Automatic signature verification system

Raghuram Malladi

A Thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science at the Faculty of Natural Sciences, University of the Western Cape

Supervisor: Prof. Kailash C. Patidar

November 2013

KEYWORDS

Signature verification system

Hand written signatures

Dynamic signature verification

Hidden Markov Models

Multiple single variate models

Feature extraction

Biometrics

Time series modelling.

ABSTRACT

Automatic signature verification system

R. Malladi

PhD thesis, Department of Computer Science,

Faculty of Natural Sciences, University of the Western Cape.

In this thesis, we explore dynamic signature verification systems. Unlike other signature models, we use genuine signatures in this project as they are more appropriate in real world applications. Signature verification systems are typical examples of biometric devices that use physical and behavioral characteristics to verify that a person really is who he or she claims to be. Other popular biometric examples include fingerprint scanners and hand geometry devices. Hand written signatures have been used for some time to endorse financial transactions and legal contracts although little or no verification of signatures is done. This sets it apart from the other biometrics as it is well accepted method of authentication. Until more recently, only hidden Markov models were used for model construction. Ongoing research on signature verification has revealed that more accurate results can be achieved by combining results of multiple models. We also proposed to use combinations of multiple single variate models instead of single multi variate models which are currently being adapted by many systems. Apart from these, the proposed system is an attractive way for making financial transactions more secure and authenticate electronic documents as it can be easily integrated into existing transaction procedures and electronic communications.

November 2013.

DECLARATION

I declare that *Automatic signature verification system* is my own work, that it has not been submitted before for any degree or examination at any other university, and that all sources I have used or quoted have been indicated and acknowledged by complete references.

Raghuram Malladi November 2013

Signed . . . .

ACKNOWLEDGEMENT

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Kailash. C. Patidar who encouraged me when I was finding the things very difficult and appreciated whenever I have done good job during the course of the project. Secondly, I would like to thank some key personal at the University of the Western Cape. This include Prof. Lorna Holtman, other colleagues working at the division of post-graduate studies, and other families and friends at UWC; salute to all.

I thank my friend Pavan Kumar Rallabandi for many stimulating discussions, especially during those sleepless nights when we were working together before deadlines, and for all the fun we have had despite very difficult circumstances at times.

I can not afford to forget my family, in particular, my parents for giving birth to me at the first place and supporting me spiritually throughout my academic career. My wife's support in this last journey of my thesis work is unforgettable. She was actually very inspirational.

DEDICATION

This thesis is dedicated to

my Mother

for

her love, endless support and encouragement she gave me all the way of my journey

and to

my Late Father

who was truly a great father. I miss you dad.

Contents

List of Tables

List of Figures

List of Publications

We are busy finalizing following papers which will soon be submitted for publication in international journals:

1. R.R. Malladi and K.C. Patidar, Optimal feature selection for automatic signature verification.
2. R.R. Malladi and K.C. Patidar, Application of complexity measure of signatures to determining the tolerance of signature acceptance/rejection.

Chapter 1

General introduction

Last century has seen a tremendous growth, both in the populations as well as in resources. The identification of many human individuals therefore does not remain as simple as it was in olden days. As the society is getting more and more modernized requirement of unique identity is growing. In the olden days, people were recognized with their surnames. But as the families grew, there are more members and therefore the use of surnames as their identity was not enough. This was the time when researchers realized that there is crucial need for a special identity. Hence, at this second stage, the differentiation was done based on their physical appearance, for example, with their skin colour, height and colour of eyes, etc. However, such physical aspects were just not enough for making a clear distinction among different people. Some preliminary algorithms were designed to study such problems scientifically. However, solving such algorithms was another challenging task. With the advent of technology in past few decades, whole scenario has completely changed. As the advanced computational tools came into the picture, it became a bit easier for the identification of a specific person.

Having a perfect identity was always an issue. When some organizations, for example, banks, insurance companies, etc., needed to identify some individuals, they used to assign passwords. However, there were two major issues with this approach: either these passwords were easily cracked or majority of these individuals just forgot them. As an immediate remedy, some other identification systems, such as passports or social security numbers (both of which are unique) were introduced. The problem was not fully resolved at this stage because even these passports can be manipulated or stolen.

To overcome all these above mentioned problems, biometrics came into the picture. Biometric process is used by the forensic people from many years now to find out the criminals. Biometrics is a science that examines and quantifies unique biological traits to verify the identity of an individual. There are many biological features of individuals which are unique that is all the idea behind the modern biometrics. The small things were then looked up in a new aspect. Signature of a person is as unique as the voice of that person. Deriving the fact that the signature of a person will be saved as a image in the brain through which from holding of pen till signing on the paper will be done in a specific manner. These are all done by the signals from brain. Thus by measuring the signals of the brain, one can make a unique identity of an individual.

In this thesis, we look into different aspects of modern biometrics that are used to give a unique identity individuals. We investigate various aspects involved in the development of an automated signature verification system. In order to better understand the role of signature verification, we first take a look at the notion of biometrics. To this end, one may note that the signature creation is a dynamic time varying process which can be measured by modern hardware. This has resulted in possibly a number of parallel sampled signals each describing some aspect of the signing process. Note that no writer can succeed in exactly duplicating a signature despite several successive attempts. This leads to a variance in the signal profiles of different signature exemplars of a single writer. Mathematical signature models are therefore used to better understand these variances. In order for not to misinterpret future signing attempts as forgeries, these models need to be solved efficiently. Moreover, acceptance of variances in authentic signatures must not lead to acceptance of forgeries beyond a required minimum performance level.

1.1 Objectives of this thesis

The main objective of this thesis is to apply theory of hidden Markov models to the application of dynamic signature verification with the hope of creating a signature model with similar or better performance. We apply the famous Neyman Pearson algorithm to compute the optimal operating points on the receiver operating characteristic (ROC) curves, i.e., for a given selection of features and a chosen acceptable false alarm rate (i.e., the probability of rejecting a genuine signature), the algorithm computes the optimally achievable probability of detecting a fraudulent signature. This gives the user control over the system performance in a rigorous fashion. It also allows us to rank features according to their power of discrimination. Then, apply the support vector machine whether to accept or reject the signature. We will present results of extensive experiments which prove the feasibility of the proposed solution.

Hidden Markov models lie at the heart of our signature verification approach. Indeed, these are standard models used in automatic speech recognition (see, e.g., [30, 65, 91, 156]). However, they can in principle model any non-chaotic time-varying system. They provide us with a great deal of control over various aspects of a model and have the ability to learn from examples. Given the variation in the consistency of different individuals signature, model is imperative for creating an automated signature verification system with the ability to adapt to the signatures of different users given samples of their signatures. We investigate a number of different semantic models in search of a suitable signature model which lends itself to efficient and effective automated signature verification.

1.2 Signature verification as a biometric

Unlike the biometrics approach discussed above which identify an individual by physical attributes, signature verification measures an action of an individual which can be repeated. As Schmidt [119] stated, a signature contains special stroke sequences which are not used in ordinary handwriting. These shapes evolve from routine and training and from the conscious and unconscious influence of the rule to create a unique and individual signature. Signature verification systems rely on the assumption that a person can reproduce his/her signature fairly consistently: it is difficult for a forger to simultaneously duplicate the overall signature appearance, writing speed, force on the pen tip, and the angle with which the pen are held. Wu *et al.* [143] confirmed this by arguing that imitating either overall shape or dynamics of a signature is achievable, but to achieve both is difficult. The imitator is not likely to construct a similar overall shape of a signature without showing his hesitation in the waveform of the writing velocity.

Visual examination of a signature is unreliable for authentication. Untrained human eyes can hardly analyze detailed writing features [143]. The advent of hardware able to measure writing dynamics opened the way for more detailed measurement of the signing process. In addition to the final signature image, several time varying aspects of signatures can be recorded. The analysis of these signals is called dynamic signature verification. Handwritten signatures have been used for some time to endorse financial transactions even though little or no verification of the signatures is done. This sets it apart from other biometrics as it is a well-accepted method of authentication. It is therefore a particularly attractive solution for making financial transactions more secure; it can more easily be integrated into existing transaction procedures. Although current signature verification systems are not as reliable as some other biometrics such as fingerprints and iris scans, even less than perfect authentication performance can reduce the financial losses incurred by credit card companies due to fraud.

Development of commercial products targeted at signature verification such as the technologically advanced SMARTPEN [125] underlines the importance of this biometric. This instrumented pen measures accelerations and pen angles during the signing process. The perception is that there is definite commercial value in developing automated static signature verification is concerned only with the analysis of captured signature images. Stress, illness and intake of neuromuscular stimulants can in the signing process. A signature can also evolve and change over the lifetime of an individual. This dynamic nature of a handwritten signature sets it apart from many other biometrics and poses a somewhat different set of challenges to researchers. This brings us to the problem statement addressed by the work presented in this thesis.

1.3 Literature review

Automatic signature verification has become very popular nowadays. It consists of feature selection, feature extraction and modelling of features. The key contributor is the biometric identification system. This system is very popular and has received significant attention of many researchers. Below we present a critical review of some of the most recent work in this field.

A signature verification algorithm proposed by Lee *et al.* [73] was based on a segment-to-segment matching is studied to find the segment-to-segment correspondence. Here, the geometric extrema were used as segmenting boundaries with two reasons: extrema are reproduced with high stability and the properties of extrema are useful in finding segment-to-segment correspondence. A set of rules for legitimate correspondence between extrema is defined based on their properties, which is utilized for the similarity evaluation between the segments. Dynamic programming is then applied to find an optimal correspondence map. Their experiments revealed that the proposed algorithm is assessed to be effective in improving discriminative ability and also shown that the overall performance is further improved by combining the proposed method with the traditional global parametric algorithm.

Teoh *et al.* [130] introduced cancelable biometrics to denote biometric templates that can be canceled and replaced. They mentioned that the disadvantage of BioHash (a form of cancelable biometrics) is its great decline in performance when the authentic token was stolen and used by the fraud to claim as the legitimate user. In this work, they employed a modified probabilistic neural network as the classifier to alleviate this problem. They tested their experiments on the FERET face data set and obtained promising results.

Signature and voice characteristics, facial features, and iris and fingerprint patterns have all been used (see, e.g., Doroteo *et al.* [31]) to identify a person or just to verify that the person is who (s)he claims to be. The work in [31] was intended to promote user-centered design and evaluation of biometric technologies. To this end, these authors have developed a platform to perform empirical evaluations of commercial biometric identity verification systems, including fingerprint, voice and signature verification.

Yao *et al.* [158] presented a novel approach based on feature level biometrics fusion. They combined two kinds of biometrics: one is the face feature which is a representative of contactless biometrics, and another is the palmprint feature which is a typical contact biometrics. They extracted the discriminant feature using Gabor-based image preprocessing and principal component analysis techniques. Then they designed a distance-based separability weighting strategy to conduct feature level fusion. The experimental results, which they conducted by using a large face database and palmprint database as the test data, showed that their approach significantly improves the recognition effect of single sample biometrics problem, and there is strong supplement between face and palmprint biometrics.

One of the major problem in off-line signature verification is to solve non-linear rotation of signature patterns. Wen *et al.* [141] described about the two models which used rotation invariant structure features to tackle the problem. In principle, the elaborately extracted ring-peripheral features are able to describe internal and external structure changes of signatures periodically. In order to evaluate match score quantitatively, discrete fast fourier transform is employed to eliminate phase shift and verification is conducted based on a distance model. These author's evaluated the similarities between test signature and training samples ring-hidden Markov model was constructed. After all the analysis they proposed a selection strategy to improve the performance of system. The results indicated that this method was effective to make better the verification accuracy.

