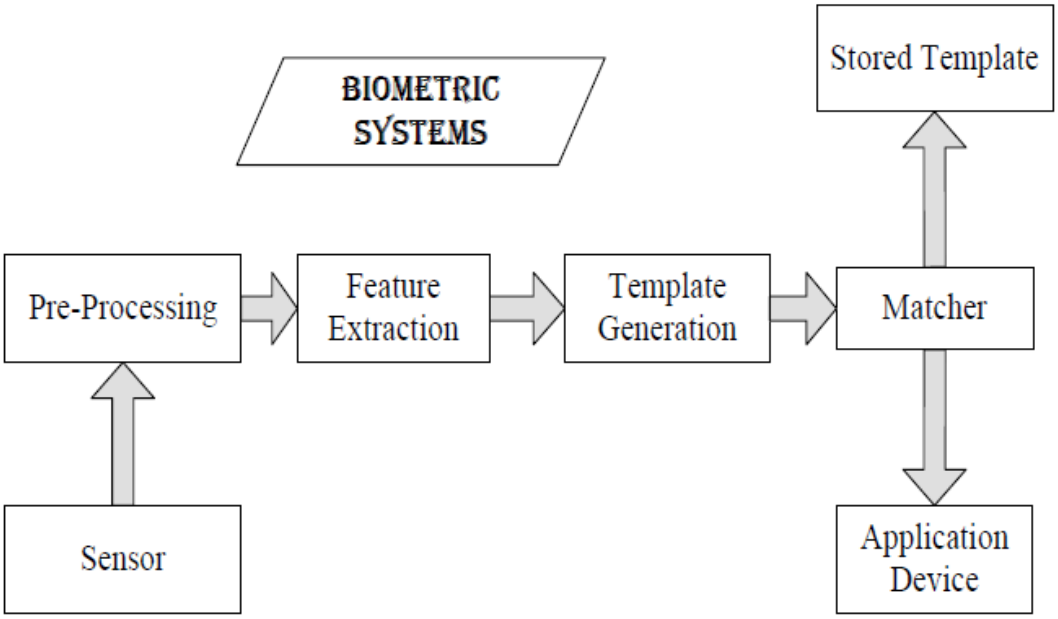


Figure 2.1 Biometrics Authentication System

2.2 Signature Verification

Signature verification is a common behavioral biometric to identify human beings for purposes of verifying their identity. Signatures are particularly useful for identification of a particular person because each person's signature is highly unique, especially if the dynamic properties of the signature are considered in addition to the static features of the signature. Even if skilled forgers can accurately reproduce the shape of signatures, but it is unlikely that they can simultaneously reproduce the dynamic properties as well.

2.2.1 Types of Signature verification

Signature verification is of two categories according to the available data available in input.

Offline (Static): here the image of a signature be input of offline signature verification system and is useful in automatic verification of signatures found on bank checks and documents. Some examples of this type of offline signature shown in Figure 2.2.

Figure 2.2: Offline Signatures

Online (Dynamic): Signatures that are captured by data acquisition devices like pressure-sensitive tablets (shown in Figure 1.3) and webcam that extract dynamic features of a signature in addition to its static shape features like curvature, length, width etc.,and can be used in real time applications like credit card transactions, protection of small personal devices (e.g. PDA), authorization of computer users for accessing sensitive data or programs, and authentication of individuals for access to physical devices or buildings.

Figure 2. 3: Online Signatures

2.2.2 Why Online (Dynamic)

Off-line signatures systems usually may have noise, because of scanning hardware or paper background, and contain less discriminative information since only the image of the signature is the input to the system. While genuine signatures of the same person may slightly change, the differences between a forgery and a genuine signatures may be difficult, which make automatic off-line signature verification be a very challenging pattern recognition problem. In addition, the difference in pen widths and unpredictable change in signature's aspect ratio are other difficulties of the problem. It is worth to notice the fact that only 70% of correct signature classification rate (genuine or forgery) is performed by professional forensic examiners .Unlike offline, On-line signatures are more unique and difficult to forge because of additional dynamic information speed, pressure, and capture time of each point on the signature trajectory are associated with signature data to be involved in the classification that's why on-line signature verification is more reliable than the off-line.

2.2.3 Advantages

In the point of view of adaption in the market place, signature verification presents three likely advantages over other biometrics techniques.

- First nowadays it is a socially accepted verification method already in use in banks and credit card transaction.

- Second, it is useful for most of the new generation of portable computers and personal digital assistants (PDAs) use handwriting as the main input channel.

- Third, a signature may be changed by the user. Similarly to a password while it is not possible to change finger prints iris or retina patterns.

Therefore, automatic signature verification has the unique possibility of becoming the method of choice for identification in many types of electronic transactions, not only electronics but also for other industries.

2.2.4 Applications

Signature verification has been and is used in many applications ranging from governmental use to commercial level to forensic applications.

A few of them are discussed below:

Security for Commercial Transactions: Nowadays signature verification used for commercial use. It can be used for authentication on ATMs, for package delivery companies. The internationally recognized courier service UPS has been using signature verification for many years now for personnel identification.

Secure computer system authentication: Logging on to PCs can be done with a combination of signature verification system and fingerprint identification system to achieve a higher level security in a sensitive area. We can also use a combination of password and signature verification system. This would allow the users to have a higher level of security and confidentiality for their clients and protection of their work.

Cheque Authentication: Signatures have been using for decades for cheque authentication in banking environment. But even experts on forgeries can make mistakes while identifying a signature. In general, Off-line signature verification can be used for cheque authentication in commercial environment.

Forensic Applications: Signature verification techniques have been used for cheque fraud and forensic applications.

2.3 General System Overview

A dynamic signature verification system gets its input from data acquisition device like a digital tablet or other, dynamic input device. The signature is then represented as time-varying signals. The verification system focuses on how the signature is being written rather than how the signature was written. This provides a better means to grasp the individuality of the writer but fails to recognize the writing itself [15]. There have been several studies on on-line signature verification algorithms. On-line signature verification

systems differ on various issues like data acquisition, preprocessing, and dissimilarity calculation.

2.3.1 General Diagram

In general online signature verification system has different phases. These phases are treated as an individual processes. The general system diagram for signature verification is as given below in Figure 1.4:

The Figure 1.4 shows the process used for development of system. Input is taken from a digitizer or such kind of device like webcam. This input is in the form of signal.

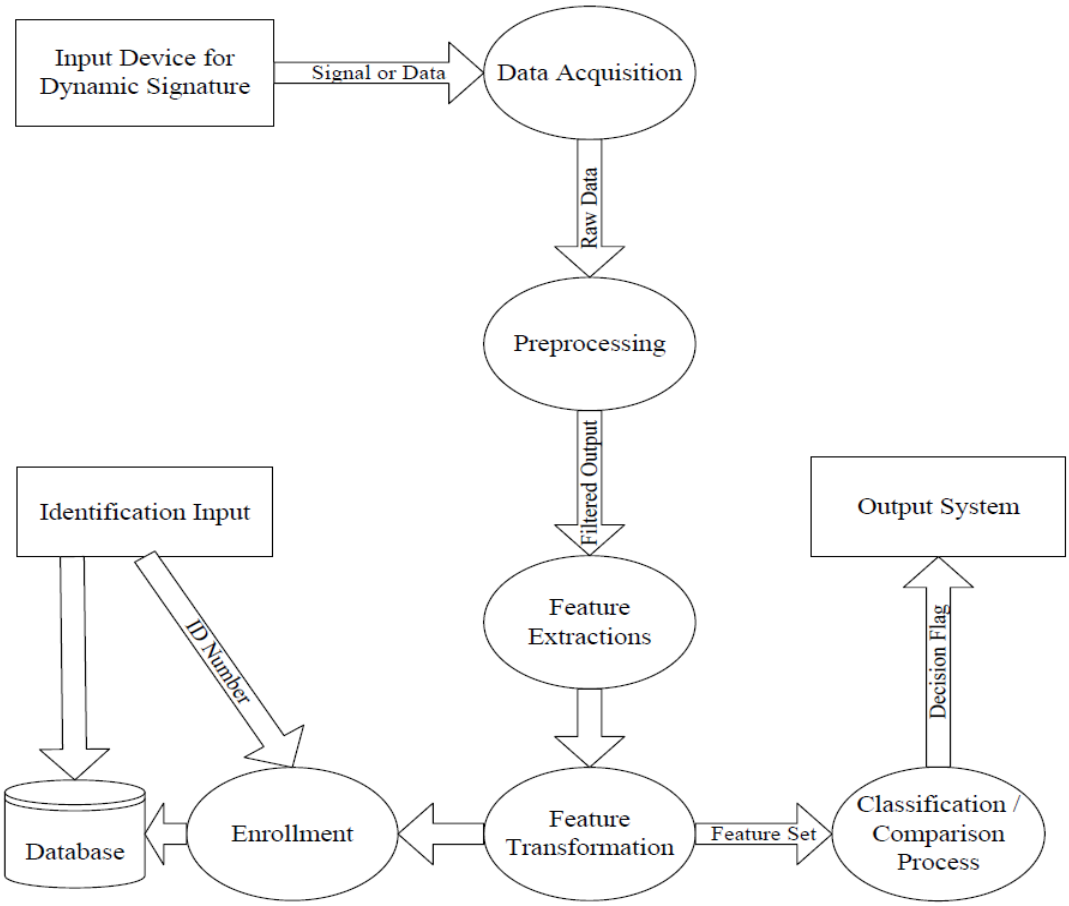


Figure 2.4 General System Overview

2.3.2 Input

For an on-line signature verification system, input is dynamic. This input is normally captured through a digital tablet or like other device. [16]This input is digitized and fed for processing. First of all pre-processing is done on the input received and then some features are extracted from the captured online data on the basis of which the signature is validated.

2.3.3 Output

The output obtained from an online signature verification system is a decision if the person providing the signature is authorized or not.

2.3.4 Preprocessing

There are some common preprocessing steps, aimed to improve the performance of a verification system. These include size normalization, smoothing of the trajectory and re-sampling of the signature data. Low resolution tablet or low sampling rates tablets may give signatures that have jaggedness which is commonly removed using smoothing techniques. In the systems where tablets of different active areas are used, signature size normalization is a frequently used as preprocessing technique. Comparing of two signatures having the same shape but of different sizes would result in low similarity scores. Size normalization is applied to remove that affect. Modern digital tablets have a sampling rate of more than 100 trajectory points per second. In some of the previous methods, re-sampling, as a preprocessing step, was used to remove possibly redundant data. After successful re-sampling, features can be reliably extracted.

2.3.5 Feature Extraction:

Feature extraction stage is one of the crucial stages of an on-line signature verification system. Features cans be classified as global or local, where global features represents signature's properties as a whole and local ones correspond to properties specific to a sampling point. The global features examples are signature bounding box, trajectory

length or average signing speed, and distance or curvature change between consecutive points on the signature trajectory are local features.

Feature Types for Signature Authentication

It is important to implement identity verification modality which provides high degree in performance and is still acceptable by a majority of users. A signature can be authenticated using either static (off-line) or dynamic (on-line) verification.

• Static (off-line): The signature is written either on a piece of paper and then scanned or directly on the computer using devices such as the digital pad. The shape of the signature is then compared with the enrolled (reference) signature. The difficulty with this technique is that a good forger will be able to copy the shape of the signature.

• Dynamic (on-line): The user's signature is acquired in real-time. By using this dynamic data, further feature such as acceleration, velocity, and instantaneous trajectory angles and displacements can be extracted.

The selection of features for extraction is difficult for the performance of a bio-metric authentication system. The features extracted must have able to describe the signature, separable between classes and also invariant within the same class. Two types of features can be extracted are both dynamic and static feature sets. For both dynamic and static feature sets, they are parameter based features and function based features. In general, function based features give better performance than parameters, but they usually time-consuming matching procedures. Parameter based features are easily computed and matched because of its simplicity.

When creating a system, it is important task is to take into account different external factors. For example like a bank or teller application, the retrieval of features and computation of matching has to be fast as well as accurate for feasibility for such an application. For daily access control depending on the level of security, speed is an issue. The cost of creating a system is also an issue for certain applications.

Certain criterions have to be established during feature extraction to obtain the suitability of the feature set. The list of the criteria shown below, which act as a guideline to obtain the appropriate features.

1. Selected features must have a high inter-personal variance to ensure that the signatures are separable based on different classes. This allows for low equal error rates during verification.

2. It is must to have a low intra-personal variance for the selected features. This will allow the same type of signatures to group together, enabling better performance for the system.

3. The features set extraction should be fast, quite simple and easy to compute in order to have a system which has low computational power.

4. The amount of features extracted has to be small enough to be stored in a smart card. The number of features should be small, will in turn allow for quicker and faster computation.

5. The number of features should be large enough to ensure that the signatures of different users are distinguishable with minimum computational risk.

6. Selected features cannot be reverse-engineered to get the original sketch of the signature.

This is to ensure that even if the features were to be obtained, the original knowledge of the signature is still unknown.

2.4 Performance Evaluation of Signature System:

The performance of biometric verification systems is typically described based on terms of the false accept rate (FAR) and a corresponding false reject rate (FRR). A false acceptance occurs when the system allows an forger's sign is accepted. A false reject ratio represents a valid user is rejected from gaining access to the system. As these two

are inversely related, lowering one often results in increasing the other. The equal error rate (EER) which is the point where FAR equals FRR.