In [21], Broek proposed a new class of biometrics that is founded on processing biosignals, as opposed to images. He indicated that for the past 40 years the significance of automated verification of users has remained the same. In this paper, after a brief introduction on biometrics, he discussed biosignals, including their advantages, disadvantages, and guidelines for obtaining them. Then he illustrated the use of biosignals by considering two biosignal-based biometrics: voice identification and handwriting recognition. Additionally, he introduced the concept of a digital human model.

Zhang *et al.* [163] presented a new member of the biometrics family, namely, tongueprint, which uses particularly interesting properties of the human tongue to base a technology for noninvasive biometric assessment. As they mentioned, tongue is a unique organ which can be stuck out of the mouth for inspection, whose appearance is amenable to examination with the aid of a machine vision system. One may also note that the involuntary squirm of the tongue is not only a convincing proof that the subject is alive, but also a feature for recognition. This implies that the tongue can present both static features and dynamic features for authentication.

In [160], Yasuda *et al.* proposed a visual-based online signature verification system. The input module of the system consists of only low-cost cameras (webcams) and does not need an electronic tablet. They obtained the online signature data from the images captured by the webcams by tracking the pen tip. The pen tip tracking is implemented by the sequential Monte Carlo method. Then, they calculated the distance between the input signature data and reference signature data enrolled in advance is computed. Finally, the input signature is classified as genuine or a forgery by comparing the distance with a threshold. They consider seven camera positions. They performed experiments using a private database consisting of 150 genuine signatures to decide the best camera position. Their experimental results show that they should place the webcam to the side of the hand. Finally, they evaluated the system with a camera placed to the side of the hand against a different database consisting of 390 genuine signatures and 1560 skilled forged signatures. They achieved an equal error rate of 4.1

percent against this database.

Nanni *et al.* [96] presented an on-line signature authentication system based on an ensemble of local, regional and global matchers. The following matching approaches are taken into account: the fusion of two local methods employing dynamic time warping, a hidden Markov model based approach where each signature is described by means of its regional properties and a Linear Programming Descriptor classifier trained by global features. They discussed a template protection scheme employing the BioHashing and the BioConvolving approaches for biometric recognition. Their experimental results which was evaluated on the public MCYT signature database, shown that the best ensemble obtains an impressive equal error rate of 3 percent, when only five genuine signatures are acquired for each user during enrollment. They observed that the equal error rate achieved in the worst case scenario is equal to 4.51 percent.

In [138], Vargas *et al.* explained a method for conducting off-line handwritten signature verification. It works at the global image level and measures the grey level variations in the image using statistical texture features. Here co-occurrence matrix and local binary pattern are analyzed and used as features. This method begins with a proposed background removal. These author's processed an histogram to reduce the influence of different writing ink pens used by signers. Genuine samples and random forgeries have been used to train an SVM model and random and skilled forgeries have been used for testing it. Their validated results indicates reasonable according to the state-of-the-art and approaches that use the same two databases: MCYT-75 and GPDS-100 Corpuses. The combination of the proposed features and those proposed by other authors, based on geometric information, also promises improvements in performance.

Bailador *et al.* [4] identified each user by drawing his/her handwritten signature in the air (in-air signature) to assess the feasibility of an in-air signature as a biometric feature. They have analyzed the performance of several well-known pattern recognition techniques, such as, hidden Markov models, Bayes classifiers and dynamic time warpingto cope with this problem. Each technique has been tested in the identification of the signatures of 96 individuals. Furthermore, the robustness of each robustness of each method against spoofing attacks has also been analyzed using six impostors who attempted to emulate every signature. They achieved the best results in both experiments have been reached by using a technique based on dynamic time warping which carries out the recognition by calculating distances to an average template extracted from several training instances. Finally, they carried out a permanence analysis in order to assess the stability of in-air signature over time.

A method for conducting off-line handwritten signature verification was described by Vargas *et al.* [138]. The proposed method works on global image level and measures the grey level variations in the image using statistical texture features. The co-occurrence matrix and local binary pattern are analyzed and used as features. Initially, their method begins with a proposed background removal. A histogram is also processed to reduce the influence of different writing ink pens used by signers. Genuine samples and random forgeries have been used to train an SVM model and random and skilled forgeries have been used for testing it. Their validated results indicates that they are reasonable according to the state-of-the-art and approaches that use the same two databases: MCYT-75 and GPDS-100 Corpuses.

In [11], Batistav *et al.* proposed a Hybrid generativediscriminative method to design an off-line signature verification system from few samples, where the classifier selection process is performed dynamically. To design the generative stage, they trained multiple discrete left-to-right hidden Markov models (HMMs) using a different number of states and codebook sizes, allowing the system to learn signatures at different levels of perception. To design the discriminative stage, HMM likelihoods are measured for each training signature and assembled into feature vectors that are used to train a diversified pool of two-class classifiers through a specialized Random Subspace Method. This signature verification system was suitable for incremental learning of new signature samples. These authors performed an experimental analysis with real-world signature data (composed of genuine samples and random, simple and skilled forgeries) and indicated that the proposed dynamic selection strategy can significantly reduce the overall error rates with respect to other EoCs formed using well-known dynamic and static selection strategies.

As is mentioned, Biometric systems including keystroke-dynamics based authentication have been studied in the literature. Stefan *et al.* ([128]) the effects of synthetic forgery attacks in the context of biometric authentication systems. Their study is performed on a concrete keystroke-dynamic authentication system. The main focus of their work was to evaluate the security of keystroke-dynamics authentication against synthetic forgery attacks. They performed their analysis in a remote authentication framework called TUBA that they designed and implemented for monitoring a users typing patterns. They modelled the keystroke sequences forged by the two bots using first-order Markov chains. They used support vector machine for classification. Their simulation results showed that keystroke dynamics is robust against the two specific types of synthetic forgery attacks studied, where attacker draws statistical samples from a pool of available keystroke dataset other than the target. They further described TUBAs use for detecting anomalous activities on remote hosts and presented its use in a specific cognition-based anomaly detection system. They concluded that the use of TUBA provides high assurance on the information collected from the hosts and enables remote security diagnosis and monitoring.

A simple and efficient approach to on-line signature verification proposed by Rashidi *et al.* [112] which was based on a discrete cosine transform, which has been applied to 44 time signals, such as position, velocity, pressure and angle of pen. They carried out the experiments on two benchmark databases, SVC2004 and SUSIG. The forward feature selection algorithm is used to search for the best performing feature subsets. Their proposed system was tested with different classifiers, with skilled forgery, and equal error rates. Their results are 3.61 percent, 2.04 percent and 1.49 percent for SVC2004 Task1 and 2, Task2 and SUSIG databases, respectively.

According to Giot and Rosenberger [46], generally the biometric system provides a good performance but for some individuals it is exceptional as its output is based upon the quality of capture. To solve some of these problems, they explored the use of multi-biometrics. In this approach, they combined different biometric applications, for example, multiple captures of the same biometric modality, multiple feature extraction algorithms, multiple biometric modalities, etc. These authors were interested in score level fusion function's application, i.e., they used a multi-biometric authentication scheme which accept or deny the claimant for using an application. They validated the proposed method on three significant biometric benchmark datasets.

Jin *et al.* [61], used a set of minutiae points via a polar grid based 3-tuple quantization technique. They outlined some merits of their method and used four publicly available benchmark datasets: FVC2002 DB1, DB2 and FVC2004 DB1, DB2 are used to judge the accomplishment of the this method. They also analyzed the diversity, revocability and non-invertibility criteria.

In [148], Xianye *et al.* proposed biometrics technique based on metric learning approach. They used this approach to achieve higher correct classification rates under the condition that the feature of the query is very different from that of the register for a given individual. Stimulated by the definition of generalized distance, they defined the criterion of this new metric learning by finding an embedding that preserves local information and obtained a subspace that best detects the essential manifold structure. Using a generalized eigen-decomposition, they obtained the two transformation matrices for the query and the register. They tested their experiments on biometric applications of gait and face databases and realized that their method performs better than classical metric learning methods as well as the radial basis function algorithms for such applications.

Wang and Liew [139] mentioned that unlike some other traditional biometric features such as face, fingerprint, or handwriting; lip biometric features contain both physiological and behavioral information. On one hand, physiologically, different people have different lips. On the other hand, people can usually be differentiated by their talking style. As they have correctly mentioned, current research on lip biometrics generally does not distinguish between the two kinds of information during feature extraction and classification. Till this work, the issue of whether the physiological or the behavioral lip features are more discriminative, was not addressed. In this work, these two authors studied different physiological and behavioral lip features with respect to their discriminative power in speaker identification and verification. Their experimental results showed that both the static lip texture feature and the dynamic shape deformation feature can achieve high identification accuracy (above 90%) and low verification error rate (below 5%).

In [147], Xianye *et al.* presented a kernel coupled distance metric learning (KCDML) method to study the biometrics which may have adverse impact by different walking states, walking directions, resolutions of gait sequence images, pose variation and low resolution of face images. By using a kernel trick and a specialized locality preserving criterion, they formulated the problem of KCDML as an optimization problem whose aims are to search for the pair-wise samples staying as close as possible and to preserve the local structure intrinsic data geometry. They mentioned that instead of an iterative solution, one single generalized eigen-decomposition can be leveraged to compute the two transformation matrices for two classifications of data sets. They empirically demonstrated the effectiveness of the proposed method on gait and face recognition tasks, results of which outperform four linear subspace solutions and four nonlinear subspace solutions.

Islam *et al.* [59] presented automatic extraction of local 3D features (L3DF) from ear and face biometrics and their combination at the feature and score levels for robust identification. To the best of their knowledge, this work is the first to present feature level fusion of 3D features extracted from ear and frontal face data. They used a weight sum rule to get the scores from L3DF based matching. They also achieved identification and verification (at 0.001 FAR) rates of 99.0 percent and 99.4 percent, respectively, with neutral and non-neutral facial expressions on the largest public databases of 3D ear and face.

Chakraborty *et al.* [23] mentioned that against the backdrop of growing concerns about security, face-based biometrics has emerged as a methodology to reliably infer human identity. They highlighted that active learning algorithms automatically select appropriate data samples to train a classifier and reduce human effort in annotating data instances. They applied a novel optimization based batch mode active learning strategy to a face recognition problem. They tested their results on the VidTIMIT and the NIST MBGC datasets and certified that the potential of their method in being used for real world biometric applications.

Lujan *et al.* [86] analyzed the performance of several well-known pattern recognition and dimensionality reduction techniques when applied to mass-spectrometry data for odor biometric identification. Motivated by their previous works on capturing the odor from other parts of the body, in this work, they attempted to evaluate the feasibility of identifying people by the odor emanated from the hands. By formulating this task according to a machine learning scheme, they identified this problem with a small-sample-size supervised classification problem in which the input data is formed by mass spectrograms from the hand odor of 13 subjects captured in different sessions.

In [126], Smeets *et al.* described a meshSIFT algorithm and the benefits in 3D face recognition. This algorithm consisted of four major components. Firstly, in the scale space salient points on the 3D facial surface were detected by the means of curvature extrema. Secondly, adjustment were done to each of these salient points. Thirdly, in a feature vector consisting of concatenated histograms of shape indices and slant angles, the neighbourhood of each salient point was described. Finally, feature aim of the two 3D facial surfaces were matched by comparing the angles in feature space.