There are two types of forgeries:

- A skilled forgery is signed by a person who has had practiced to sign a genuine signature of other person.

- A random or zero-effort forgery is signed randomly at any time without having any prior information about the signature, or even the name of the person whose signature is to be forged.

The performance of the available on-line signature verification algorithms give equal error rate between 1% and 10% , while off-line verification performance is still between 70% and 80% equal error rate.

The two errors false rejection rate (FRR) of genuine signatures and the false acceptance rate (FAR) of forgery signatures are directly correlated, where a change in one of the rates will inversely affect the other. A common alternative to describe the performance of system is to calculate the equal error rate (EER). EER corresponds to the point where the false accept and false reject rates are equal. In order to visually comment the performance of a biometric system, receiver operating characteristic (ROC) curves are drawn. Biometric systems generate matching scores that represent how similar (or dissimilar) the input is compared with the stored template. This score is compared with a threshold to make the decision of rejecting or accepting the user. The threshold value can be changed in order to obtain various FAR and FRR combinations [17].

The ROC curve represents how the FAR changes with respect to the FRR and vice-versa. An ROC curve example is shown in Figure 1.5. These curves can also be plotted by using the genuine accept rate versus the false accept rate. The genuine accept rate is obtained by simply one minus the FRR.

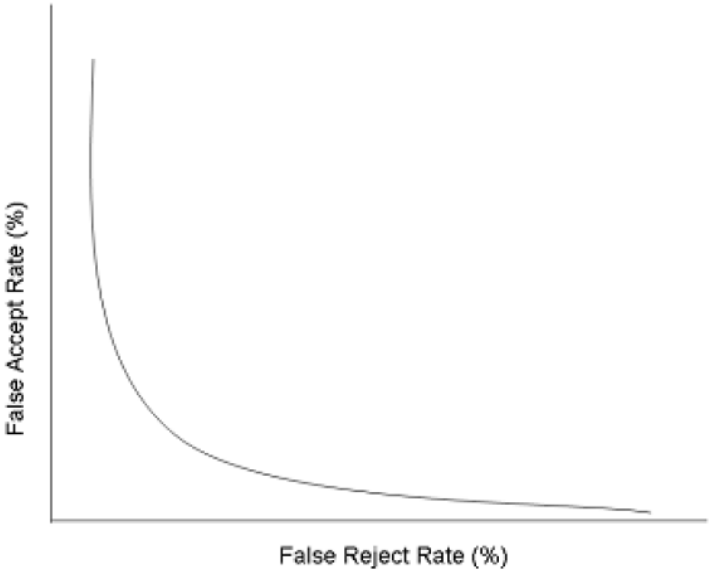


Figure 2.5 Example of a receiver operating characteristic (ROC) curve

Chapter 3

RELATED CONCEPTS

Here we are explaining all the concepts that are to be used in our signature verification system. SVM Classifier concepts is explained here and given complete details of RBF neural network parameters so as to implement it by self. Similarly Dynamic time warping, regression and extended regression is also explained theoretically as well as mathematically so as to give full visualization and provide clear concept.

3.1 Dynamic time warping (DTW)

Dynamic time warping (DTW) is a famous technique to optimally align two given (time-dependent) sequences under certain restrictions. Warping on two time sequences is performed in a nonlinear fashion to match them. Initially DTW has been used to compare different speech patterns in automatic speech recognition. In fields like data mining and information retrieval, DTW has been successfully applied to automatically cope with time deformations and different speeds associated with time-dependent data [18].

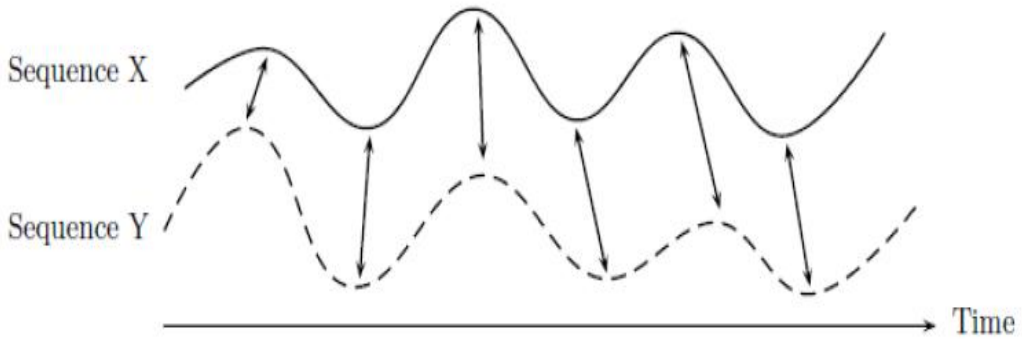


Figure 3.1: Time alignment of two time-dependent sequences. Aligned points are indicated by the arrows.

Given two sequences $X = (x1, x2, ..., xn)$ and $Y = (y1, y2, ..., ym)$, the distance DTW(X,Y) is calculated by construct an n-by-m matrix d , where d(i, j) is the distance between x_i and y_j. Although many different distance metrics can be employed, typically the Euclidean distance is used, so

$$\mathrm{d}(i,j) = \left(x_i - y_j\right)^2 \qquad (3.1)$$

Each matrix element of d(i, j) is the alignment between points x_i and y_j . A warping path P is a contiguous set of matrix elements that define a mapping between the series X and Y . The complete warping path is described as:

$$P = \{P1, P2, \ldots, PK\} \quad \max(n,m) <= K <= n + m\text{-}1 \qquad (3.2)$$

For Optimal warping path in the matrix which starts from cell (1, 1) to cell (n,m) so that the average cumulative cost along the path is minimized. If the path passes cell (i, j), then the cell (i,j) contributes d(x_i, y_j) to the cumulative cost. This path can be determined using dynamic programming [16],

$$P_{l-1} = \begin{cases} (1, m-1), & if, n = 1 \\ (n-1, 1) & if, m = 1 \\ \arg\min\{D(n-1, m-1), D(n-1, m), D(n, m-1)\} & otherwise, \end{cases} \qquad (3.3)$$

3.2 Linear Regression

Let us consider two sequences $X = (x1, x2, \ldots\ldots, xn)$, $Y = (y1, y2, \ldots\ldots, yn)$, linear regression statistically analyses the distribution of points $(x1, y1), (x2, y2), ..., (xn, yn)$ in the X-Y space. If X and Y have strong linear relation, i.e., $Y \approx \beta_0 + \beta_1 X$, we can expect the distribution of these points is along a line, called the regression line. To get regressed sequence Y on X, we a model is establish: $Y = \beta_0 + \beta_1 X + u$, where u is the error term. Note that $u = (u_1, u_2, \ldots\ldots, u_n)$. Then we estimate the parameter β_0 and β_1 in the sense of minimum-sum-of squared-error, ie

$$\sum_{i=1}^{n} u_i^2 = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2 \qquad (3.4)$$

is minimized. From a geometric point of view, we estimate the regression line which is determined by β_0 and β_1 so that the line fits the points in the X-Y space as close as possible [4].

$$\text{Let } Q(\beta_0, \beta_1) = \sum_{i=1}^{n} u_i^2 \qquad (3.5)$$

Note that Q is a function of β_0 and β_1. To minimize $Q(\beta_0, \beta_1)$, we have $\frac{\partial Q}{\partial \beta_0} = 0$ and $\frac{\partial Q}{\partial \beta_1} = 0$, Starting from this, we can obtain the following results

$$\beta_0 = \bar{Y} - \beta_1 \bar{X} \qquad (3.6)$$

and $$\beta_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{X})(y_i - \bar{Y})}{\sum_{i=1}^{n} (x_i - \bar{X})^2} \qquad (3.7)$$

where $\bar{X} = \frac{1}{n}\sum_{i=1}^{n}(x_i)$ and $\bar{Y} = \frac{1}{n}\sum_{i=1}^{n}(y_i)$

With β_0 and β_1 as above, the regression line is determined. There remains a question: how well the regression line fits the points in the X-Y space? As a measure of the goodness-of-fit, R-squared is defined as.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} u_i^2}{\sum_{i=1}^{n} (y_i - \bar{Y})^2} \qquad (3.8)$$

R-squared is also called the coefficient of determination. It can be interpreted as the fraction of the variation in Y that is explained by X. R-squared can be further derived as:

$$R^2 = \frac{\left[\sum_{i=1}^{n} (x_i - \bar{X})(y_i - \bar{Y})\right]^2}{\sum_{i=1}^{n} (x_i - \bar{X})^2 \sum_{i=1}^{n} (y_i - \bar{Y})^2} \qquad (3.9)$$

The properties of R^2 are follows:

• Reflexivity, i.e., R^2 (X,X) = 1.

• Symmetry, i.e., R^2 (X, Y) = R^2 (Y,X). According to equation (4), no matter Y regresses on X or X regresses on Y , R^2 is the same.

• $R^2 \in [0, 1]$. The closer the value to 1, the more the points tend to fall along the regression line, thus, the stronger linear relation the two sequences have. R^2 = 1 means the two sequences have perfect linear relation, while R^2 = 0means they have no linear relation at all.

Based on the properties of R^2 as above, R^2 is defined as the confidence of the linear relationship. Also, R^2 is a good measure for similarity and threshold based on R^2 is much more intuitive than some distance tolerance Ɛ, such as Euclidean distance or DTW distance. Given two sequences, R^2 directly tells their similarity [4].

3.3 Extended Regression (ER^2)

Traditionally, simple linear regression is only applied to 1-dimensional sequence. There are many kinds of multidimensional sequences like on-line handwritten signature sequence is multidimensional, because it includes coordinates (x and y), pressure, inclination, etc. To match M-dimensional sequence, Hansheng Lei, S. Palla, V. Govindaraju proposed an extended regression ER^2 as follow

$$ER^2 = \frac{\left[\sum_{j=1}^{M}(\sum_{i=1}^{n}(x_{ji}-\bar{X}_j)(y_{ji}-\bar{Y}_j))\right]^2}{\sum_{j=1}^{M}\sum_{i=1}^{n}(x_{ji}-\bar{X}_j)^2 \sum_{j=1}^{M}\sum_{i=1}^{n}(y_{ji}-\bar{Y}_j)^2} \qquad (3.10)$$

where $\bar{X}_j$ and $\bar{Y}_j$ is the average of the j-th dimension of sequence X (Y) [4].

ER^2 has similar properties as R^2, i.e., reflexivity, symmetry and $ER^2 \in [0, 1]$. The only difference is that ER^2 can measure multidimensional sequences. ER^2 tells about the similarity of two multidimensional sequences.

3.4 Principal component analysis (PCA)

It is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in data can be hard to find in data of high dimension, where the luxury of graphical representation is not available, PCA is a powerful tool for analysing data. The other main advantage of PCA is that once you have found these patterns in the data, and you compress the data, ie. by reducing the number of dimensions, without much loss of information. Principal component analysis (PCA) involves a mathematical procedure concerned with elucidating the covariance structure of a set of variables. In particular it allows us to identify the principal directions in which the data varies. Traditionally, principal component analysis is performed on the symmetric Covariance matrix or on the symmetric Correlation matrix calculated from the data matrix. First we obtain eigenvectors of a square symmetric matrix with sums of squares and cross products and then eigan value is obtained. The eigenvector associated with the largest eigenvalue corresponds to the first principal component i.e. the feature that is most important to identify the object.

3.5 Radial Basis Function Network (RBFN)

A Radial Basis Function Network (RBFN) is special type of artificial neural network. In this section, we are describing how it can be used as a non-linear classifier.

An Artificial Neural Networks is consisting of an input layer, some hidden layer and at end output layer. Each layer has some computation units called neurons. Input is given to input layer in form of input vector is also called feature vector of the object to be classified. Now this input is multiplied by a coefficient called weight which is assigned to each connection joining one neuron to other neuron. Weighted sum of input layer neurons is passed to activation function which is nothing but a function that transformed the input supplied to it in another form. There are many types of functions which can be used as activation function. Sometimes a bias may be needed before transferring weighted sum of input to activation function to control the input in given range. The output of one neuron unit of previous layer is passed as input to neuron unit of next layer. And same process gets repeated here like multiplication of weights to input, addition of

bias to it and transforming value of weighted sum of inputs to another value by activation function. At end when it reaches to output layer we assign different values of neurons for corresponding classes. Setting of weights is done through training phase of neurons where we start from random weights and on the basis of known output error is calculated at end phase and on the basis of this error we again change the weights of neurons in each layer. This process gets continued till weights get saturated. Now after training network can be used as classifier. For simple linear classifier problems only 1 layer of fewer neurons in hidden layer is sufficient is called simple Perceptron model but for complex one we need more complex model of neural network called multilayer Perceptron model.

Radial basis neural network is more intuitive than multilayer Perceptron model. In RBF approach similarity measured from training input set is used for classification of next samples. The difference between a multilayer Perceptron model and Radial basis neural network is we can have multiple layers in hidden layer of MLP while we have only one layer in hidden layer of RBF neural network. Another difference is weights of hidden layer are computed from calculating the error from training samples and propagating it through all the layers and repeating the process till we achieved at saturation level in weights. While in radial basis neural network we use any clustering technique to find weights of hidden layer.

Classification of an input sample is done on the basis of Euclidean distance between the input and its prototype which is nothing but the samples which were used to calculate the hidden layer weights in training phase. The prototype from which this distance is minimum becomes its class. This was the basic idea of functioning of radial basis neural network. Lets discus it in detail.

The bellow is typical architecture of a radial basis neural network. It is consist of input a layer of RBF neurons to which an input vector also called feature vector is passed, and an output layer with same number of neurons as total number of classes identified by this model. Each node in output layer corresponds to an output class.

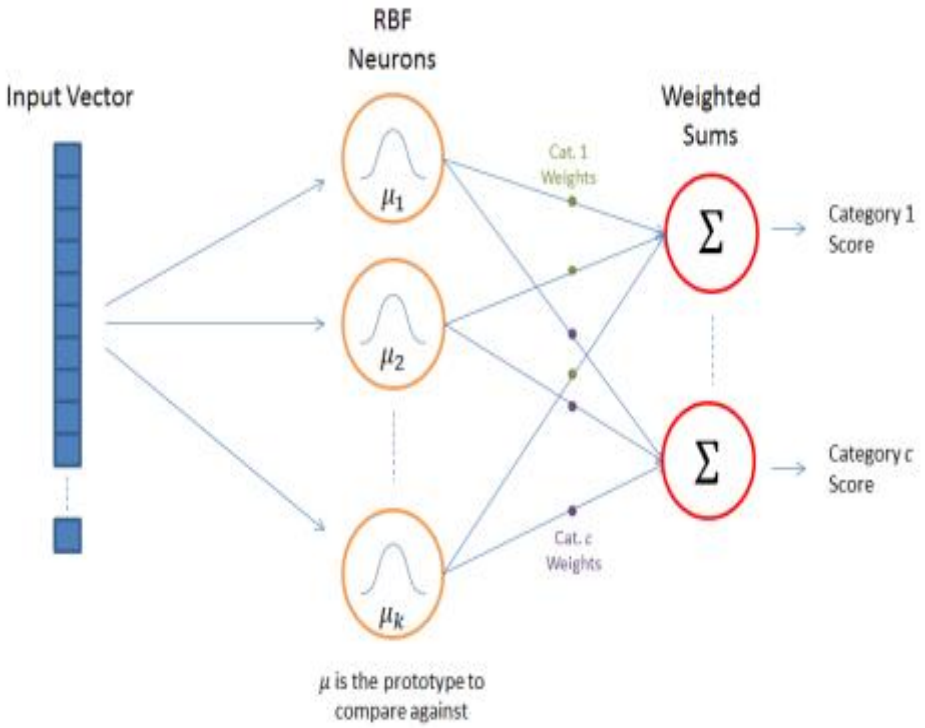


Figure 3.2: Basic RBF Network Architecture

The Input Vector

The input vector is the *n*-dimensional feature vector of the sample or objects that to be classified. The entire input vector is shown to each of the RBF neurons.

The RBF Neurons

Each RBF neuron stores a cluster or "prototype" vector which is calculated from training set feature vectors. Here we basically transform an n dimensional problem to higher m dimensional problem because the probability of classifying an object of which is not separable in lower dimension gets increased when problem is represented in higher dimension. The n dimensional training samples are used to create m cluster using any clustering technique and thus RBF neuron is created. RBF neuron weight is centred mean of the entire sample that falls in that that cluster.

The Output Nodes

The output layer of RBF consists of same numbers of neurons as many numbers of classes is to be identified by it. A classification decision is made by assigning the input to the category with the highest score.

This score is calculated on the basis of weighted sum of the activation values from every RBF neuron. Score is computed for each output node of different category and every output node has its own weights.

RBF Neuron Activation Function

There are different activation functions, but the famous one is based on the Gaussian. Below is the equation for a Gaussian with a one-dimensional input.

$$f(x) = \frac{1}{\sigma\sqrt{2\Pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad (3.11)$$

Where x is the input, u is the mean, and σ sigma is the standard deviation. This produces the following bell curve centred at the mean (in the below plot the mean is 5 and sigma is 1).

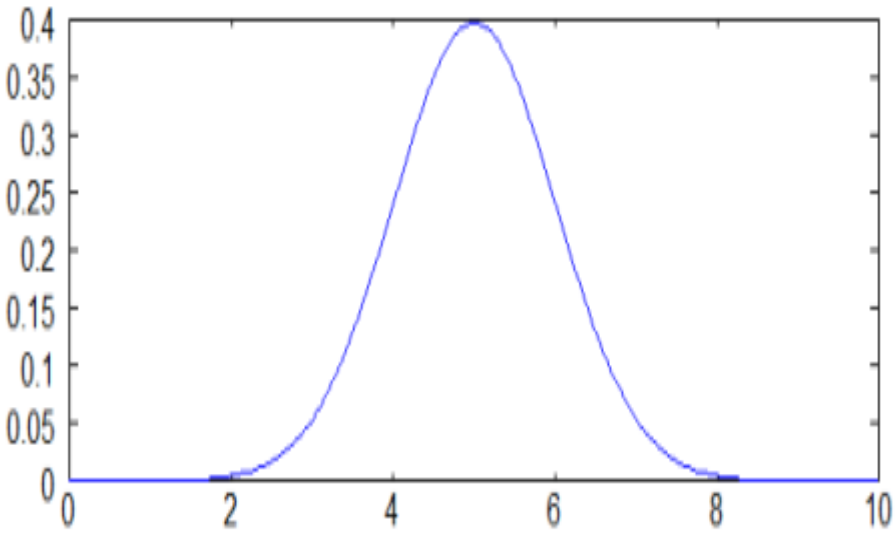


Figure 3.3: RBF Neuron Activation Function plot

The RBF neuron activation function is slightly different, and is typically written as:

$$\Phi(x) = e^{-\beta\|x-\mu\|^2} \qquad (3.12)$$

In the Gaussian distribution, μ refers to the mean of the distribution. Here, it is cluster mean which is at the centre of the bell curve. The coefficient β controls the width of the bell curve.

For the activation function, Φ, we aren't directly interested in the value of the standard deviation σ, so we make a couple simplifying modifications. The first change is that we can remove the outer coefficient $\frac{1}{\sigma\sqrt{2\Pi}}$. This term normally controls the height of the Gaussian. Here, though, it is redundant with the weights applied by the output nodes. During training, the output nodes will learn the correct coefficient or "weight" to apply to the neuron's response. The second change is that we've replaced the inner coefficient $\frac{1}{2\sigma^2}$, with a single parameter β. This β coefficient controls the width of the bell curve. Again, in this context, we don't care about the value of β, we just care that there's some coefficient which is controlling the width of the bell curve. So we simplify the equation by replacing the term with a single variable.

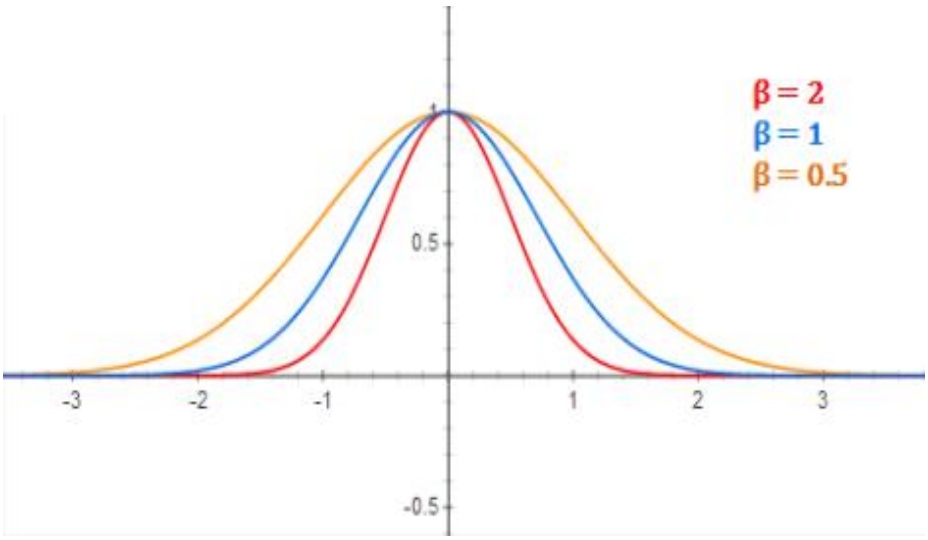


Figure 3.4: RBF Neuron activation for different values of beta

here we are apply this equation to n-dimensional feature vectors that why we are taking the Euclidean distance between x and μ, and squaring the result. $(x-\mu)^2$ is only for 1-dimensional feature vector.

Also, each RBF neuron will produce its largest response when the input is equal to the prototype vector. This allows to take it as a measure of similarity, and sum the results from all of the RBF neurons.

If there is more than one predictor variable, then the RBF function has as many dimensions as there are variables. The following picture illustrates three neurons in a space with two predictor variables, *X* and *Y*. *Z* is the value coming out of the RBF functions:

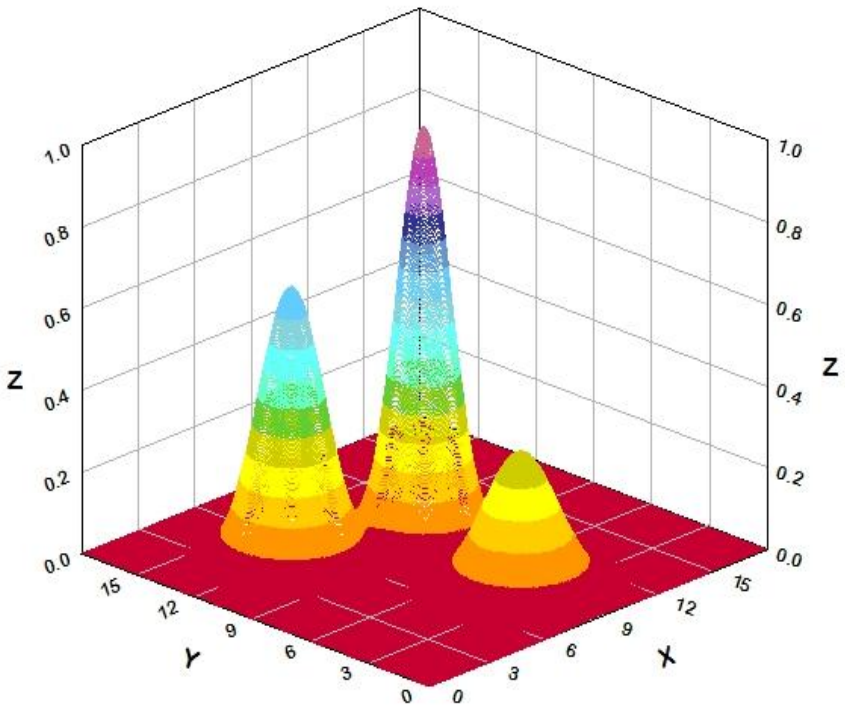


Figure 3.5: Three neurons in a space with two predictor variables *X* and *Y* and Z is activation function value.

The best predicted value for the new point is found by summing the output values of the RBF functions multiplied by weights computed for each neuron.

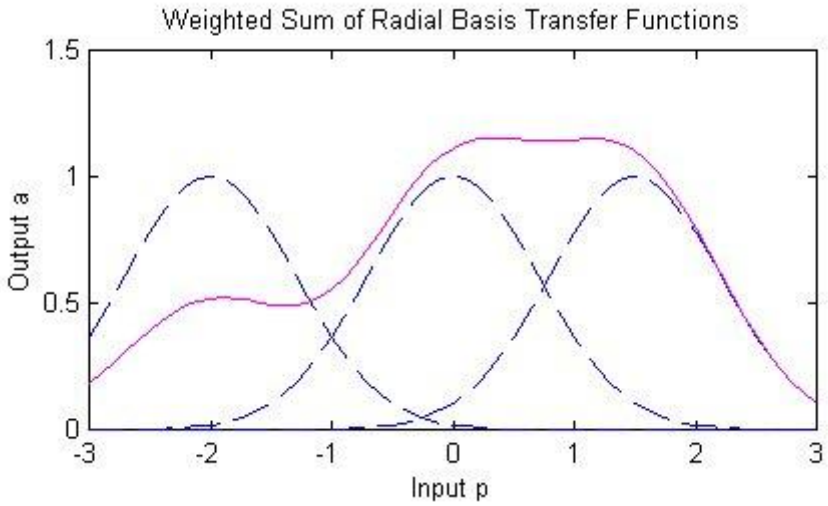


Figure 3.6: Weighted sum of radial basis transfer function

The radial basis activation function for a RBF neuron has a centre and a spread. The radius may be different for each neuron in each dimension. With smaller spread the neuraon at a distance from a point has lee influence because it is less selective while in case of larger spread its vice versa.