Imamverdiyev *et al.* [58], considered the texture descriptors, namely, the Gabor filter-based FingerCode, a local binary pattern, a local direction pattern and their various combinations. They binarized these fingerprint texture descriptors using a biometric discretization method and used it in a fuzzy commitment scheme. They built the biometric cryptosystems by combining discretized fingerprint texture descriptors and using effective error-correcting codes. They tested the proposed system on a FVC2000 DB2a fingerprint databases.

The characterization of the handwriting process involving a vectorial summation of lognormal functions: the Sigmalognormal model proposed by Plamondon *et al.* [106].

These authors described a new stroke extraction algorithm suitable for the reverse engineering of handwriting signals. It is shown how the resulting representation can be used to study the writer and signer variability. Human movement modeling can be of great interest for the design of pattern recognition systems relying on the understanding of the fine motor control (such as on-line handwriting recognition or signature verification) as well as for the development of intelligent systems involving in a way or another the processing of human movements. They reported on two joint projects dealing with the automatic generation of synthetic specimens for the creation of large databases. Their first application was concerned with the automatic generation of totally synthetic signature specimens for the training and evaluation of verification performances of automatic signature recognition systems. The second application has dealt with the synthesis of handwritten gestures for speeding up the learning process in customizable on-line recognition systems to be integrated in electronic pen pads.

The detection of alcohol intoxication on the basis of handwritten signatures was investigated by Shin and Okuyama [121]. They found that the signature attestation rate varies at an individual level according to sex, age, acetaldehyde removal efficiency, and individual constitution from the previous study. For this study, they employed 30 people to evaluate the change in a handwritten signature before and after alcoholic intake. Firstly, they measured the signature verification rate using the online signature verification system. The signature verification rate measured using the WACOM Tablet pen before alcohol consumption was 97.0 %. They detected the level of alcohol intoxication on the basis of the total time taken for writing the signature, the average pressure of the brush, the two-dimensional writing speed, and internal angle of stroke turns. Their results indicated that the maximum alcohol detection rate of this method was 95.1 % achieved when they examinees are tested 35 min after alcohol consumption. Finally, they observed that the rate of alcohol detection increases with the alcohol density of an examinees breath.

In order to provide some insight on their actual discriminative power for online signature verification, recently, Parodi and Gomez [102] analyzed feature combinations associated with the most commonly used time functions related to the signing process. They defined a consistency factor to quantify the discriminative power of these different feature combinations. Then they proposed a fixed-length representation of the time functions associated with the signatures, based on Legendre polynomials series expansions. The expansion coefficients in these series are used as features to model the signatures. They considered two different signature styles, namely, Western and Chinese, from a publicly available signature database to evaluate the performance of the verification system. They used two state-of-the-art classifiers, namely, support vector machines and random forests in the verification experiments. They presented error rates which were comparable to the ones reported over the same signature datasets in a recent signature verification competition in the then literature. The experimental results indicates that there is a good correlation between the consistency factor and the verification errors.

Some other works revelent to this research are [2, 18, 27, 29, 36, 37, 38, 42, 44, 48, 52, 57, 60, 64, 70, 71, 81, 84, 86, 97, 99, 107, 108, 109, 115, 116, 117, 124, 129, 131, 139, 146, 152, 154, 157]. On the other hand, some more relevant research contributions are reviewed in the individual chapters.

1.4 Outline of the thesis

The rest of this thesis is organized as follows. In Chapter 2, we discuss about the biometrics and their applications. Chapter 3 deals with the preprocessing of the signature features and information about the features extraction. In Chapter 4, we explore some methods for signature modelling. Signature modelling using HMMs is discussed in Chapter 5. In Chapter 6, we explain the main implementation and results of the signature verification system. Finally, in Chapter 7, we present some concluding remarks and also indicate scope for future research.

Chapter 2

Biometrics-Signature system

Biometrics is primarily concerned with identifying the unique physical characteristics or behavior of an individual in order to grant or deny them access to some type of a system. In this chapter, we are going to discuss in detail about the biometrics and their identification systems such have fingerprint, face recognition etc. We then defined about the automatic signature verification and signature acquisition.

2.1 Introduction

Modern day security procedures have gained more important offlate. This can be mainly attributed to their reliability and accuracy in obtaining desired results. Biometrics is one of those modern day security procedures in which isolation and measurement of biological or behavioral characteristics play a vital role as the claimed identity of a person can be authenticated by measuring his/her unique biological feature and matching it to a known authentic sample. The biological characteristics isolation and measurement play an important role in modern day security procedures as the claimed identity of a person can be authenticated by measuring a unique biological feature of that individual and matching it to a known authentic sample. Moreover, identification of a person can be performed by matching these measurements to an entire database of a known population.

Biometrics are preferred over more traditional PIN (personal identification number) oriented means of authentication for a number of reasons. Biometrics requires a person to be physically present at the point of verification whereas a PIN can be entrusted to other persons; thus, a positive identification is not beyond doubt. Whereas in the past, biometric authentication has been carried out by human forensic experts, advances in computing technologies in the last two decades have made the automation of the process possible; they allow deployment of biometric authentication systems in commercial environments apart from their more traditional use in the criminal justice system.

In general terms, all biometric identification systems work in the same way. A user must be enrolled into the system by taking measurements of the specific biological characteristics. The digital representation are created by these measurements which is then stored in a database together with supplementary information about the individual such as a PIN. Whenever a user needs to be identified, e.g., when entering a sensitive area or conducting a financial transaction, the scan is repeated and the PIN is entered into the system at a verification terminal. This code is then compared to the code in the database by some algorithm to decide on the authenticity of the claimed identity.

By the invention of the SMARTCARD it was made possible to store the code on a card which is carried by the user and presented whenever personal identification is required. The code generated by the second scan is compared to the encrypted code on the card which obviates the need for a central user database.

The efficiency of biometric identification schemes are generally judged by three criteria:

1. The false-acceptance rate (FAR) which is the percentage of authentication attempts deemed to be true but are in fact false,

2. The false-rejection rate (FRR) which is the percentage of authentication attempts deemed to be false but are in fact true, and

3. The required processing time of authentication.

Biometric identification schemes have to deal with a trade-of between the FAR and FRR as they often antagonise each other; in an attempt to lower a system's FRR, the allowed variance has to be increased which naturally leads to a higher FAR. The optimal performance point in a system is achieved where the FAR and FRRs intersect; this point is referred to as the equal error rate (EER). The aim of biometric authentication schemes is to achieve the lowest possible EER which often necessitate developing complex algorithms to process and compare the measured biological features. Developing such algorithms is the topic of many research efforts in the field of biometrics. Person identification can be measured by different biological features which are facilitated now. Each of these have advantages and limitations in measuring the personal identification and the choice of identifying features depends largely on the context they will be used in. The general considerations when choosing a biometric plot for a certain application include

1. The level of reliability needed,
2. The development and deployment costs,
3. The target population demographics,
4. The target operating environment,
5. The speed of operation, and
6. The susceptibility to forgery.

The rest of the chapter is organized as follows. We elaborate a bit more on biometric identification system in Section 2.2. A detailed account of work on automatic signature verification is discussed in Section 2.3. Then we study some signature databases in Section 2.4. Finally, we give a brief summary in Section 2.5.

2.2 Biometrics identification systems

Now, we will discuss some common biometrics for identification in use today.

Fingerprints

Fingerprint scanning is probably the most common biometric in use today. The low cost and fairly high recognition performance make this an attractive solution for many person identification applications. The scanning of a fingerprint is performed by detecting heat variations on the finger surface; a sensor builds a map of an individual's finger. There are unique maps to each individual which makes it suitable to identification. Other approaches are based on optical imaging or measurement of small electrical variations across the finger surface. The optical imaging approach is most sensitive to forgery as it is the easiest to duplicate from an authentic sample.

The large scale practical implementation of a fingerprint based system in the payout of pension funds in South Africa has revealed some problems with fingerprint recognition. Some individuals poses fingerprint patterns which are inherently difficult to verify by current available algorithms. For individuals depending largely on their hands to perform their work, recognition performance can be impaired by scars. This suggests that fingerprint recognition is better suited for environments where fingers are less prone to damage. We are presenting the reviews of most recent works in the field of fingerprint system.

Voice recognition

Voiceprint identification relies on the unique characteristics of the vocal tract of a person which results in a distinct voice character for individuals. Humans can very often recognize a person over telephone only by hearing the person speak which reinforces this claim. Voice printing has become an attractive solution to the endorsement of telephonic banking transactions. The recognition algorithms are challenged though by variance induced in the speaker's voice due to illness, high noise ratios on telephone lines and the acoustics at the point of recording.

Iris scanning

The iris is the coloured ring of tissue surrounding the pupil of the eye. It consists of a unique pattern of features such as striations and freckles; they remain unchanged over a lifetime of an individual and are thought to be impossible to forge. This makes iris scanning a highly effective personal identifier. However, it is not well accepted by users due to the sensitive nature of the eye. Iris scanning is at present used mainly to restrict access to high-tech and high risk security environments. The second phase of the Noisy Iris Challenge Evaluation attracted participation by 67 research groups from around the world. In contrast to all current commercial Iris biometrics technology, the NICE competitions focus on performing Iris biometrics on visible-light images. Whereas NICE.I focused on segmentation, NICE.II focused on performance in feature extraction and matching. The eight top-performing algorithms from NICE.II are considered and suggestions are made for lessons that can be drawn from the results.

Retinal scanning

The retina is the light sensitive layer at the back of the eye which triggers nerve impulses via the optic nerve to the brain. With retinal scanning, the unique patterns on the retina are scanned by a low intensity light source via an optical coupler. It has proven to be quite accurate but does require the user to look into a receptacle and focus on a given point. This is inconvenient if the person wears glasses or has concerns about intimate contact with the reading device. For these reasons, retinal scanning has low user acceptance although the technology itself can work well. In practice, retinal scanning is used marginally compared to iris scanning.

DNA prints

DNA (deoxyribonucleic acid) is the hereditary material found in all body cells. It contains subunits called bases which vary significantly across the population and apart from identical twins, the overall pattern of these sequences is unique for each person. A single cell from a biological sample, e.g., blood, saliva, semen or hair, is sufficient for laboratory analysis to extract a DNA print. However, it is unlikely that it will be used in the near future as a commercial identification scheme due to the complexity of extracting the DNA print by current methods. Its use is restricted to forensics to link suspects to biological trace evidence found at crime scenes. As a biometric, it is so reliable that courts accept it as irrefutable proof of guilt or innocence of suspects.

Dental records

Dental records of a person can sometimes serve as a valuable identification characteristic. Due to their nature, teeth are less subject to decay than other biological features. In cases where fire has destroyed other biological features beyond recognition, the unique arrangement of an individual's teeth can be used as a last resort to identify the unknown person. Bite marks on victims of criminal abuse can also provide a useful clue to the identity of an assailant.

Hand geometry

Hand recognition systems require users to place a hand palm down into a reader. An infrared source within the reader projects an image of the hand as a silhouette; it is captured by a high-resolution digital camera. The reader computes the widths and lengths of fingers and makes up to 90 other measurements from the captured sillhouettes. Hand geometry based systems offers a good balance of performance and ease of use. This methodology may be suitable for large user databases or users who may access the system infrequently and may therefore be less disciplined in their approach to the system. Although it is one of the earliest developed biometric systems, it remains a popular identification solution.

Face recognition

Face recognition inspects a digital snapshot of a person's face in an attempt to verify the identity of the person. The position and size of the eyes, nose and mouth and the overall shape of the face contribute to the decision process. The face of the average person undergoes changes over time due to changing hairdos, weight facial hair growth or removal, and glasses. This variability poses a challenge to face recognition systems. Such systems are currently becoming ubiquitous at large public venues such as sport stadiums to assist authorities to detect the presence of assailants in the crowds and as such have raised some public concern about invasion of privacy.

2.3 Automatic signature verification

Automatic signature verification (ASV) has been a research topic for quite some time. The first active research period appears to date back to the mid-seventies [26, 34, 51, 55, 127, 159] where the majority of the efforts went into developing special hardware to capture the signing process. The aim of this survey is to provide an annotated summary of the different modelling methods published mainly during the period 1989-2013. We hope it will provide the interested reader with insight into the different aspects involved in developing a signature verification system and serve as an overview of the avenues already pursued within the field. Only dynamic signature verification (as opposed to static signature verification) is considered. For other related works, see [49, 72]. From a global point of view, the main issues involved in developing an ASV system are

1. The choice of device to acquire signatures,

2. The choice of computing hardware to perform the various tasks involved,

3. The algorithms used to achieve the desired effect,

4. The enrollment and maintenance procedure for signatures,

5. The configuration of the test database for R&D purposes which apart from the enrolled authentic signatures also contains forgeries,

6. The configuration of the database for a production version of the system, and

7. The possible need for networking if the system is to be deployed in a distributed scenario.

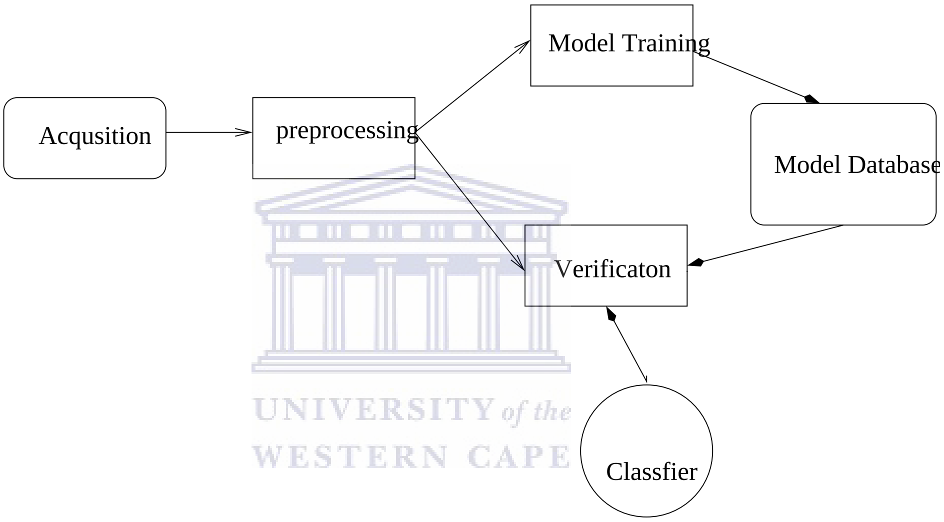

Figure 2.3.1: Automatic signature verification system

Most ASV systems adhere to the abstraction depicted in this figure. The functioning of most ASV systems adheres to the abstraction depicted in Figure 2.3.1. Users enroll in the system by providing a set of their signatures. Signatures are then preprocessed to make them invariant to transformations and to convert them into a format suitable for the modelling process. These signatures are then submitted to a modeller which extracts a number of values from this training set which serve as the parameters defining the modelling approach's view of the set. These values are stored in a database along with the necessary details of the user. When the system is presented with a suspect signature claiming to have originated from some user known to the system, the user's model parameters are retrieved from the database and used to decide on the authenticity of the signature according to the semantics of the model. A number of modelling approaches have been applied with varying degrees of success which are covered in chapter 4. Performance results reported by the earlier studies will not be mentioned. the simple reason being that the results can be particularly misleading in the field of ASV due to the lack of a standard test bench and the disparate conditions under which the results are produced by Nalwa [95].

As Parizeau and Plamondon [100] stated, differences in the quality and types of forgeries are enough to render any comparison meaningless, as are the differences in the sizes of the training and test subsets, the number of trials permitted and the type of classifier used. Any attempt to compare results of different schemes becomes meaningless unless the results are based on the same data set and the same training/testing partitioning [98]. The credibility of results depends largely on the test database from which the results are deduced.

2.4 Signature database and acquisition

A benchmark signature database plays a very important part during development of an ASV system. With a well planned database, an algorithm's performance can be gauged and the effect of changes to the algorithm monitored with confidence that one is actually gaining ground. A good database represents a possible real-life deployment scenario as closely as possible. This means that several factors need to be taken into account when creating the database. Various discrepancies between test setups and real world scenarios are revealed in [95] where the author argues that reported results are often an over-optimistic re of the performance of the systems were they to be implemented in practice. Ideally, a system is tested with a large nonhomogeneous population over a long period of time [105].

On the other hand, very often databases are reported to have been collected on campuses or in the offices of technical institutions. This contradicts the statistical principle of a population representative sample. Unfortunately, creating such a database is a resource intensive task and to our knowledge there does not exist any database suitable for benchmark purposes which have been donated to the research community. Factors that need to be taken into account when drawing up a list of users to be enrolled in an experiment are gender, age and dexterity. Various factors regarding the signing environment should also be considered [30, 140].

Users might need to be given time to familiarize themselves with the writing device as it might not have the same feel as an ordinary pen. Crane and Ostrem [25] described in a fair amount of detail a data collection procedure for creating a signature database. The procedure requires half of the signature set donated by a user to be created in the standing position to ascertain whether there is any significant difference in the two groups of signatures for an individual. For this study, signatures were collected over a four-month period with one or two data-collection sessions per week. Finding volunteers willing to commit to such a lengthy experiment may be difficult. Crane and Ostrem [25] noted that due to the lack of motivation to produce signatures as consistent as possible during an experimental session, signatures might not be of the quality which could be expected in a scenario where the user incurs some penalty for failing to produce an acceptable signature such as being denied access to a secure area.

For this reason, Crane and Ostrem [25] and others offer cash incentives to users to improve the quality of both authentic signatures and forgeries. Some studies collect all signatures in a single session but it is arguably a more realistic approach to gather the signatures of a user over a longer period of time. This is both to prevent boredom and muscle fatigue and to capture natural variations due to physical and psychological changes which is more likely to surface over a longer period of time. Nalwa [95], collected the signatures in two sessions at least a week apart. In [100] Parizeau and Plamondon collected ten signatures from the users in each of five sessions during one week. In [103], Paulik *et al.* were collected signatures over ten sessions with ten signatures per session.

Matsuura and Sakai [89] were collected signatures over a six-month period in which, if time permits, would be handy to determine a more reliable measurement of the true performance of a system in a practical setting. This is because a practical scenario typically requires users to provide a signature set in a single session to minimize inconvenience. Even though there is to our knowledge no study which investigates the variations of signatures over an extended period of time, one can expect a statistically significant change in the signatures of at least a small percentage of users [140, 145].

This means that the actual performance of a system might deteriorate over time because models are built from a set donated in a relatively short time; they do not capturing the variances a user's signature might undergo over time. For this reason then, production quality systems also incorporate adaptive measures for model parameters from authenticated signatures. The assumption being here that users will access a system often enough that authentic signatures will not change so drastically between sessions that they will be rejected. The total number of signatures needed by a model to deduce its parameters, varies among different modelling approaches.

Some studies, Bromley *et al.* [20] perform cleanup of the database to rid it from noisy signatures. The criteria used to prune a database include legibility, signing duration within a tolerable distance from the average duration and sabotage such as volunteers signing as Mickey Mouse. Nalwa [95] mentioned the term goat in ASV literature refers to a user whose signature has a large negative impact on the overall performance figures of a system. Pruning often seeks to remove such signers. This can result in a false interpretation of performance statistics. Instead, we believe that it is useful when results highlight the number of goats as perceived by the particular modelling approach and provide figures with and without the goats.

For research purposes, forgeries are very important to measure the performance of a modelling approach. The mere fact that a system accepts authentic signatures is by no means a guaranty that it will reject forgeries. Therefore, the quality of forgeries in a database will to a large extent determine the credibility of results derived from the database. The first kind of forgery often encountered in the literature is the zero-effort forgery also known as a random forgery. This term refers to a signature taken from one enrolled user and presented as the signature of another user. At the very least, a system must be able to reject such 'forgeries' with a great amount of confidence.

In Munich and Perona [93], the authors state that a system which performs well on random forgeries are likely to perform well on actual forgeries which is a statement open to debate [30]. It does serve a purpose though as Nalwa [95] pointed out that credit cards can be stolen while in transit before it is signed by the owner. The forger will in such a case have no idea what the signature looks like. Parizeau and Plamondon [100] used only random forgeries for this study which does not attempt to maximize performance but rather serve as a comparison between different modelling approaches. Random forgeries are adequate in such cases as only relative performance is of importance. In some studies, Nelson [98] showed a static image of a signature to forgers and allow them to practice the signature before producing the actual forgery for the database. Crane and Ostrem [25] were collected forgeries by selecting motivated individuals with good manual dexterity and the capability of understanding the basic principals of the system.

It was explained that the system inspects dynamic information as well as final appearance. Cash prizes were awarded to the creators of the best forgeries in an attempt to motivate forgers to help create a quality database. The first set of forgeries was created after static images of the signatures to be forged were shown to the forgers. After this, video recordings of the actual signing process were shown to the forgers. They were given three weeks to practice as much as they wanted before submitting the second set of forgeries. As there is no a priori knowledge of the forger population expected to attack a dynamic signature verification, this approach seems to be a step in the right direction.

In [105], Plamondon modulated sound recording of the signing process was provided to forgers together with the trajectory information of the signature to be forged. Forgers were given time to practice the signature while listening to the recording after which a set of forgeries is recorded. Again, cash incentives were offered to the creators of the best forgeries. Dolfing *et al.* [30] distinguished between three types of forgeries

1. Home improved forgeries,
2. Professional forgeries, and
3. Over-the-shoulder forgeries.

The home improved variant is created after the forger had only access to a paper copy of the signature. Over-the-shoulder forgeries are, as the name suggests, created after the forger could see the entire signing process of the signature to be forged by standing behind the forger. For the professional forgeries, forensic document examiners provided forgeries based on paper copies of the forged signatures. From the above, one can understand why it is difficult to compare performance results obtained from different databases. There are simply too many factors which can in the results; publicly available benchmark databases such as found in the field of speech recognition would be a great advantage to the field of ASV. There is unfortunately some legal aspects involved in releasing the signatures of volunteers for public scrutiny.

Data acquisition

With static signature verification, only static images of signatures are available. This implies that no clue as to the order of signature rendering can be non-trivially deduced from the data. With dynamic signature verification, one or more aspects of the signing process are sampled from a time varying signal. This means that the captured signature can be seen as a time series and well founded modelling techniques can be employed. The acquisition process is very important because the quality of the signals is critical to optimizing the comparison process. Following is a summary of signature acquisition methods reported.