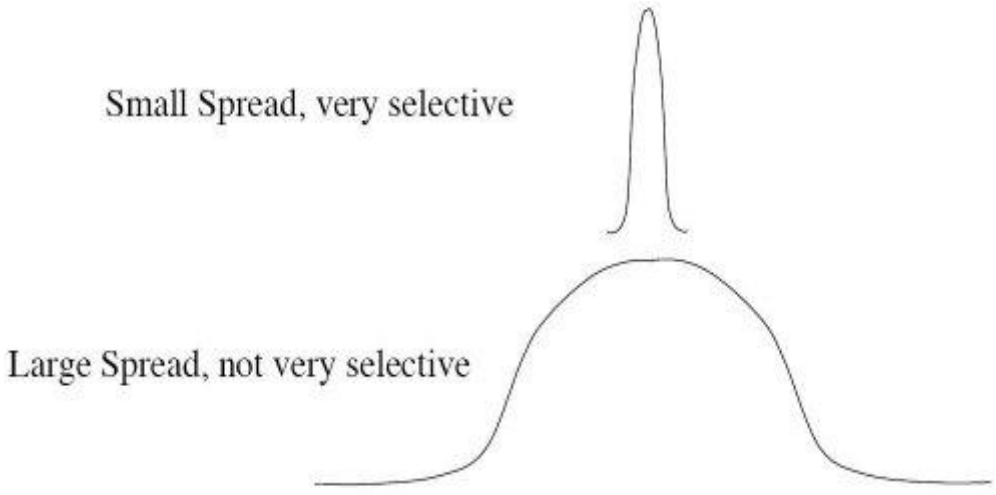


Figure 3.7: Small and large spread in activation function dependent on σ

The Number of Neurons in the Hidden Layers

Deciding number of neurons in hidden layer of RBF neural network is important task as accuracy of this model completely depends upon this factor. Basically we have to decide number of clusters that is to be formed with the help of training feature set.

Too few neurons in the hidden layers cause under fitting of network. It is a situation when too few neurons of hidden layers sufficiently detect the signals in a complicated data set.

Too many neurons in the hidden layers may result in over fitting. It's a situation when the neural network has much capacity to store information but training set is not enough to train all of the neurons in the hidden layers hence these neurons have much less information. On other hand if a sufficient training feature set is provided to train a RBF network with too many neurons in hidden layer it will increase the time complexity of training. Obviously, some compromise must be reached between too many and too few neurons in the hidden layers.

There are some facts that need to be kept in mind while selecting the number of hidden layer neurons mentioned below.

- size of the input layer < number of hidden neurons < size of the output layer.

- number of hidden neurons <= 2/3 * size of the input layer + size of the output layer.

- The number of hidden neurons < size of the input layer.

These three rules helps us to start framing network however the selection of final architecture of particular RBF neural network can be decided on basis of trial and error.

K-means clustering

The K-means clustering algorithm for selection of RBF neuron m dimensional centres is as follows:

STEP 1: Initialize the centres of each cluster to a different randomly selected training feature vector.

STEP 2: calculate distance of each training set to all the clusters and keep each feature set in the cluster whose distance is minimum from it. This distance is calculated as eculidian distance because patterns are not 1 dimensional.

STEP 3: After assigning all the feature set a cluster compute the updated cluster centre of each of cluster.

STEP 4: Repeat steps 2 and 3, until there is a saturation means cluster centres becomes fixed and there is no further change in cluster centre during next iterations.

Selecting Beta Values

First we need to calculate σ. Many researchers have given different way of selecting sigma. Some of them are described bellow.

If you use k-means clustering to select your clusters, then one simple method for specifying the beta coefficients is to set sigma equal to the average distance between all points in the cluster and the cluster centre.

$$\sigma = \frac{1}{m}\sum_{i=1}^{m}\|x_i - \mu\| \qquad (3.13)$$

Here, μ is the cluster centroid, m is the number of training samples belonging to this cluster and x_i is the ith training sample in the cluster.

Another way in which the width of each RBF unit i.e. signa can be calculated using the K-nearest neighbours algorithm. A number K is chosen, and for each center, the K nearest centers is found. The root-mean squared distance between the current cluster center and its K nearest neighbours is calculated, and this is the value chosen for the unit width (σ). So, if the current cluster center is μ_j, the σ value is:

$$\sigma_j = \sqrt{\frac{\sum_{i=1}^{k}(xi-\mu j)^2}{k}} \qquad (3.14)$$

A typical value for K is 2, in which case s is set to be the average distance from the two nearest neighbouring cluster centres.

Once we have the sigma value for the cluster, we compute beta as:

$$\beta = \frac{1}{2\sigma^2} \qquad (3.15)$$

Pseudo-Inverse Technique for selection of weights in output layer

Using the linear mapping, **w** vector is calculated using the output vector (**y**) and the design matrix.

$$\Phi . y = w\Phi \qquad (3.16)$$

$$w = (\Phi^T\Phi)^{-1}\Phi^T y \qquad (3.17)$$

This is based on fact that it is not possible calculate the inverse of a non square matrix so we can obtain a square matrix by multiplying Φ^T with Φ . After this inverse is calculated and multiplied with Φ^T and known output of training set so as to get weights of output layer of RBF network.

3.5 Support Vector Machine (SVM)

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples [19].

How we can say the hyperplane obtained is optimal? Let's consider the following simple problem: For a linearly separable set of 2D-points which belong to one of two classes, find a separating straight line.

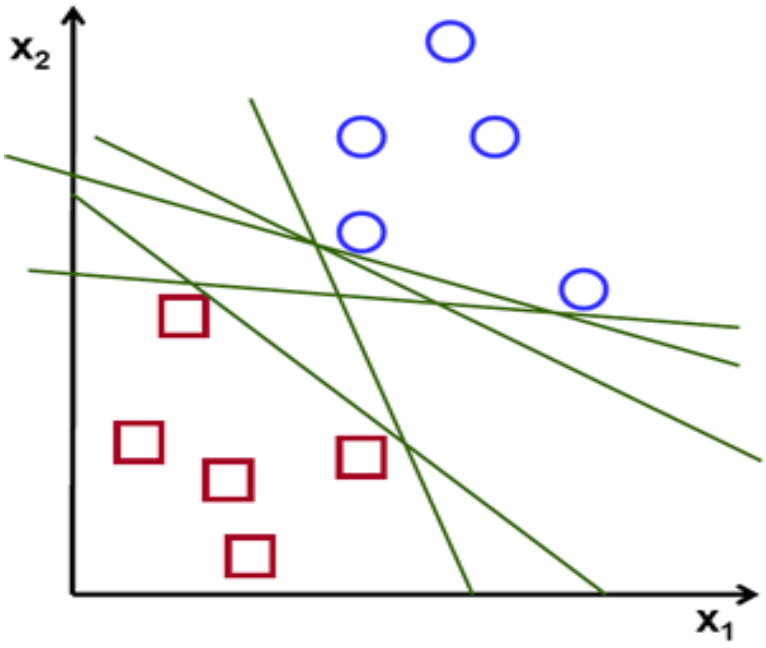


Figure 3.8: Linearly separable two class objects (one circle class and other rectangle class)

In the above picture you can see that there exist multiple lines that offer a solution to the problem. But which is better than the others? A criterion can be define to estimate the worth of the lines. A line is bad if it passes too close to the points because it will be noise sensitive and it will not generalize correctly. Therefore, our goal should be to find the line passing as far as possible from all points [19].

The aim of the SVM algorithm is to find the hyperplane that gives the largest minimum distance to the training examples. Twice this distance is called as margin within SVM's theory. Therefore, the hyperplane that optimaly maximizes the margin of the training data is known as separating hyperplane [19].

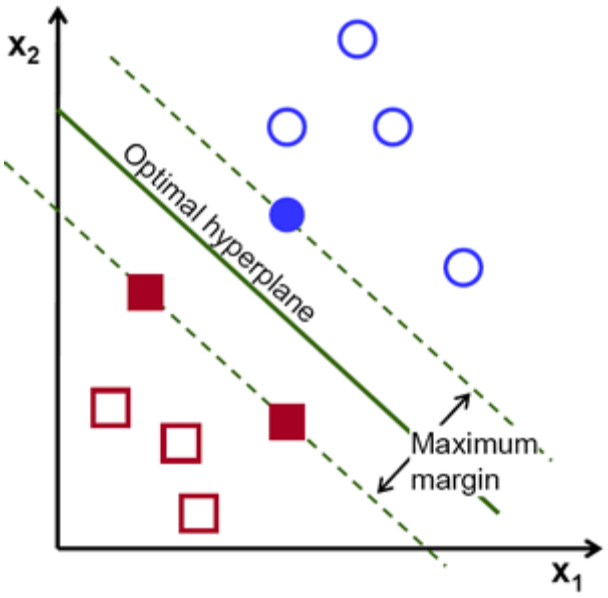


Figure 3.9: Optimal hyperplane separating two class objects (one circle class and other rectangle class)

How is the optimal hyperplane computed? Let's introduce the notation used to define formally a hyperplane:

$$f(x) = \beta_0 + \beta^T x \qquad (3.18)$$

where β is known as the weight vector and β_0 as the bias.

The optimal hyperplane can be represented in an infinite number of different ways by scaling of β and β_0 . As a matter of convention, among all the possible representations of the hyperplane, the one chosen is $|\beta_0 + \beta^T x| = 1$, where $\mathbf{x}$ symbolizes the training examples closest to the hyperplane. In general, the training examples that are closest to the hyperplane are called support vectors. This representation is known as the canonical hyperplane [19].

Now, we use the result of geometry that gives the distance between a point $\mathbf{x}$ and a hyperplane (β,β_0) :

$$distance = \frac{|\beta_0+\beta^T x|}{\|\beta\|} \qquad (3.19)$$

In particular, for the canonical hyperplane, the numerator is equal to one and the distance to the support vectors is

$$distance_{support\ vector} = \frac{|\beta_0 + \beta^T x|}{\|\beta\|} = \frac{1}{\|\beta\|} \qquad (3.20)$$

Here M is margin and it is twice the distance to the closest examples from whole dataset:

$$M = \frac{2}{\|\beta\|} \qquad (3.21)$$

Finally, the problem of maximizing M is equivalent to the problem of minimizing a function $L(\beta)$ subject to some constraints. The constraints model the requirement for the hyperplane to classify correctly all the training examples x_i. Formally,

$$\min_{\beta,\beta_0} L(\beta) = \frac{1}{2}\|\beta\|^2 \text{ subject to } y_i(\beta_0 + \beta^T x) \geq 1 \forall i \qquad (3.22)$$

where y_i represents each of the labels of the training examples [19].

This becomes a problem of Lagrangian optimization that can be solved using Lagrange multipliers to obtain the weight vector β and the bias β_0 of the optimal hyperplane.

Chapter 4

PROPOSED METHODOLOGY

This chapter will describe the proposed methodology in detail the techniques used in different phases. Firstly over all methodology and details of signature data base used are described. Then remaining sections describe the techniques developed in this thesis to select global features and classifier used.

4.1 Methodology

Online signature verification process is mainly consisting of extracting the features of signature, training the system so as to decide classification boundary and at end verification of signature by comparing predefined decision boundary. Many features can be extracted from signature database but selection of most discriminative features means features that can identify genuine signature as genuine and forgery signature as forgery is important for such type of classification system. We have considered both global as well as local features for signature verification.

4.1.1 Process

Figure 4.1 describe our overall methodology for verification of online signature. This is consisting of three phases. First phase is called global feature selection in which 33 global features are extracted from signature database and then mean and variance analysis is applied to rank these features on the basis of their decreasing genuine and forgery signature discriminating property. First top 10 features are selected and then next 11 remaining global features are processed through PCA to generate second set of global features. The second phase is called as local feature selection in which we have extracted positioning features x and y and velocity features v_x and v_y as local features to convert them to similarity features using DTW and ER^2 so as to increase their discriminative power. Third phase is classification where signatures are identified as valid/invalid. This is done using two machine learning methods RBF neural network and SVM classifier.

The input feature vector to RBF neural network and SVM is consisting of combination of features from three methods of features selection.

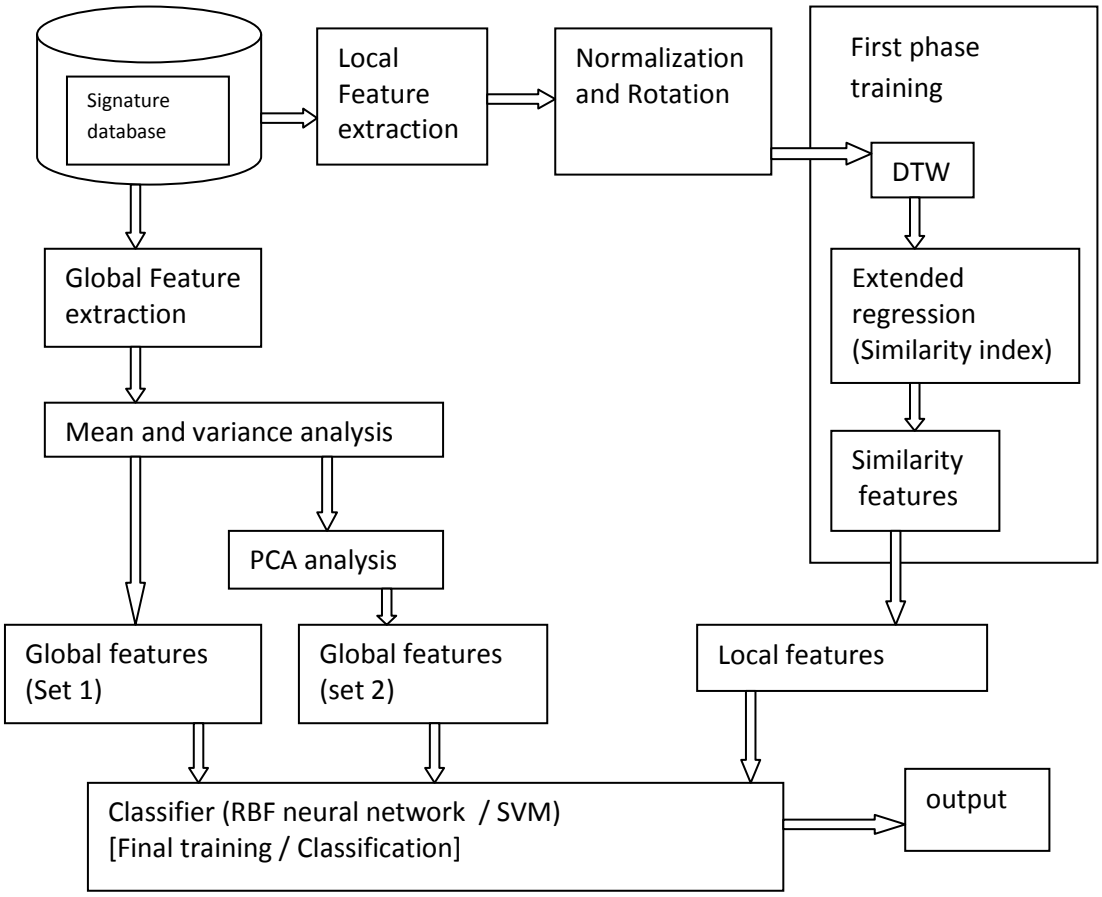


Figure 4.1: Over all methodology for online signature verification system

4.1.2 Signature database

Here is the details of signature database used in system training and testing. Signature database is available on the internet. Signature verification competition 2004 (SVC2004) provided a well designed database which is constructed using WACOM Intuos tablet and includes 40 sets of Chinese and 40 sets of English signature data. Each of them contains 20 genuine signature and 20 skilled forgeries with full information including position, orientation and pressure. The data has been normalized in some level and is almost ready for directly being used by researchers though we don't know much about their collection [6].

The original data has seven columns :

1. X-coordinate - scaled cursor position along the x axis
2. Y-coordinate - scaled cursor position along the y axis
3. Time stamp - system time at which the event was posted
4. Button status - current button status (0 for pen-up and 1 for pen-down)
5. Azimuth - clockwise rotation of cursor about the z axis
6. Altitude - angle upward toward the positive z-axis
7. Pressure - adjusted state of the normal pressure

The position coordinates *X*, *Y* , speed *V*, and the pressure *P* are the most reliable dynamic features while azimuth and altitude have relative high standard deviations [7]. So we will only use the first two columns of the data which are x and y coordinates, and the last one which is the pressure information. The sampling time is 0.01s.

4.2 Global feature selection

Global feature selection process is shown in fig. 4.1. First extract the global features from the database. We considered the following global features of online signature extracted from database.

1. Avg. writing speed ($\bar{v}$)
2. Max. writing speed (v_{max})
3. Time of max writing speed(t(v_{max}))
4. Total signing duration (T_s)
5. Total pen down duration
6. Min. horizontal writing speed
7. time of feature 6
8. number of pen ups
9. Duration of $v_{x>}$ 0
10. Duration of $v_{x<}$ 0
11. Duration of $v_{y>}$ 0
12. Duration of $v_{y<}$ 0
13. Average positive v_x

14. Average negative v_x
15. Average positive v_y
16. Average negative v_y
17. Total v_x = 0 events recorded
18. Total v_y = 0 events recorded
19. Max v_x - Avg v_x
20. Max v_y - Avg v_y
21. Max v_x - Min v_x
22. Max v_x - Min v_y
23. Max v_y - Min v_y
24. t(v_{max}) T_w
25. t(v_{max})/ T_w
26. $(x_{max}-x_{min}) \times (y_{max}-y_{min})=A_{min}$
27. Signature length/ A_{min}
28. $x_0 - x_{min}$
29. $x_{end} - x_{max}$
30. $x_{end} - x_{min}$
31. $(x_{max} - x_{min})/ (y_{max} - y_{min})$
32. Standard deviation of *x*
33. Standard deviation of *y*

Now rank them according to mean and variance analysis described in next section and select top 10 global features. Then take remaining top 11 features i.e. rank 11 to rank 21 features and again rank them according to their PCA analysis, set threshold for selecting the features and select the features crossing threshold limit as described in section 4.2.2. we combined the features obtained from both methods to get final global feature set.

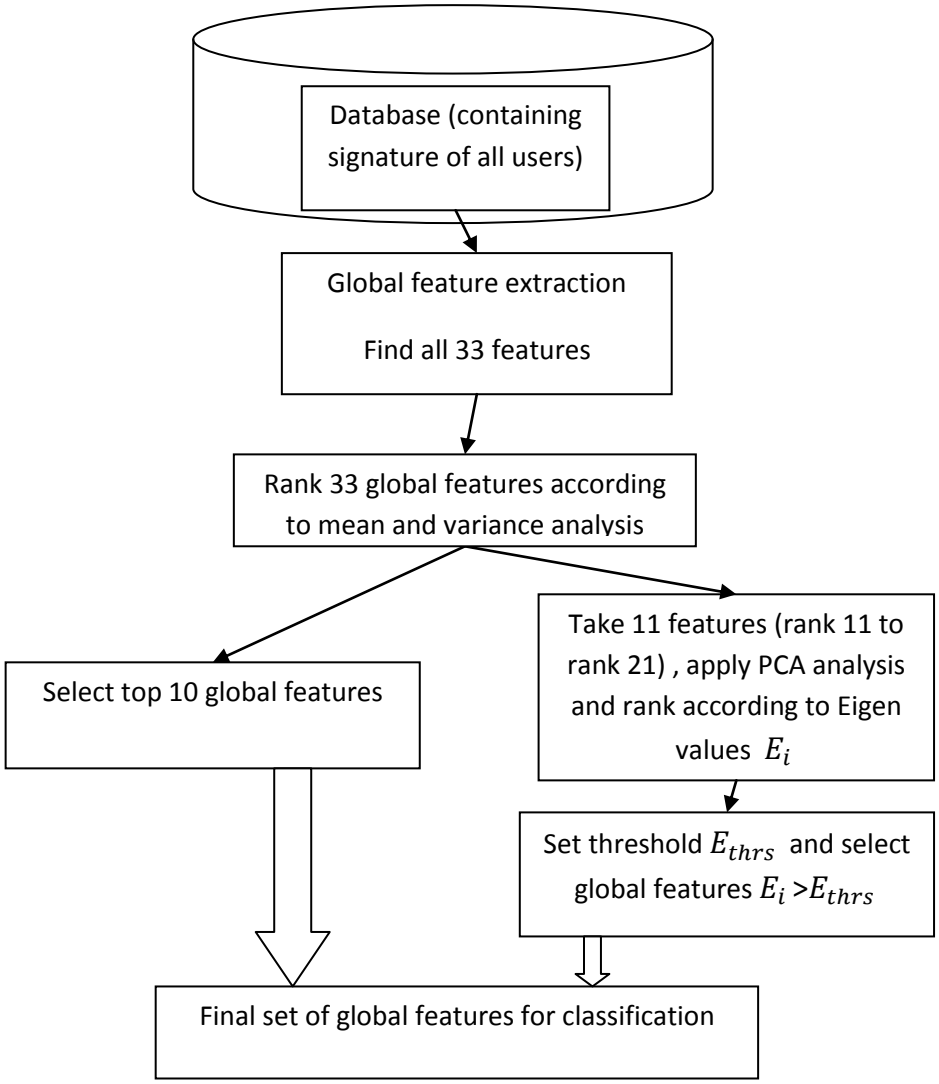


Figure 4.2: Block diagram of global feature selection scheme.

4.2.1 Mean and variance analysis based Global feature selection

The distance of feature i of a user *a* is given by

$$d_i(a) = \frac{|m(a,i) - m(f,i)|}{\sqrt{\sigma^2(a,i) + \sigma^2(f,i)}} \qquad (4.1)$$

Where $m(a,i)$ and $\sigma^2(a,i)$ are mean and covariance of ith feature of user a's signature, and $m(f,i)$ and $\sigma^2(f,i)$ are mean and covariance of feature i computed from database of forgeries of user a's signature [3].

So on comparison with forgery data feature *i* is more important than feature *j* if $d_i(a) > d_j(a)$.

We performed an ordering of 33 global features based on their total distance in all 80 users i.e.

$$D_i = \sum_{a=1}^{80} d_i(a) \qquad (4.2)$$

where $d_i(a)$ represents distance of *i*th feature of user *a* and top 10 features are included here in table below.

s.no	Feature no. (i)	$d_{i\,sum}(j)$	Ranking
1	12	4.1692	1
2	5	4.1462	2
3	4	4.1446	3
4	9	3.9001	4
5	1	3.8533	5
6	11	3.7569	6
7	13	3.6954	7
8	10	3.5541	8
9	14	3.4040	9
10	15	3.2129	10

Table 4.1: Ranking of top 10 global features according to mean and variance analysis.

4.2.2 PCA based Global feature selection

Principal component analysis (PCA) involves a mathematical procedure concerned with elucidating the covariance structure of a set of variables. In particular it allows us to identify the principal directions in which the data varies. Principal component analysis is performed on the symmetric Covariance matrix or on the symmetric Correlation matrix calculated from the data matrix. I solved for the eigen values and eigenvectors of a square symmetric matrix with sums of squares and cross products. The eigenvector associated

with the largest eigen value corresponds to direction of the first principal component of the object means the feature that contribute maximum to identify that object.

We have performed PCA analysis on 80 users taking 20 valid genuine instances of each user with all 11 global features next to those selected above in mean variance analysis according to their decreasing D_i value described above. Than we have calculated E_i average of eigen values of *i*th global feature in all 80 users i.e.

$$E_i = \frac{1}{80}\sum_{a=1}^{80} eigen_i\,(a) \qquad (4.3)$$

where $eigen_i\,(a)$ represents eigen value of *i*th feature of user a . We ranked all these features them according to their decreasing E_i value.

From the table of decreasing E_i of 11 features it was observed that there was a downfall in eignvalues of global features after some feature so the features having low E_i are not important for us to identifying the genuine signature so we can discard them. For selecting top eigen value containing feature we set a threshold for eigen value E_{thrs} and selected the features having $E_i > E_{thrs}$.

ranking	Feature #	Eigen value
1	17	1.4E+13
2	33	629374.1
3	16	11550.21
4	31	2596.941
5	18	855.9975
6	7	14.72024
7	24	2.428199
8	32	0.1048
9	27	0.004083
10	26	0.000805
11	30	2.29E-11

Table 4.2: Ranking of 11 global features according to PCA analysis

4.3 Local feature selection

Pre-processing:

There is scaling and rotational inconsistency in signatures signed by any user because of orientation of writing pad or pen holding or space provided to user for signing. So these inconsistencies in signature need to be resolved using normalization and rotation.

Rotation:

To remove rotation inconsistency we have taken line of regression on signature plot. Line of regression of a curve is a line that have minimum sum of distance from all the points in curve. Than rotated the signature by the angle created by slope of line of regression obtained.