Digitizers

Digitizers (or tablets as they are also known) are at present the most commonly used devices for dynamic signature acquisition [72]. They are often used in Computer Aided

Design applications and boast a high spatial resolution for capturing pen movements. In Dolfing *et al.* [30] a Phillips proprietary digitizer called Phillips Advanced Interactive Display (PAID) is used. This device consists of an LCD and orthogonal sensors for pen and finger input sampling. With a sampling rate of 200Hz, the device provides a tuple of $(x, y,\ pressure)$ and pen-tilt information with each sample. It should be noted that only the most expensive tablets possess a LCD display on the tablet surface. It is more common for the tablet to use the display of the workstation it is attached to. Other studies employ tablets with differing functionality. In [53, 91] digitizers without the ability to sense pen tilt information are used. There is much variability in the resolution of tablets which may range from 100 to 1000 dpi. Some tablets allow users to use their own pen.

As Herbst and Richards [54] explained, the problem with these are that fingers can protrude into the pressure sensitive area and be registered as part of the signature. Where a special pen has to be used, Herbst and Richards [54] mentioned that the pen might not have the same natural feel as normal pens but the quality of the acquired signatures is much better. For a tablet to report pen tilt, a special instrumented pen has to be used. In the future, tablets could be used as the man-machine interface for tele-banking systems enabling ASV as the preferred method of transaction authentication [156]. Bromley *et al.* [20] has developed the NCR such a signature capturing device for the banking industry. According to Nalwa [95] some tablets have extended functionality normally performed by software such as signal smoothing and compression. In [103], Paulik *et al.* described that the advent of Personal Digital Assistants employing miniature digitizers as the man-machine interface, has spurred renewed interest in the field of handwriting recognition and ASV. To read more about graphical tablets, see, WACOM graphical tablets.

Instrumented Pens

A problem with using tablets is their size and cost which seriously hampers their chances of ever finding their way into mainstream ASV applications. Development of specially instrumented pens is an attempt to overcome these problems. A microprocessor-based interface control card is presented by Mital and Lau [92]. A piezoelectric transducer pen is used to convert the signature pressure to an electrical signal before being amplified by a charge amplifier. The output of the charge amplifier is then fed to the interface control card to be digitized.

Baron and Plamondon [5] presented a system which employees an instrumented pen with the ability to sense gravitational acceleration. The pen also incorporates a pressure transducer which delivers an electrical signal proportional to the force exerted between the pen and paper. Various problems with accelerometer-based systems and possible solutions are highlighted by Baron and Plamondon [5]. Special pens are less commonly used than tablets for acquisition. However, the SMARTPEN biometric authentication system might change it. This device has only recently been introduced to the market but is the first serious device dedicated to ASV in a commercial environment. The state of the art technology employees an off-the-shelve ballpoint tip. Sensors producing uncorrelated measurements of forces in three directions exerted on the pen tip is located just behind the tip. It also contains sensors to detect the angles the pen makes with the horizontal plane. On pen circuitry takes care of data sampling and conditioning. A radio frequency transmitter conveys the signals in secure encrypted form to a base station. This solves the problem of its predecessors which had to be connected to the station by cable. The pen is driven by standard off-the-shelve batteries.

Cameras

Munich and Perona [93] proposed a new approach to signature acquisition is through the use of a camera. It is argued that cameras are becoming ubiquitous in computing environments and are smaller and easier to handle than digitizers. The tracking of the pen tip during signing is, however, a difficult process and appears to be not as reliable as one would have hoped for. The system sometimes looses track of the pen tip when the signer signs fast. In such a case, the system requires the user to adapt his/her signature to the system. This constraint could result in some difficult in a commercial environment.

2.5 Summary

It is clear from the discussion in this chapter that the signature verification research has found its way into a number of commercial applications. A simple web search reveals various commercial ventures which utilize automatic signature verification. Applications range from financial transaction authentication to restricted area access control. In next chapter, we discuss about the signature preprocessing, feature extraction and their importance in the real world applications.

Chapter 3

Feature extraction for signature verification system

Feature extraction is very important topic in the field of pattern recognition. In this chapter, we will present an overview of different signature preprocessing techniques and various types of features. We then discuss about the feature selection methods.

3.1 Introduction

Preprocessing is an attempt to convert a raw sampled signature to some canonical form by carrying out various operations on the data. Munich and Perona [93] assumed that users are consistent in their style of signing and therefore no normalization is performed: They write their signatures with a similar slant, in a similar amount of time, with similar dimensions and with similar motion. Experience has shown that this is be an optimistic assumption. A considerable amount of effort was spent on normalization of signature data in several studies. This section focuses on some aspects tended to by the various studies examined.

To compensate for differences in the resolution of tablets, [143] linearly normalize x and y coordinates to reside within a known interval. To compensate for the differences in sampling rates of tablets, Wu *et al.* [143] used interpolation to resample signals into a fixed number of points. Various studies [53, 54, 66] reported on using cubic smoothing B-splines for interpolation.

Different samples of a writer's signature might be created on differing baselines if the acquisition phase does not restrict the signing action to a uniform orientation. It would be advantageous if such a restriction could be lifted as there is no guaranty that signatures will not deviate from a given baseline for some writers even if provided. Lee ([74]) disagreed with this by stating that there is no need for rotational normalization if a baseline is provided. Furthermore, one cannot assume that users will sign their signature the same size every time. This imposes the need for an operation which makes signatures scaling invariant. Paulik *et al.* [103] stated that rotational differences can serve as a distinguishing feature. Various techniques are employed in the literature to make signatures rotational and scaling invariant which are summarized here.

In [145], Wu *et al.* a signature is normalized by finding a smallest enclosing circle for it. The center of this circle is selected as a reference point to convert the signature into polar coordinate form, i.e., (r_t, θ_t). Once in this form, the r_t component is normalized with respect to the radius. The θ_t component is normalized by subtracting the value of the previously sampled point, i.e., $\theta_t - 1$. Rotational and scaling invariance was achieved by creating a sequence of components.

Yang *et al.* [156] rotational invariance is achieved by regarding a signature as a sequence of vectors in the two dimensional Cartesian plane. Each vector is normalized by subtracting the angle the very first vector makes with a principal axis. A potential problem with this approach is the dependence on a single vector for normalization. This might degrade performance if a signature is unstable in the starting sequence of a signature.

In Kashi *et al.* [65, 66] transform a signature into canonical form in the frequency domain. The first derivative of the sampled coordinates are obtained to reduce end-point distortions. This derived signal is converted to the frequency domain by applying the Fourier transform. Transformations are carried out in the frequency domain to achieve rotational and scaling invariance. It is done this way as the intended operations are conceptually much simpler in this domain. Smoothing is achieved by zeroing small amplitude frequencies. After the transformations, the signal is converted back into the time domain for modelling.

The neural approaches used in [20, 74] call for a fixed sequence length. This is achieved by linear-time normalization of a signature's spatial time function $(x(t), y(t))$. The data is resampled with respect to the time parameter. This might inherently distort the input signature [143] especially if the resampled sequence length is shorter than the original. They exclude important information carried in frequency bands excluded by the resampling. Other modelling methods such as dynamic time warping do not require sequences of a fixed length. For these methods however, computed distance values between two sequences are often later subjected to length normalization.

Effective preprocessing is unavoidable if a verification system is to attain commercially acceptable performance and work for a wide variety of hardware. For credit card transactions, Nalwa [95] regards a 1% false acceptance rate and a 7% false rejection rate as reasonable. Given that there is currently hardly any verification done and the potential user resistance to having one out of every 14 signatures rejected, we would rather see these numbers reversed. LeClerc and Plamondon [72] required a base-line performance of 0.05% FRR and 20% FAR for inclusion of the results in their survey. In practice though, the required error rates depend largely on the penalties incurred
5 pen tilt is a measurement of the angle the pen makes with the surface of the tablet
6 commonly known as the Nyquist frequency by making an error of each of the two types in the specific scenario [105]. Reference Paulik *et al.* [103] underlines the need for preprocessing. They perform very little preprocessing and conclude that in order for their modelling approach to obtain acceptable error rates, more attention needs to be paid to preprocessing. If used in a sensible fashion, the information removed during a normalization phase can, when isolated, be used to improve a system's performance [74]. Furthermore, depending on the modelling approach, normalization might not be necessary as far as rotation is concerned. In such cases, only rotational invariant features are used to represent a signature (see [74]). The absolute velocity of the pen tip

is one such feature.

It is true that large number of features does not always give the better performance and may create some difficulties. For instance, if a method uses many features, the storage needs to store the values of those features for the reference signature is going to be relatively large and device like credit card may lack sufficient capacity to store all the values. When a reference signature is compared to the, given that no two genuine signatures are identical, mostly a genuine test signature may not match with test signature which have all features values close to the values for the reference signature. To make sure that genuine test signatures are authenticated, a technique using a large number of features either must have a large threshold for the norm of the distance or use some criterion similar to the majority classifier used by Lee [75]. The major part of classifier is not particularly satisfactory since it cannot be easily analyzed theoretically. Although a large number of features are considered important, however several of them are ignored while comparing the test signature to the reference signature. Hence this arises many arguments.

There are many investigations about the global features and which are considered. Crane and Ostrem [25] studied large a number of features they used an instrumented pen to sample three forces of the writing tip (viz. downward force and the x and y forces) and then initially computed 44 global features. In this method they used to remove one feature in turn from the current set of features (initially 44) and found the feature whose removal gave the lowest EER. The method continues until removing a feature does not reduce EER. In that experiment, the selected 25 features. To evaluate the proposed technique, a database of 5220 genuine signatures from 58 subjects was collected over a four-month period and 648 skilled forgeries from 12 forgers that were allowed to practise the signatures to be forged. EER as low as 1.5% was obtained although about half the genuine signatures were used for selecting the subset of features while the other half for testing.

Lee *et al.* ([75, 76]) described a set of 42 features. One of the techniques that uses only genuine signatures, for each feature, the mean for subject is compared with the means of the same feature for all other subjects and the maximum of such distance is computed for each subject for each feature. The priority of each feature for an individual is then given by this maximum distance. Than this algorithm is used to find several subsets for each user; 34 features performed better than 42 features. After the forgery data was available, distance was computed between the i^{th} feature of the subject and the matching feature of the forgeries for that individual and features with the largest distances were selected. By this it was shown that 23 or 24 features gave the best performance.

Ketabdar *et al.* [67] present a methodology for selecting the most particular global features. More than 150 global features from 60 papers were considered and an initial subset of 46, was investigated. In this approach they used, a near optimal feature space search algorithm that avoids the exhaustive search was used. A cost function based on within-class variability being small and between-class variability being large was used. In order to take into account effects of correlation between feature vector components, the cost is computed on the whole feature vectors instead of individual features. The results of applying the method to a 25-users subset of the MYCT database resulted in 12 features.

Fierrez *et al.* [40] evaluated 100 global features to find their particular power. It was carried by computing the Mahalanobis distance between the mean of the training signatures of a person and the set of all training signatures from all users. Ranking for the features were given according to inter-user class separability many others researchers have studied global features. For example, [114] *et al.* studied 46 global features and used MYCT database to select 12 best features including AV, AP, TT, pen down samples, DPVX, average and maximum pressure.

There were many transformational techniques proposed, the simplest only suggest smoothing the data. But their was no major contribution shown by such transformations. Phelps [104] used a space-domain approach in which a close-fitting polygon was formed around the signature image. The signature area was normalized and centered on a coordinate plane. Phelps [104] showed that the area of overlap for a pair of valid signatures was consistently higher than for forgeries. [65], *et al.* developed cleaning process that included removal of irregular and excessive points, in that cusps were detected and marked, cubic B-spline smoothing was applied and the signature was then re-eximed at intervals of equal arc length. It was considered earlier that at least five signatures are needed for acceptable performance of a parametric technique. Reference signature is based on a set of sample signatures and for each element of the set of selected features the mean and standard deviation (SD) of the feature values has to be estimated. To get the approximate calculation of the mean and standard deviation's features, values for the genuine signatures population it needs several sample signatures. Experimentation by Fierrez *et al.* [39], Gupta and Joyce [50] and Nalwa [95] and showed that the performance improves with the number of sample signatures used five or six sample signatures lead to acceptable performance.

In Parks *et al.*, [101] suggest that at least six sample signatures should be used. They mentioned that if the six sample signatures are gathered under identical conditions, the standard deviation's of the features might be too small to be an accurate estimate of the standard deviation's of the person's signatures and a method for either increasing the standard deviation's or in some cases remove the previously obtained sample signatures was suggested. In some instances the a new set of sample signatures were acquired.

Reference signature should be updated regularly, as the user's signature progress over time. Crane and Ostrem [25] modified the reference signature when a signature was successfully verified by adding the new signature vector to the mean reference vector with a weight of 1/8. This approach does not make much difference on users whose signatures do not change over time and should have a positive effect on users whose signatures changes. Parks *et al.* [101] also suggest that the reference signature should be updated every time a signature is verified by applying a weighting of 10% to the new verified signature. Many researchers, like Fairhurst and Brittan [35], supported the fact of use of individual sets of features and individual thresholds.

There are difficulties in finding a set of individual features and individual thresholds without having access to a large number of training signatures of each individual. They proposed that if five sample signatures can be used and based on the mean and SD of each feature's values make a decision about which features and what threshold could be used for each individual. Crane and Ostrem [25] investigated that by using personalized feature sets for each person can improve the performance of a signature verification system. Parks *et al.* [101] purposed that if different thresholds for different individuals is used and the possibility based on the threshold in credit card identification can be done on the basis of the goods purchase and the credit rating of the person. [75] identified that the accomplishment of the best 10 common features was considerably poor in comparison of 10 individual features for each subject.

Signature acquisition

We used a WACOM Intuos graphical tablet to capture signatures. Figure 3.1.1 shows an image of the tablet. Data is sampled at a rate of 200 Hz. Each sample consists of the current x and y position of the pen tip on the surface of the tablet. The pressure the pen tip exerts on the tablet surface quantized to 64 levels. The angle the pen makes with the x and y axis respectively. These raw signals are plotted in Figure 3.1.2 together with the signature they were sampled from. The tablet has a resolution of 1000 lpi.

The rest of this chapter is organized as follows. In Section 3.2, we explain in details what we mean by signature processing. Section 3.3 deals with the theoretical framework used in this chapter. Some useful notions of statistical modelling are discussed in Section 3.4 where optimal feature selection method is discussed in Section 3.5. Finally, we provide a brief summary of this chapter in Section 3.6.

3.2 Signature preprocessing

The raw sequences of signature components are not in a form which is suitable for modelling. The hidden Markov models (HMMs) will require signatures to be in a canonical form prior to training and verification. Samples of a signature can be transformed dif-

Figure 3.1.1: The device for signing process

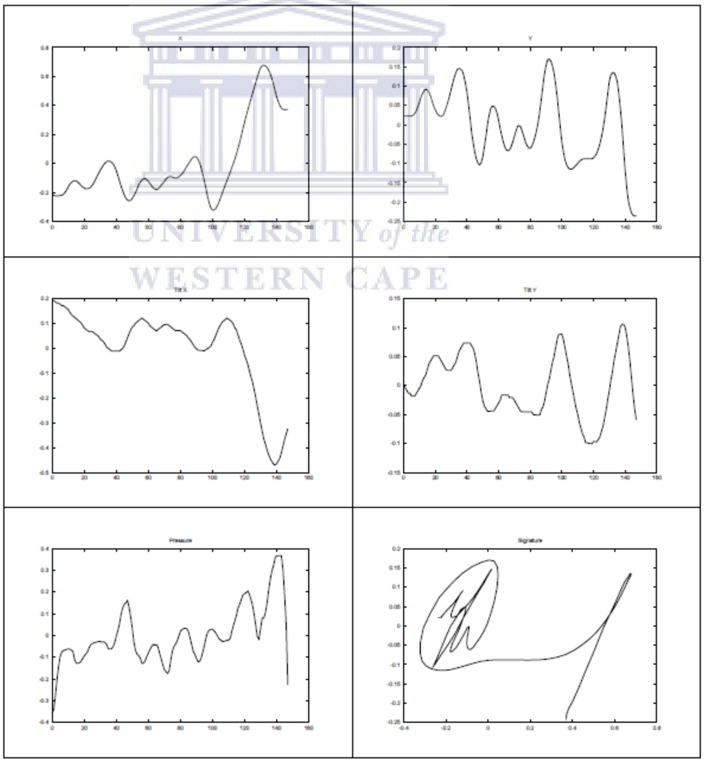

Figure 3.1.2: Signature: positional, pressure and tilt information

ferently by rotation, translation and scaling when initially sampled from a user. The preprocessing actions seek to convert a raw signature into canonical form with respect to orientation. Once in this form, we can include the signature in either the training database or subject it to verification. The ideal scenario is for a system not to restrict a writer to sign on a certain baseline. Inspection of a couple of signatures revealed that even if a baseline is provided, signatures are not guaranteed to follow them. Some writers start signing on the baseline but progress a direction pointed to the top right of the signing space. It goes to argue whether this imaginary baseline can be assumed to be constant. Furthermore, signatures seldom start exactly at the beginning of the baseline.

Generally speaking, they might start anywhere in the first quarter of the line. The size of signatures also vary from one exemplar to another. These factors opt for some procedure to convert a signature into a uniform reference frame. This describes some approaches to solve this problem as found in the literature.

Rotational Invariance

Rotational invariance is achieved by calculating an angle θ of corrective rotation about the centroid of the $(x; y)$ samples. Rotating the signature by θ normalizes it to a line running through the centroid. We calculate θ by maximizing the deviation of the data in one direction, e.g., the x direction 1 The normalized signature is obtained as follows: The mean μ_x of the x sequence is calculated by

$$\mu_x = \frac{\sum_t^T x_t}{T},$$

and the standard deviation of the x sequence from μ_x by

$$\sigma_x = \sqrt{\frac{\sum_t^T (x_t - \mu_x)^2}{T}}.$$

In order to maximize the deviation, we need only to maximize

$$\sum_{t}^{T}(x_t^{\star} - \mu_x)^2,$$

where $x_t^{\star}$ indicates a rotated x value. We need to choose a point which is rotational invariant within the framework of the normalization scheme [143]. It is easy to show that the centroid is such a point within this scheme. A rotation about the centroid $(\mu_x; \mu_y\ )$ can be expressed as

$$x_t^{\star} = (x_t - \mu_x)\cos(\theta) + (y_t - \mu_y)\sin(\theta) + \mu_x,$$

By substituting $x_t^{\star}$ into the equation to be maximized, we obtain $f(\theta)$ as

$$\begin{aligned} f(\theta) &= \sum_{t}^{T}\left[(x_t - \mu_x)\cos(\theta) + (y_t - \mu_y)\sin(\theta) + \mu_x - \mu_x\right]^2, \\ &= \sum_{t}^{T} a_t^2\cos^2(\theta) + 2a_tb_t\cos(\theta)\sin(\theta) + b_t^2\sin^2(\theta), \\ &= \cos^2(\theta)\sum_{t}^{T} a_t^2 + 2\cos(\theta)\sin(\theta)\sum_{t}^{T} a_tb_t + \sin^2(\theta)\sum_{t}^{T} b_t^2, \\ &= \cos^2(\theta)P + 2\cos(\theta)\sin(\theta)Q + \sin^2(\theta)R, \end{aligned}$$

where

$$\begin{aligned} a_t &= x_t - \mu_x, \\ b_t &= y_t - \mu_y, \\ P &= \sum_{t}^{T} a_t^2, \\ Q &= \sum_{t}^{T} a_tb_t, \\ R &= \sum_{t}^{T} b_t^2. \end{aligned}$$

Differentiating the above function f with respect to θ, and setting the right hand side equal to zero, we obtain the roots as

$$\pm \cos^{-1}\left(\pm\frac{1}{\sqrt{2}}\sqrt{\frac{1+(P-R)}{\sqrt{P^2+4Q^2-2PR+R^2}}}\right)$$

and

$$\pm \cos^{-1}\left(\pm\frac{1}{\sqrt{2}}\sqrt{\frac{1+(R-P)}{\sqrt{P^2+4Q^2-2PR+R^2}}}\right).$$

We adopt the value for θ closest to zero which will result in a maximum of f_θ. We also need to make sure that the time series evolves in a consistent direction by imposing a possible 180^o rotation. This is done by fitting a least squares line to the rotated x data. If the slope is less than 0, we infer that the signature strokes are increasing from right to left and not left to right as is the case with normal signers. We then apply an additional 180^o rotation to conform to the norm. This means that the tablet can be upside-down when signing without affecting the normal operation of the system.

Unfortunately, there does exist a scenario where this scheme fails. If a writer's signature is a borderline case where the maximum deviation varies from one axis to the other with different samples, this scheme will result in an inconsistent perpendicular normalization. Fortunately, these signatures are rare as signers usually sign in a predominantly left-to-right fashion. One partial solution to this problem is achieved by providing a baseline. Now, when a θ value larger than some acceptable deviation from the baseline, e.g., 45^0, is attained, we conclude that the algorithm is confused by a borderline case as explained earlier. In such a case we have to trust the writer blindly and perform no rotation. Another solution is to calculate the ratio of variances in the x and y directions. If this ratio is within a threshold distance from 1.0, we conclude that the signature is not suitable for normalization by this technique.

The pressure signal is invariant to the rotation of a signature baseline. However, it seems that a common oversight is the pen tilt signal which needs to be transformed

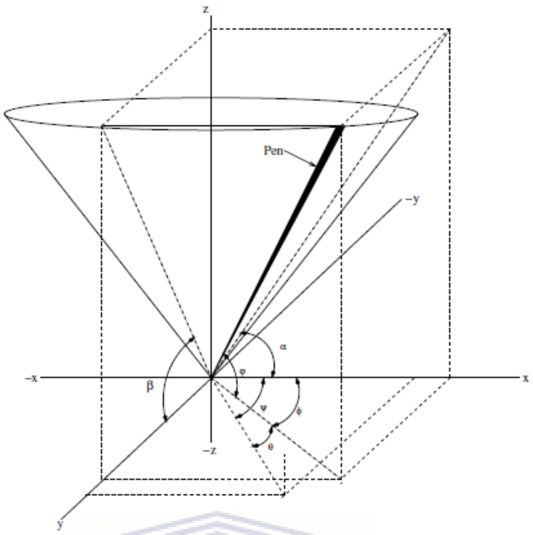

Figure 3.2.1: Pen tilt rotation

along with the positional information. The tablet reports the tilt as the angle the pen makes with the x and y-axes on the tablet surface. To understand why a rotation might affect these tilt angles, imagine the pen coinciding with the surface of a cone where the cone tip is situated at the pen tip. A rotation of the coordinate system (i.e., the tablet surface) will result in a sweeped cone. To calculate the new tilt angles, we create a top-view of the pen using the reported tilt angles. The pen is then rotated by the angle θ calculated in the previous paragraph. Images of this rotated pen are then projected back onto the the xz- and yz- planes, respectively, to calculate the new angles. Figure 3.2.1 graphically depicts this process. In the figure, α is the angle formed with the x-axis and β the angle formed with y-axis. To transform α and β, we project the image of the pen as seen from above, onto the xy- plane resulting in the pen angle θ in the figure. This projected pen vector is then rotated by the normalization angle θ mentioned previously, resulting in a pen vector with angle ψ . From this vector, we project the image of the pen onto the xz- and yz- planes, respectively and calculate the new values for α and β.