Normalization:

Normalization is done according to following equations.

$$x(n) = \frac{X(n) - X(mean)}{X(max) - X(min)} \qquad (4.4)$$

$$y(n) = \frac{Y(n) - Y(mean)}{Y(max) - Y(min)} \qquad (4.5)$$

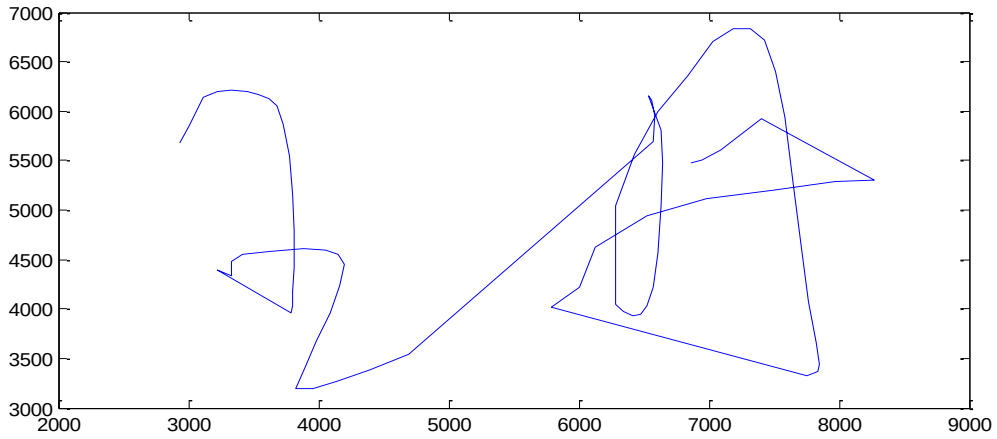


Figure 4.3(a): Signature before scaling and rotation

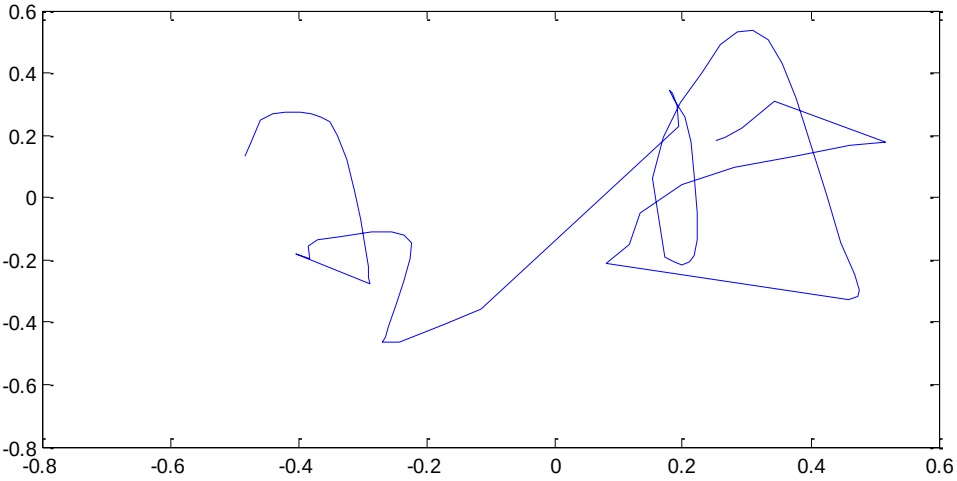
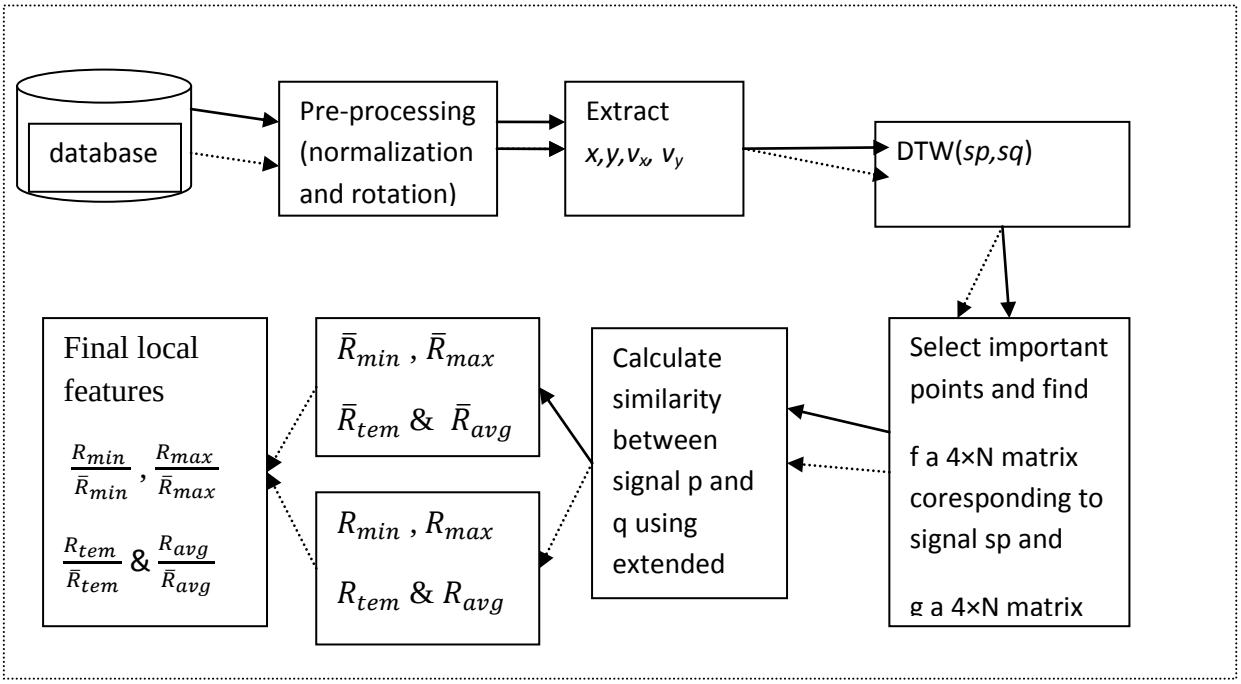


Figure 4.3(b): Signature after scaling and rotation



Here _____ solid line represent first part training with 5 signatures and *p,q* Ɛ {1,2,3,4,5}
Here ---------- dashed line represent final local feature extraction of signature s_m using 5 training signatures and *p* Ɛ *m* and *q* Ɛ {1,2,3,4,5}
Where s_i represents ith signature of user *s* and *r*={1 ,2,3,4,5}.

Figure 4.4: Extraction and transformation of local features *x, y, v_x* and *v_y* into similarity index features using DTW and Extended regression.

For selection of local features we considered *x* and *y* coordinates, velocity in *x* direction v_x and velocity in *x* direction v_y. v_x and v_y will be obtained using following equations.

$$v_x(n) = \sum_{\tau=1}^{3} \frac{x(n+\tau)-x(n-\tau)}{\tau} \qquad (4.6)$$

$$v_y(n) = \sum_{\tau=1}^{3} \frac{y(n+\tau)-y(n-\tau)}{\tau} \qquad (4.7)$$

Since it is almost impossible that a person takes equal duration of time for signing at two different instances, so we need to align two signatures before comparing them .

For this alignment we are using DTW.

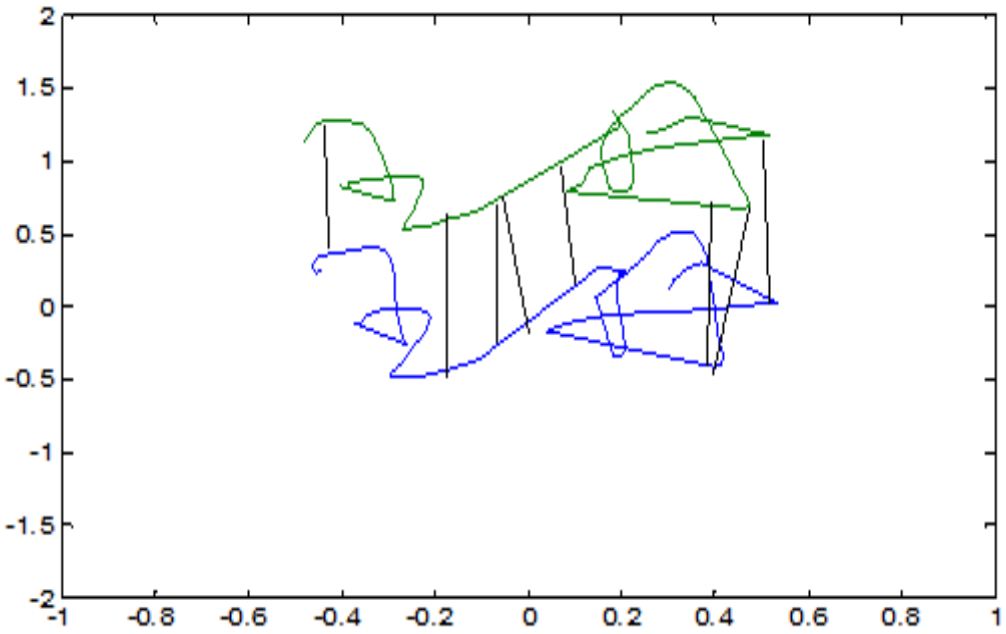


Figure 4.5(a): Alignment of 2 signatures *S1* in green and *S2* in blue

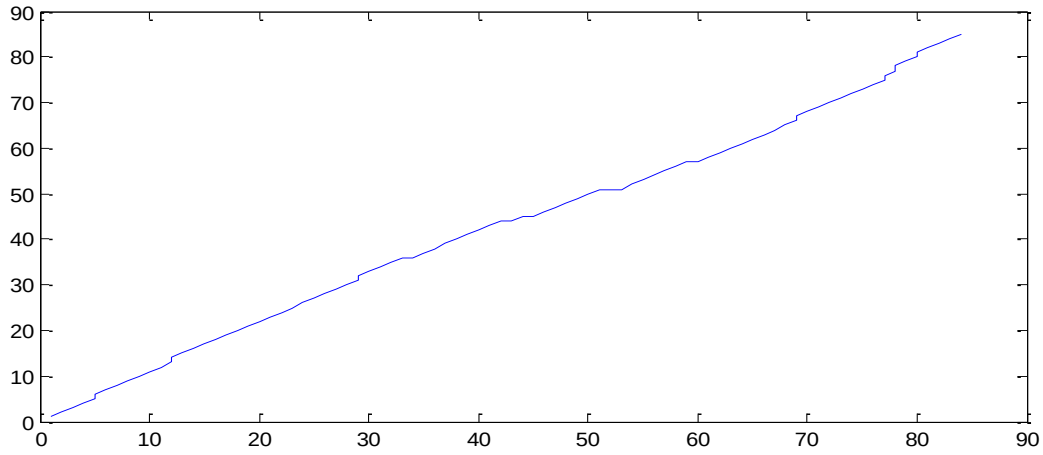


Figure 4.5(b). Warp path of signatures S1 (x axis) and S2 (y axis).

After equalization of related *x*, *y*, v_x and v_y signals of two signatures using DTW, we have extracted the important points in the velocity signals of the signatures using zero-crossing points are defined by following equation.

$$v_x(n) \times v_x(n-1) < 0 \text{ OR } v_y(n) \times v_y(n-1) < 0 \qquad (4.8)$$

we put corresponding important point signals of one signature in matrix g and corresponding signals of other signature in matrix f and calculated the similarity index using Extended regression which gives the similarity between two identical time duration multi dimensional sequences directly [4].

$$\text{Similarity} = \frac{\left[\sum_{j=1}^{M}\left(\sum_{i=1}^{N}(g_{ij}-\overline{G_j})-(f_{ij}-\overline{F_j})\right)\right]^2}{\sum_{j=1}^{M}\left(\sum_{i=1}^{N}(g_{ij}-\overline{G_j})^2\right)\sum_{j=1}^{M}\left(\sum_{i=1}^{N}(f_{ij}-\overline{F})^2\right)} \qquad (4.9)$$

In above mentioned equation, N is number of important points, M=4, g and f are 4×N matrixes.$\overline{G_j}$ and $\overline{F_j}$ indicate average *jth* dimension of f and g sequences and n is signals' length after time duration equalization.

When similarity between two signatures is obtained next is to take proper estimate for this similarity measure, which we have obtained through first step training where we have selected five signature of same user and calculated the similarity between each. This process will provide us how much variation in similarity that can occurs when we are

comparing 2 genuine signatures. So four features are extracted from these similarities which are mentioned below.

We find Minimum and maximum similarity between each signature and other 4 signatures than their averaged valve to get $\bar{R}_{min}$ and $\bar{R}_{max}$ respectively. Template signature or S_T which is signature with minimum distance to other signatures and its average value will be $\bar{R}_{tem}$. And fourth feature will be the averaged similarity between each pair of signature$\bar{R}_{avg}$. Now to create transformed local features ,we will calculate similarity of a signature with 5 signatures that were used to calculate $\bar{R}_{min}$, $\bar{R}_{max}$, $\bar{R}_{tem}$ and $\bar{R}_{avg}$, than we will determine R_{min} , R_{max}, R_{tem} and R_{avg} of this signature. To make feature set to be passed to final classifier we will combine global features selected from Mean and variance analysis and PCA analysis and transformed local features $\frac{R_{min}}{\bar{R}_{min}}$, $\frac{R_{max}}{\bar{R}_{max}}$, $\frac{R_{tem}}{\bar{R}_{tem}}$ and $\frac{R_{avg}}{\bar{R}_{avg}}$ extracted from local features using DTW and extended regression.

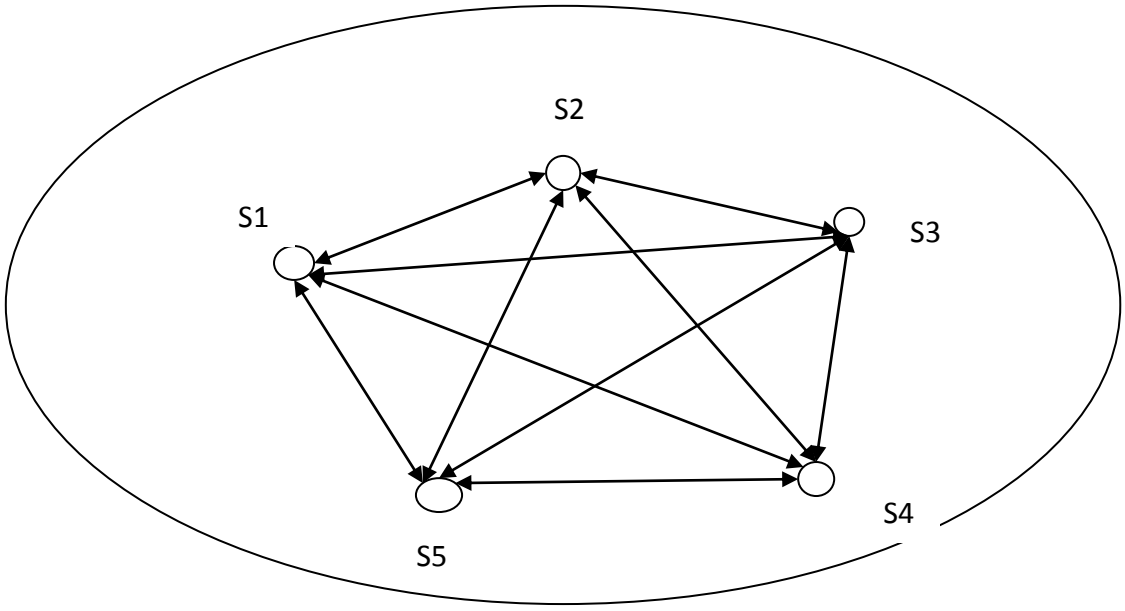


Figure 4.6(a) Calculation of similarity between each pair of 5 signatures. This similarities are used to calculate $\bar{R}_{min}$, $\bar{R}_{max}$, $\bar{R}_{tem}$ and $\bar{R}_{avg}$.