Translation Invariance

To compensate for the fact that a signer need not always start on the exact same place on a baseline, we apply a translation to the signature. After this operation, the leftmost part of the signature will coinside with the vertical axis of the two-dimensional Cartesian plane and the bottommost part with the horizontal axis. The translation vector is simply taken to be the smallest coordinate value in the signature on both axis. This vector is then subtracted from each $(x; y)$ sample.

Scaling Invariance

To achieve scaling invariance, we need to find a scaling factor which will transform the size of the signature to be contained in a 1-by-1 box yet maintain its aspect ratio. The dimensions of the box are arbitrary as we merely need a unifying signature size. The scaling factor is taken as the minimum of $\frac{1}{X_{dim}}$ and $\frac{1}{Y_{dim}}$ and both the x and y components are multiplied by this factor. This operation maintains the original aspect ratio.

Acquisition device invariance

Even though there was no need for this step to be performed in our experimental system, it will be important for a production system to be sure that differences in hardware do not hamper the performance of the system. The software drivers on different platforms may also translate the sampled values to different intervals than what is reported by the hardware. As tablet brands may quantize the pressure signal differently, we need to convert the pressure values to a uniform interval. The default tablet we use in our system reports the pressure in a range of [0; 63] We scale these values to reside within the interval [0; 1]. The angle of tilt is reported to be in the range [1; 1] by the software drivers used in the graphical user interface. On the other hand, the benchmark database also described records the tilt in degrees. As we interpret the tilt value as an explicit angle during pre-processing, we opt to convert the sampled value to [90^o; 90^o]. Some systems might require for uniform sampling rate as signature duration is seen as an important feature.

If a signature is obtained from a tablet at a different rate than what was used to derive the model parameters, it might need to be resampled to conform to a unifying standard. To understand why this is important, we visualize a scenario where the system is used in an open commercial environment. Different transaction end-points might deploy different tablet brands. For the system to function correctly across different platforms, it might be necessary to agree on a uniform sampling frequency. Resampling to such a frequency can be performed by fitting an interpolating spline to the data. We make use of relative durations wherever timing information is required. Our final signature likelihood values are also normalized with respect to duration. We therefore do not need to perform this action.

3.3 Arc-length parameterization

The arc-length parameterization is the preferred means of reference to signatures in [93, 95]. As we sample signatures from a tablet at a fixed sampling rate, we have a time parameterization of the signature signals. Apart from this parameterization, we would also like to conduct experiments on the arc-length parametized versions of the signatures. The arc-length parameterization of a curve can be constructed from another differentiable parameterization by the following process:

1. The cumulative arc-length of a parameterized curve $r(t) = (x(t), y(t))$ measured from $t = a$ is given by

$$l(t) = \int_a^t \parallel \bar{r}(\tau) \parallel d\tau = \int_a^t \sqrt{x'^2(\tau) + y'^2(\tau)} d\tau.$$

2. The inverse of the arc-length function is used to create an arc-length parameter-

ization of a curve by composition

$$s(u) = (x \circ l^{-1}(u), y \circ l^{-1}(u)).$$

A necessary and sufficient condition for a curve to be in arc-length parameterization form is

$$l(t) = t \qquad \forall t.$$

By taking the derivative of $l(t)$ and the previous condition we arrive at another condition for the arc-length parameterization

$$x'^2(t) + y'^2(t) = 1,$$

from this we can see that

$$|\bar{r}(t)| = 1$$

which means that the arc-length parameterization describes the traversal at unit-speed of the curve.

Velocity

Velocity is regarded as the most important discriminating feature of signatures [105] with the odd exception [95]. It is argued that a forger may succeed at duplicating the shape of a signature but will have difficulty in doing so at the same tempo as the original signer.

Velocity can be calculated from the positional signals as (see [74])

$$|V(n)| = \frac{\sqrt{\triangle x(n)^2 + \triangle y(n)^2}}{\triangle t(n)},$$

where $\triangle f(n) = f(n+1) - f(n)$. Alternatively, the first derivatives, $v_x(t) = D_t x(t)$ and $v_y(t) = D_t y(t)$, of functions fitted to the x and y-signals can be taken and used to compute velocity as

$$V(t) = \sqrt{V_x(t)^2 + V_y(t)^2}.$$

As mentioned earlier, we fit cubic smoothing B-splines to the measured signals which give access to derivatives 3. We prefer to stick to the term ‘velocity’ as this is how this feature is commonly coined in the literature. Physicists, however, might point out that it should in fact be referred to as ‘speed’. Velocity is the rate of positional change of an object in a certain direction whereas speed is the magnitude of such a velocity vector. Speed by itself cannot represent a signature unambiguously. Even though it is highly unlikely for a forger to recreate a velocity profile in a random way, predictable behaviour of a system is an important consideration. Note that we can easily create an actual velocity signal by creating two- dimensional observations of speed and direction. Speed is in itself rotational invariant which is an important attribute if no normalization of signatures is performed.

Path tangent

The path tangent is the missing directional component of velocity. In the previous section, we have shown how V_x and V_y are calculated. The path tangent is related to these values by

$$T_\theta = \tan^{-1}\frac{V_y}{V_x}.$$

The literature agrees that cubic smoothing B-splines are an appropriate choice [53, 54, 66].

Acceleration

Systems based on instrumented pens measure accelerations involved during the signing process directly. When acquiring signatures through tablets we generally need to calculate the acceleration profiles from the positional signals as a post acquisition step. We achieve this by taking the second derivative of the splines fitted to the sampled x and y-signals. The total acceleration is related to the axial accelerations by

$$A(t) = \sqrt{A_x(t)^2 + A_y(t)^2}.$$

It should be noted that taking second derivatives is a process known to be numerically instable [54, 95]. We will however still explore its discriminating ability as a feature signal.

3.4 Statistical modelling for feature extraction

The well established field of modern statistics provides a solid basis from which to build pattern recognition systems. Various statistical techniques have been applied to ASV. In a sense, one can argue that most ASV systems will incorporate some fundamental statistical concept somewhere. This section contains studies which makes use of predominantly statistical concepts.

Feature-based statistical methods apply transformations to the data which result in a set of features. They are chosen to expose differences between genuine signatures and forgeries. The feature extraction process can be seen as signature compression. The challenge is to extract features which do not discard relevant information. Dynamic features describe aspects which are not apparent from an examination of a copy of a signature. A forger needs to duplicate the shape and the way it was signed. Therefore the verification procedure must include a mixture of both shape and dynamic-related features. Nelson *et al.* [98] used a set of 25 features. Some examples are the total signature time, the root mean square speed, the integrated absolute centripetal acceleration, a direction histogram (0-2 divided into eight sectors) and the X, Y speed correlation. It is desirable for shape-related features not to be strongly correlated to dynamic features. There may be the extra constraint that the parameters of features must not exceed the storage limit for a particular application (e.g., 80 bytes for credit cards).

According to Nelson *et al.* [98], a feature is a good discriminator between genuine and forged signatures, if its values on genuine signatures constitute a cluster which can be separated with high accuracy from that of forgeries. To verify a signature, they computed its feature vector and compared it to a template vector by some distance metric. The study reports on Euclidean, Mahalanobis and Quadratic distance models. In this study, they assumed the feature vectors to come from mixtures of multi-variate Gaussian probability density functions. They claimed that the statistical properties of genuine signatures should be reasonably predictable as they are produced by a single known signer (in contrast with forgeries for which no a priori knowledge is available). Then they defined decision rules for both the cases where forgeries are and are not available.

In Kashi *et al.* [66], 23 global features are used. These features are divided into roughly two categories: shape-related and dynamical features. Care is taken to ensure that the shape-related features are not strongly correlated to the dynamical features. For each of the features, the mean and the standard deviation are calculated from training samples for a specific signer. These are then used in a joint distance measure to determine the degree of similarity of an unknown signature. This study shows that a verification system need not comprise of only a single modelling approach. The feature-based model is further augmented by Kashi *et al.* [66], call stroke direction coding. The results show that the combination of SDC and this feature based approach outperforms each approach on their own.

One of the most attractive qualities of feature based verification systems, is the relatively small amount of memory needed to store a signature model. This is an important consideration for many current commercial applications where storage ability is restricted e.g., credit/SMART cards.

A signature can also be seen as a stochastic process. In [89], it is shown how the random impulse response for a system is calculated where the relationship with the sampled $(x; y)$ signal is

$$y(t) = \int_0^T h(t, \tau)x(\tau)d\tau,$$

with $h(t, \tau)$ the random impulse response. For verification, a distance measure between two sequences of random impulse response parameters is defined.

Matsuura and Togiishi [90] approximated a signature by a piecewise linear function, i.e., the locus of pen movement is approximated by line segments. Each line segment is depicted in magnitude/argument form as is commonly used to represent complex vectors. For this sequence of magnitude/argument pairs, a two-dimensional AR model is defined and its parameters are obtained by solving a set of simultaneous equations. The cross spectral density is calculated for the AR parameters. Then the discrete cosine transform is performed on the cross spectral density and the logarithm of the transform coefficients is the features representing a signature. The distance between the test signature's feature vector and a reference feature vector is calculated and if within an acceptable threshold difference, the signature is classified as authentic. The reference feature vector is built from a randomly selected subset of the signature samples of a subject.

In [103], Paulik *et al.* explored a vector autoregressive model. A sampled $(x; y)$ sequence is resampled to a fixed sequence of length 512. This sequence is divided into a fixed number of sections. The sections are then each modeled by a VAR. It is argued that the VAR coefficients matrix eigenvalues, the scalar VAR coefficients, the mean vectors and the noise measures built into the model, own intraclass invariant properties. This makes them excellent candidates as features for classification and verification. The eigenvalues of the VAR model coefficients matrices are used instead of the matrix elements themselves to reduce the feature vector size. The distance measure used to compare a suspect feature set with a reference feature set involves a discretization of the features to obtain likelihood values from a frequency matrix. As usual, a threshold distance decides on the authenticity of a signature. This study also compares this approach to a subset one-dimensional approach to assess whether the extra parameters obtained in this study yields a significant performance increase. The study concludes that, even though the results have improved, the improvements are not statistically significant to warrant the computational overhead.

3.5 Optimal feature selection

As we have seen in the previous section, feature-based systems compute features from sampled signatures where each feature represents some characteristic of a signature. This is called feature extraction. Features can be chosen with the hope that they constitute a consise representation of a signature. Considerations taken into account when selecting features in [98] are that they must be

- insensitive to variations in genuine signatures, and
- good discriminators between genuine signatures and forgeries.