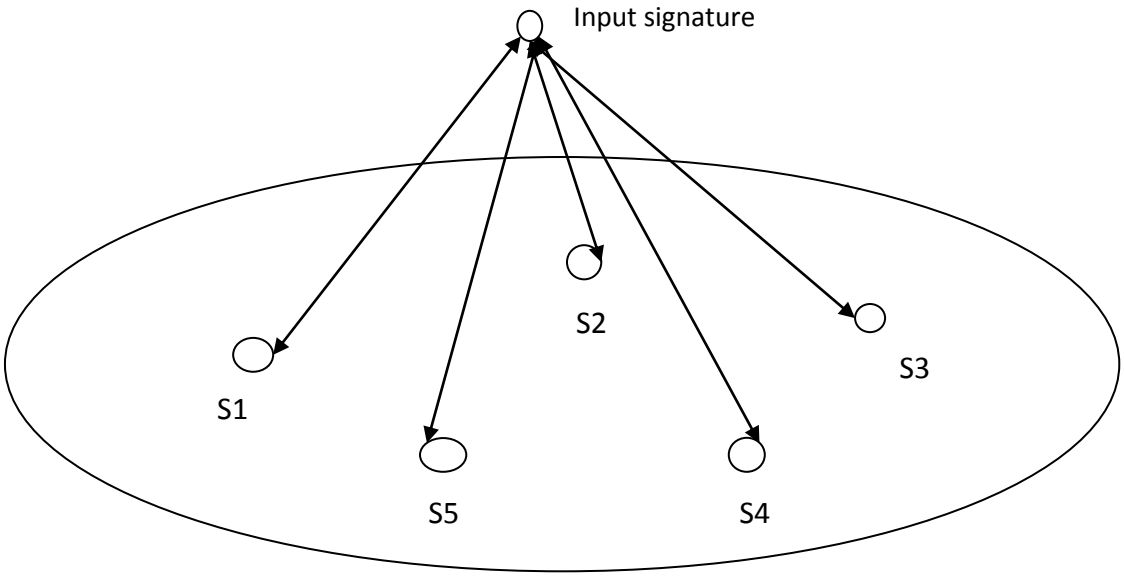


Figure 4.6(b) Calculation of similarity between input and other five 5 signatures. This similarities are used to calculate R_{min} , R_{max}, R_{tem} and R_{avg}.

4.4. Classification

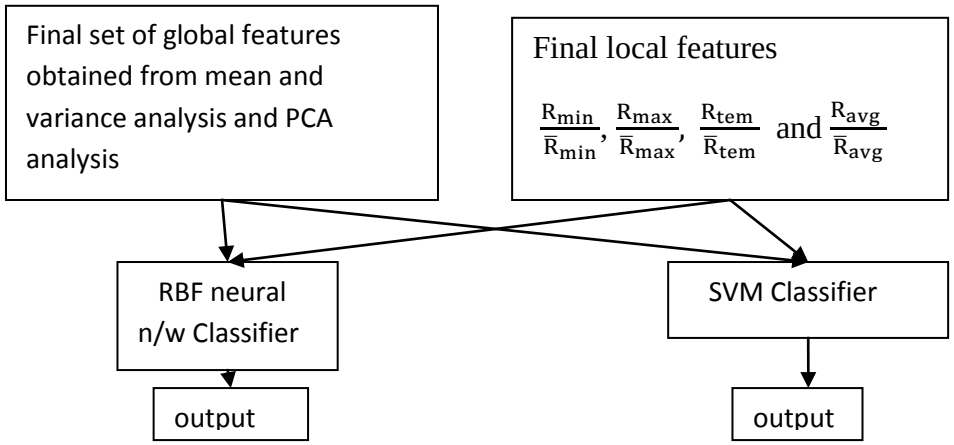


Figure 4.7: RBF neural network classifiers and SVM classifier with combination of global and local input.

We have used RBF neural network and SVM classifier with various combination of global features obtained from mean and variance analysis and PCA analysis and local features $\frac{R_{min}}{\bar{R}_{min}}$, $\frac{R_{max}}{\bar{R}_{max}}$, $\frac{R_{tem}}{\bar{R}_{tem}}$ and $\frac{R_{avg}}{\bar{R}_{avg}}$ for classification of signatures.

4.4.1 Classification using RBF neural network

The RBF network is consist of 3 layers

(1) **Input**: Here in input layer there are as many neurons as many features we are considering for classification. Suppose input to rbf neural network be *x* which is *N* dimensional feature vector.

(2) **Hidden layer**: Here we have used k means clustering for creating hidden layer neuron. We have varied the no. of hidden layer neurons between 4 and 18 depending upon input taken.

Activation function:

$$\Phi(x) = e^{-\beta\|x-\mu\|^2} \qquad (4.10)$$

Where $\beta = \frac{1}{2\sigma^2}$, $\sigma = \sqrt{\frac{\sum_{i=1}^{k}(x_i-\mu i)^2}{k}}$, x_i is ith nearest neighbour from cluster centre and k =2 as have have considerd only 2 nearest neighbours for calculating σ.

(3) **Output**: output values for each set positive i.e. >1 if it is valid signature and negative i.e. <-1 if it is invalid signature.

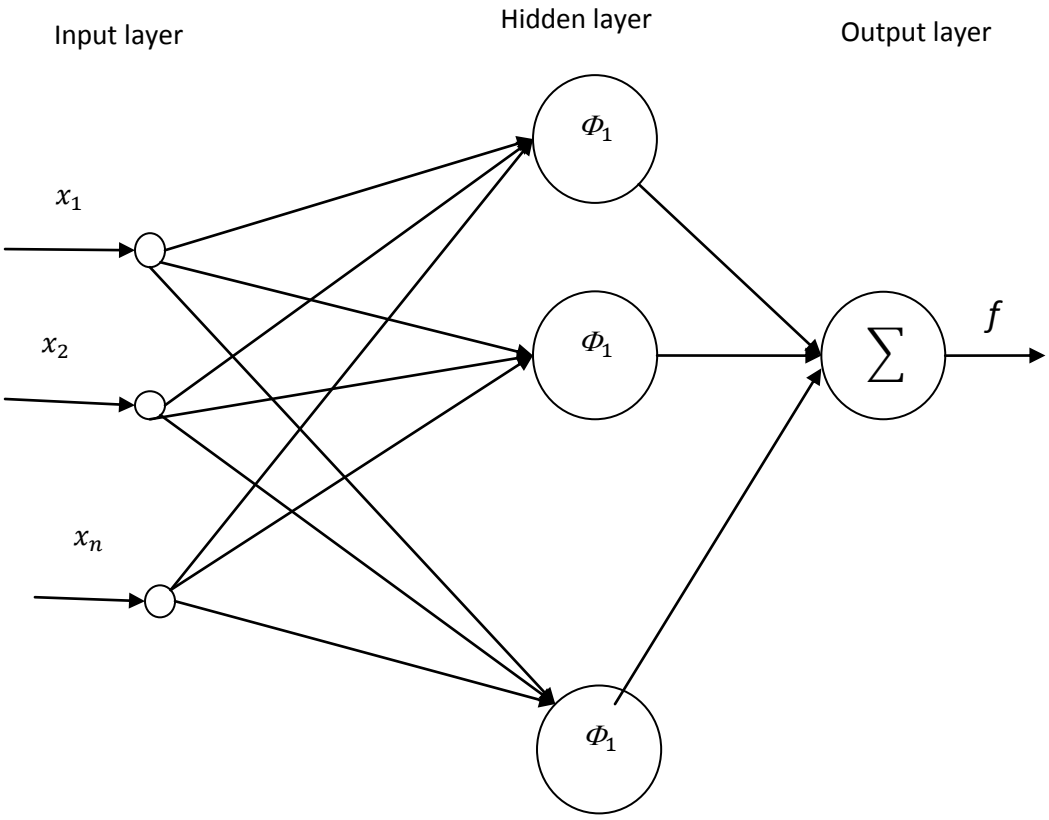


Figure 4.8: RBF neural network architecture for two class problem

4.4.2 Classification using SVM

Support Vector Machine (SVM) performs classification tasks by constructing hyper planes in a multidimensional space that separates instances of different classes. To construct an optimal hyper plane, SVM employs an iterative training algorithm by minimizing an error function hence suited for most classification problems.

We have used support vector machine (SVM) classifier with various combination of global features obtained from mean and variance analysis and PCA analysis and local features $\frac{R_{min}}{\bar{R}_{min}}$, $\frac{R_{max}}{\bar{R}_{max}}$, $\frac{R_{tem}}{\bar{R}_{tem}}$ and $\frac{R_{avg}}{\bar{R}_{avg}}$ and compared the results.

Chapter 5

EXPERIMENTAL RESULTS

The proposed scheme was implemented on Matlab platform and tested using SVC 2004 dataset consisting signatures of 80 users (20 genuine and 20 skilled forgery for each user).

We have taken first 5 genuine signatures of each user for first phase training in order to calculate $\bar{R}_{min}$, $\bar{R}_{max}$, $\bar{R}_{tem}$ and $\bar{R}_{avg}$**.** Next we have taken 15 genuine and 15 skilled forgery signatures for training and rest 5 genuine and 5 skilled forgeries for testing using RBF neural network and SVM.

5.1 Results

We ranked 33 features on the basis of decreasing D_i computed from mean and variance analysis. We have set threshold for selection of global feature on the basis of D_i which is 3.0. This is so because there is a down fall after 3.2129 value. Means top 10 ranking features which are

1. Duration of $v_{y<}$ 0
2. Total pen down duration
3. Total signing duration (T_s)
4. Duration of $v_{x>}$ 0
5. Avg. writing speed ($\bar{v}$)
6. Duration of $v_{y>}$ 0
7. Average positive v_x
8. Duration of $v_{x<}$ 0
9. Average negative v_x
10. Average positive v_y

Below is ranking of 33 global feature according to decreasing D_i .

s.no	Feature no. (*i*)	D_i	Ranking	s.no	Feature no. (*i*)	D_i	Ranking
1	12	4.1692	1	18	32	2.2516	18
2	5	4.1462	2	19	27	2.2116	19
3	4	4.1446	3	20	26	2.1792	20
4	9	3.9001	4	21	30	2.1364	21
5	1	3.8533	5	22	28	1.8398	22
6	11	3.7569	6	23	25	1.7118	23
7	13	3.6954	7	24	3	1.6837	24
8	10	3.5541	8	25	29	1.5749	25
9	14	3.4040	9	26	8	1.5618	26
10	15	3.2129	10	27	22	1.2324	27
11	17	2.6238	11	28	21	1.2312	28
12	33	2.4825	12	29	6	1.1690	29
13	16	2.4449	13	30	19	1.1512	30
14	31	2.4246	14	31	23	1.1219	31
15	18	2.4070	15	32	2	1.0489	32
16	7	2.3518	16	33	20	1.0124	33
17	24	2.3109	17				

Table 5.1: shows ranking of 33 global features according to mean and variance analysis

Now we performed PCA analysis on next 11 features in the list arranged according to D_i and ranked them again according to their averaged eigen value E_i. Threshold for selection of global feature using PCA E_{thrs} was set to 2000 and it was observed that global features no. 17, 33, 16, and 31 has more eigen value than rest of 7 features hence has more significant contribution for identifying true signature than other 7 global feature considered in PCA analysis.

Features selected from PCA analysis are

1. Total $v_y = 0$ events recorded
2. Standard deviation of y
3. Total $v_x = 0$ events recorded

4. $(x_{max} - x_{min)} / (y_{max} - y_{min)}$

We have passed various combinations of local and global features to RBF neural network classifier and SVM classifier, means feature vectors with different dimensions are used as shown below.

Feature selection method	Dimensionality of feature vectors
Using global features extracted from mean and variance analysis	10
Using local features extracted from ER^2	4
Using local features extracted from ER^2 and Using global features extracted from mean and variance analysis	14
Using global features extracted from mean and variance analysis and PCA	14
Using local features extracted from ER^2 and Using global features extracted from mean and variance analysis and PCA analysis	18

Table 5.2: Dimensionality of feature vectors depending on feature selection method

The next table shows results in terms of FRR, FAR and total error when various combinations of global and local features were passed to RBF classifier. We have considered values of output neuron for each set positive i.e. >1 if it is valid signature and negative i.e. <-1 if it is invalid signature. Variation in number of hidden layer is also shown.

Type of features selection	No. of hidden layer	FRR %	FAR %	Total error %
Using global features extracted from mean and variance analysis	7	8.25	3.5	5.875
	8	7.25	2.75	5
	9	7.5	4.5	6
	10	9.25	5.75	7.5
Using local features extracted from ER^2	4	50.25	0.5	25.25
Using local features extracted from ER^2 and Using global features extracted from mean and variance analysis	10	8.75	6.75	7.75
	11	7.5	5.25	6.375
	12	11	6	13
Using global features extracted from mean and variance analysis and PCA	10	8.75	4.75	6.75
	11	8.75	6	7.375
	12	10.5	5.25	7.875
Using local features extracted from ER^2 and Using global features extracted from mean and variance analysis and PCA analysis	12	8.75	5	6.875
	13	7	6.75	6.875
	14	7.75	8.75	8.25
	15	8.5	5.25	7.125
	16	6.25	8	7.375
	17	9.5	7	8.25

Table 5.3: Error rates of various combination features for signature classification using RBF neural network classifier.

Below the table shows results in terms of FRR, FAR and total error when various combinations of global and local features were passed to SVM classifier.

Type of features selection	FRR %	FAR %	Total error %
Using global features extracted from mean and variance analysis	4.75	14.25	9.5
Using local features extracted from ER^2	39.25	0	19.625
Using local features extracted from ER^2 and Using global features extracted from mean and variance analysis	6.75	3.5	5.125
Using global features extracted from mean and variance analysis and PCA	3.25	14.25	8.75
Using local features extracted from ER^2 and Using global features extracted from mean and variance analysis and PCA analysis	6.25	2.5	4.625

Table 5.4: Error rates of various combination features for signature classification using SVM classifier.

5.2 Discussion

First lets discuss about classification using RBF neural network. The results of RBF neural network classification tell us some important criteria for feature selection like classification by use of global features selected by mean and variance analysis give minimum error of 5% in best case with 8 hidden layers and at an average 6% as compared to other features selected from other methods. Hence these features are very

important in distinguishing genuine and forgery signature. Again FAR is minimum in case of taking local features extracted from DTW and ER^2 which is 0.5% , hence we can conclude that these local features are capable of identifying forgery signatures.

From the observations of classification using RBF neural network it can be concluded that number of hidden layers plays important role in classification using RBF neural network and they can be optimized using trial and error.

From the experimental results of classification using SVM we see that FAR using the transformed local features $\frac{R_{min}}{\bar{R}_{min}}$, $\frac{R_{max}}{\bar{R}_{max}}$, $\frac{R_{tem}}{\bar{R}_{tem}}$ and $\frac{R_{avg}}{\bar{R}_{avg}}$ is zero hence these features have capability to identify the forgery signatures. Whereas global features extracted from mean and variance analysis provide quite low 9.5 % error rate as compared to classification using local features only, it means these global features are capable of distinguishing genuine and forgery signature. Also the FRR reported using classification from features selected using mean and variance analysis is 4.75% but if we add features extracted from PCA analysis FRR reduces to 3.25%. Similarly selecting combination of global and local features extracted from mean and variance analysis and ER^2 method observed FRR is 6.75 and here addition of global features extracted from PCA analysis reduces FRR to 6.25. Hence from both the observations we can conclude that selection of feature from PCA helps in finding features that are required to classify a genuine signature. A selection from combination of three methods gives the minimum FAR = 2.5% and minimum ERR=4.625% result.

From all above it is also interesting to know that from SVM classifier we see best result is coming out with the features selected from all 3 methods (ER^2+mean and variance analysis+ PCA analysis) is 4.625% error and on other hand only global features selected from mean and variance analysis gives best result with error 5% when used by RBF neural network. Hence selection of feature is also dependable to type of classifier to be used.

The result of this experiment (classification Using local features extracted from ER^2 and Using global features extracted from mean and variance analysis + PCA analysis) using

SVM classifier and some related work of other researchers who have reported their results on SVC2004 are presented in the table 5.5.

s.no	Signature verification system	Error (%)
1	The proposed algorithm	4.625
2	H.Lei, S.Palla, V.Govindaraju, "ER2: An intuitive similarity measure for on-line signature verification" IWFHR '04: Proceedings of 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR'04), IEEE Computer Society, pp.191-195, Tokyo, October 2004	14.21
3	The First International Signature Verification Competition, 2004, (SVC2004), available in http://www.cs.ust.hk/SVC2004.	5.50
4	M. Adamski, Kh. Saeed, "Online signature classification and its verification system", Proc. IEEE, 7-th Computer Information Systems and Industrial Management Applications (CISIM'08), pp.189-194, June 2008.	7
5	A. Flores-Mendez, M. Bernal-Urbina, "Dynamic Signature Verification through the Longest Common Subsequence Problem and Genetic Algorithms", Proc. IEEE, Evolutionary Computation Conferences (CEC), Barcelona, pp. 1-6, July 2010.	10.63

Table 5.5 Error rates for comparison with some other methods

Our online signature verification system's error rate is 4.63% which is marginally higher than error rate of method proposed by M. Saeidi, R. Amirfattahi, A. Amini, M. Sajadi [10] which is 4.3% but our methodology is simpler as we are using DTW which is much simpler than ant colony optimization.

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

From the experimental results it is observed that FAR is minimum in case of taking local features extracted from DTW and ER^2, hence we can conclude that these local features are capable of identifying forgery signatures. Whereas global features give less error rate as compared to local features when used for classification hence global features are capable of distinguishing genuine and forgery signature. Addition of global features extracted from PCA analysis reduces FRR Hence we can conclude that selection of feature from PCA helps in finding features that are required to classify a genuine signature.

From the observations of classification using RBF neural network it can be concluded that number of hidden layers plays important role in classification using RBF neural network and they can be optimized using trial and error.

A selection from combination of three methods gives the minimum FAR = 2.5% and minimum total error =4.625% result when SVM classifier is used and on other hand only global features selected from mean and variance analysis gives best result with error 5% when used by RBF neural network. Hence selection of feature is also dependable to type of classifier to be used. Here in signature verification purpose SVM classifier is giving better result than RBF neural network classifier.

Hence from all above discussion we have concluded that selection of features and classifier for verification of online signature is very important criteria and can adversely affect the system when not full filled in optimized way.

6.2 Future work

In our proposed system we have used DTW for time varying signal alignment of signatures which reduces the boundary of genuine and forgery signatures. We have tried to overcome this problem with selection of global features with mean and variance analysis and PCA analysis. There are some other methods to align time varying signature signals like ant colony based alignment [10], LCSS based alignment [11] etc. which maintain the decision boundary between genuine and forgery signatures. In future our methodology of selecting global features can be combined with some of these time alignment techniques for optimized selection of features. Hence classifier could be designed so as to optimize the accuracy of online signature verification system and reliability of identity verification using such authentication system can be increased.

APPENDIX

ABBREVIATIONS USED

Avg.	Average
DTW	Dynamic Time Warping
EER	Equal error rate
ER^2	Extended Regression
FAR	False acceptance rate
FRR	False rejection rate
LCSS	Longest common sub-sequences
Max.	Maximum
Min.	Minimum
PCA	Principal Component analysis
RBF	Radial basis function
SVM	Support vector machine

REFERENCES

[1] H.Touryalai. [Online]. Available: http://www.forbes.com/sites/halahtouryalai/2012/10/22/countries-with-the-most-card-fraud-u-s-and-mexico/.

[2] J. F. M. R. F. a. J. O.-G. Javier Galbally, "Feature Selection Based on Genetic Algorithms for On-Line Signature Verification".

[3] T. B. E. A. L.L.Lee L, "Reliable online human signature verification systems," *Pattern Analysis and Machine Intelligence, IEEE Transactions on (Volume:18 , Issue: 6)*, pp. 643 - 647, JUNE 1996.

[4] S. P. V. G. Hansheng Lei, "ER2: an intuitive similarity measure for on-linesignature verification," in *IEEE, Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop*, 2004.

[5] M. M. a. H. M. A. Mostafa I. Khalil, "ENHANCED DTW BASED ON-LINE SIGNATURE VERIFICATION," in *Image Processing (ICIP), 2009 16th IEEE International Conference*, Cairo, 2009.

[6] X. W. C. X. Yu Qiao, "Learning Mahalanobis Distance for DTW based," in *Information and Automation (ICIA), 2011 IEEE International Conference*, Shenzhen, 2011.

[7] O. M.-H. L. M.-P. M. G. L. J. Liu-Jimenez, "On-Line Signature Verification by Dynamic Time Warping and Gaussian Mixture Models," in *IEEE*, 2007.

[8] O. M.-H. R. B.-G. F.-J. D.-J. Aitor Mendaza-Ormaza, "Analysis of handwritten signature performances using mobile devices," in *Security Technology (ICCST), IEEE International Carnahan Conference*, Barcelona, 2011.

[9] A. F. F. T. Saeid Rashidi, "Authentication Based on Signature Verification Using Position, Velocity, Acceleration and Jerk Signals," in *IEEE, Information Security and Cryptology (ISCISC), 9th International ISC Conference*, Tabriz, 2012.

[10] R. A. A. A. M. S. M. Saeidi, "Online signature verification using combination of two classifiers," in *IEEE, Machine Vision and Image Processing (MVIP)*, Isfahan, 2010.

[11] T. G. S. K. a. B. S. Christian Gruber, "Online Signature Verification With Support Vecto rMachines Based on LCSS Kernel Functions," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS*, Vols. VOL. 40, NO. 4, pp. 1088 - 1100 , 2010.

[12] J. C. G. a. M. L. Marianela Parodi, "Online Signature Verification Based on Legendre Series Representation.," in *IEEE, Frontiers in Handwriting Recognition (ICFHR), International*

Conference, Bari, 2012.

[13] H. M. R. Y. a. O. O. Marzuki Khalid, "Online Signature Verification with Neural Networks Classifier and Fuzzy," in *IEEE*, Bali, 2009.

[14] R. Y. a. H. M. Marzuki Khalid, "FUSION OF MULTI-CLASSIFIERS FOR ONLINE SIGNATURE VERIFICATION USING FUZZY LOGIC INFERENCE," in *International Journal of Innovative*, 2011.

[15] K. K. GURRALA, "ONLINE SIGNATURE VERIFICATION TECHNIQUES".

[16] M. Muller, "Dynamic Time Warping," in *information retrival for music and motion*, springer, 2007.

[17] J. P. P. B. K. C. Rajeev Kumar Chauhan, "BIOMETRIC APPLICATIONS FOR PUBLIC SAFETY: A BRIEF SURVEY," Cyber Times International Journal of Technology & Management, 2012.

[18] M. Müller, "Dynamic Time Warping," in *Information Retrieval for Music and Motion*, Springer Berlin Heidelberg, pp. 69-84.

[19] "Introduction to Support Vector Machines," opencv, [Online]. Available: http://docs.opencv.org/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html.

[20] S. Sarl, "Saista Sarl," [Online]. Available: http://www.saista.com/html/signature.html.

[21] J. Z. a. S.-i. KAMATA, "Online Signature Verification Using Segment-to-Segment Matching".

[22] R. S. P. M. a. A. N. Mayank Vatsa, "Signature Verification Using Static and Dynamic Features," in *Springer Berlin Heidelberg*, 2004.

[23] N. B. a. M. Verleysen, "On the Kernel Widths in Radial-Basis Function Networks",," Kluwer Academic Publishers, 2003.

[24] L. Behera, " Radial Basis Function Networks," Department of Electrical Engineering Indian Institute of Technology, Kanpur..

[25] J. A. Bullinaria, *Radial Basis Function Networks: Applications.*

[26] M. A. R.M.Chandrasekaran, "Classifier Based Text Mining For Radial Basis Function," in *7th WSEAS Int. Conf. on artificial intelligence, knowledge engineering and data bases (aiked'08), University of Cambridge, UK*, 2008.

[27] K. S. a. S.N.Deepa, " A New algorithm to find number of hidden neurons in Radial Basis Function Networks for Wind Speed Prediction in Renewable Energy Systems," in CEAI, 2013.

[28] J. S. a. R.J.F.Dow, “Neural Net Pruning - Why and How”.

[29] G.-B. H. a. N. Sundararajan, “A Generalized Growing and Pruning RBF (GGAP-RBF) Neural Network for Function Approximation,” *IEEE transactions on neural networks,* vol. 16, JANUARY 2005.

[30] R. Reed, “Pruning Algorithms – A Survey,” *IEEE transaction on neural networks,* vol. 4, 2005.

[31] M. G. A. J. J. M. a. A. P. Victor M. Rivas, “EvRBF: Evolving RBF Neural Networks for classification problems”.

[32] [Online]. Available: http://www.anc.ed.ac.uk/rbf/rbf.com.

[33] [Online]. Available: http://imp_rbf.com.

[34] [Online]. Available: http://PerceptronWikipediathefreeencyclopedia.com.