Because no a-priori knowledge is available about which of the vast array of possible features will give the best discriminating power, a feature set might contain a lot of redundant information with no guided way of pruning them. The objective of feature selection (as opposed to feature extraction) is to obtain a reduced set of features which contains essentially all the discriminating power of the original set. Feature selection addresses the following aspects of a feature based verification system

- efficiency through the removal of redundant information,
- speed by reducing the dimension of the feature vector,
- performance by working only with an optimal feature set.

In general, most feature selection techniques follow the same basic procedure. The starting point is a large set of features which the analyst believes to be useful for discriminating between samples. The discriminating power of each of the features or combinations of features is determined by performing statistical tests on a training set of data which is believed to adequately represent the population. The combination of features which yields the best performance (by some criteria) and which contains the minimum number of features is deemed the best feature set. Furthermore, the optimal feature set need not be the same among different signers. Different approaches to finding an optimal set are reported.

Lee [76] defined a set of 49 normalized indicators extracted from a positional signature signal sampled from a tablet. For a subject, the k most important features are selected amongst these by ordering the features according to their maximum distance from the rest of the entire population. The distance measure involves the mean and variance of a feature obtained from a training set. This results in an optimum individualized feature set. The study also presents a common feature set composed of those features with the highest frequency of appearance in all the individualized feature sets.

In Crane and Ostrem [25] discussed the simplest methods select features through trial and error or brute-force. Such an approach is time-consuming as there can potentially be a vast number of combinations to search through. Sub-optimal searches reduce the size of the search space by imposing certain structural or traversal restrictions on the search tree. Fairhurst and Brittan [35] considered the parallel strategies for feature vector construction . It is shown that there exist inherent parallelism in the feature selection process which can be used to perform the task on a parallel computer. Various parallel algorithms are implemented and compared. The possibility of using a transputer is attractive as it reduces the amount of time needed to select an optimal feature set.

As the name would suggest, the genetic algorithm finds its origins in the field of Biology. It is based on the way living organisms evolve on a genetic level to attain the best genes suitable for their situation. The algorithm employs these principles to find an optimal solution to a problem at hand. The challenge here is to find an encoding for the problem in terms of chromosomes. Xuhua *et al.* [150] showed how this algorithm can be applied to the problem of feature selection. It is necessary to select features of signatures which can overcome the dilemma of intra-and inter-personal variability. Xuhua *et al.* [150] stated that not all sampled points of a signature are necessary for verification. Experts concentrate on some particular parts which have distinguishing features when they engage in signature verification. What is more, different features in different parts of the signature must be used for the verification. A signature is a sequence of time ordered data and the combinations of features are unlimited. It is very difficult to predetermine an optimal set of features.

The result of the selection is not even unique. The genetic algorithm has a high degree of ability to solve this problem. This study presents a novel method to select partial curves and features of the curves of signatures for verification using the genetic algorithmic. The study also proposes a new crossover method in order to determine the number of partial curves. The described system consists of a feature selection part and a signature verification part. The location of partial curves and the features of the curves used for the verification are encoded into the chromosome. The length of a chromosome, i.e., the number of loci, corresponds to the number of partial curves of the signature. The genotypes are then modified by the genetic algorithm using the local improvement mechanism. Each chromosome is evaluated by a fuzzy network and the chromosome's fitness value is calculated. The one with the highest fitness value is selected. This elite chromosome includes the best set of partial curves and features of each curve for a true signature.

It is perhaps suitable to end the modelling section with a paper which dares to challenge accepted beliefs. In [95], Nalwa disagreed with the general notion that velocities and forces plays a pivotal role in ASV. The reason for this is that no evidence could be gathered to show a signer's pen dynamics are consistent enough to be used as distinguishing features in verification. Foremost, for two signatures to be declared as produced by the same individual, it is necessary for the shape of both to match closely. The author perseveres that for ages we have relied on visual examination of signatures to decide authenticity. He finds it difficult to justify the jump to time related information.

The author claims that all the subjects in his study could produce their signatures both as are and deliberately without visual deterioration. The study presents a number of novel aspects to ASV. The concept of jitter is introduced as a quantity measuring the act of a forger constantly correcting the pen trajectory to conform to an a priori curve. To make a signature independent of orientation and aspect, it is normalized. This is done by fitting a polygon to the ordered set of samples and use the global axes of maximum and minimum inertia running through the global center of mass and rotate the signature to normalize these axes.

The rotated signal is then scaled to normalize the aspect. This normalized signal is then parameterized over its length (instead of time). By using a moving coordinate frame, the center of mass, torque and moments of inertia at the center of the window is calculated using a Gaussian weighting function. These derived signals are used to characterize a signature. To compare a signature to a reference characteristic function set, the two sets of functions are length warped (in contrast with the more familiar time warping) as to maximize the sum of the weighted cross correlation of each function with respect to its model. The error between each characteristic function and its reference model is computed. The study then uses what it calls the harmonic mean to quantize the global error based on the joint error for the jitter, aspect and warping distance. The study describes in detail various databases used for tests and presents a real implementation combining SMART card technology, a proprietary digitizer and a notebook computer. This study highlights various topics which are central to the problem of ASV.

3.6 Summary

We have discussed in this chapter that preprocessing is very first step before we do the feature selection. This chapter provided in detail information regarding the methods used to preprocessing of features and their importance. We also discussed about the various global features and local features. Then we explained the statistical modelling of the features and optimal feature selection. This will be useful in understanding the next chapter which explains the different signature modelling techniques.

Chapter 4

Methods for signature modelling

This chapter focus on the different types of the signature modelling techniques such as dynamic time wrapping, hidden Markov models, fourier transform and Artificial neural networks. Furthermore we also discuss about the Neyman Pearson criterion and support vector machines.

4.1 Introduction

Automatic signature verification is a very attractive field of research from both scientific and commercial points of view. In recent years, along with the continuous growth of the internet and the increasing security requirements for the development, the field of automatic signature verification is being considered with renewed interest since it uses a customary personal authentication method that is accepted at both legal and social levels. Moreover, recent results achieved in international competitions using standard databases and test protocols have revealed that signature verification systems can have an accuracy level similar to those achieved by other biometric systems. Finally, different from physiological biometrics, handwritten signature is an active method that requires the user to perform the explicit act of signing. Thus, automatic signature verification is particularly useful in all applications in which the authentication of both transaction and user is required.

Accordingly, the number of possible applications for online signature verification is continuously growing along with the development of more and more sophisticated and easy-to-use input devices for online handwriting acquisition. For example, automatic signature verification can be a valuable contribution for controlling access security in computer networks, documents and databases. Applications of this can be seen in health care applications for medical record access and in the areas of passport and driving license applications. Automatic signature verification has important applications in online banking, monetary transactions, and retail point of sale. For instance, it can be used to replace the current practice of signing paper credit card receipts. In this case, the verification process can be performed by comparing the live online signature of a user with the biometric information of his/her handwritten signature that can be stored in a personal smart card to verify that the person using the card is the rightful owner.

At the heart of an Automatic verification system, we find the modelling technique employed. Even though the performance of a system depends largely on the degree to which all the aspects of the system work together, the applicability of the modelling technique and ability to recognize genuine signers and forgers are the most important factor in the quality of a verification system. It is therefore no surprise that it is in this part of the field where the most effort is exerted. This section summarizes various modelling approaches applied to automatic signature verification. The list is by no means complete but we hope it covers most of the major research current directions in the field.

The rest of the chapter is organized as follows. We explain about modelling with dynamic time wrapping in Section 4.2. A detailed work is discussed about hidden Markov model and their applications in Section 4.3. Fourier transforms for signature verification is discussed in Section 4.4. Then we study about artificial neutral networks and support vector machines in Section 4.5 and 4.6. Finally, we give a brief summary in Section 4.7.

4.2 Dynamic time warping

Dynamic time warping (DTW) stems from the field of dynamic programming. The general idea of dynamic programming is to find a least cost path through a cost matrix in an attempt to optimize some process. The challenge is to define a suitable cost function for the problem at hand. Dynamic time warping finds a non-linear time alignment between two sequences to compensate for non-regular stretching or compression in the sequences. If two sequences show similar overall shape, DTW can find a unifying time function which will align the two sequences in a way minimizing the distance between them. If they are not of similar shape, DTW will find some alignment but the warped sequences will remain far apart. It goes about finding an alignment by placing one of the sequences on the vertical axis of a discrete matrix and the other on the horizontal axis. Each matrix position is then set to the cost of aligning the partial sequences up to that position on each sequence so the distance between the sequences are minimized. Various cost functions can be used each having particular characteristics. For an in-depth discussion of DTW, see, [111, 118].

One of the earliest studies on ASV [159] uses dynamic time warping to find a non-linear alignment between signatures. Prior to alignment, the equivalent force function varying over time is calculated from the dynamically sampled data. The function is computed as follows

$$f(t) = d'' + \left[1 + \left(\frac{\mu p(t)}{\nu}\right)\right] d' \text{ with } \mu = \sqrt{x'^2 + y'^2},\ p = \frac{p}{M},$$

where d is the displacement function, p is the writing pressure function, v is the writing speed, 1, and M are the viscosity coefficient of the human hand, the friction coefficient between pencil point and writing surface and equivalent mass hand-pen coupling, respectively. This function is supposed to be a more direct representation of the control timing information of the writing movement than for instance the pencil point displacement. The force functions of two signatures are aligned by DTW. Extreme time warping during alignment is prevented by placing constraints on the DTW routine.

The output of the alignment is a normalized distance between the two signature representations. This distance is compared to a personalized threshold which is determined experimentally. Most verification systems compute some measure of similarity or distance between two signatures and compare this to a threshold value which in itself can be obtained in various ways.

It might be attractive to reduce the task of signature verification to two steps. That of low level feature detection followed by some standard method of feature-vector comparison. The price one pays for this simplification is that the overall result is only as good as the features selected. The method described by Hastie and Kishon [53] is based on the approach of representing signature data by functions of time (instead of a number of low-dimensional parameters or features). The study makes use of dynamic time warping and geometric shape analysis to perform verification. The DTW is used to match the speed signals of two signatures. It is believed that DTW should not be used to compensate for other variations such as Euclidean shape transformations which will be the case, should positional functions be aligned without being normalized first. Time warping is necessary as writing speed will change from one signature to the next, as will its consistency over different regions. The slant might increase as the writing speed increases resulting in non-regular deformation of the speed signal. Non-linear time alignment produced by DTW, for instance, can compensate for this.

Apart from alignment distances inspected for verification, further checking is done by comparison of signature segments. A segmentation of a signature into pieces which exhibit little oscillations in its various features can be achieved by using points of high curvature as the segmentation boundaries. The curvature signal has to be computed from second order derivatives though, which is numerically unstable. Therefore, the simple observation that points of high curvature coincides with points of low speed, is used instead to obtain a segmentation of a reference signature. The reference signature is selected as the signature in the training set which deviates the least from the other signatures during DTW. A segmentation is created by dropping pieces of the signal where the speed component is less than a threshold percentage (e.g., 15%) of the mean speed. For each of these segments, a template average shape is estimated up to an affine transform which allows for differences in location, scaling, orientation and shear. This template is described as

$$Y(u) = \begin{bmatrix} x(u) \\ y(u) \end{bmatrix} = A(u)F(u) + \mu(u) + e(u),\ 0 \leq u \leq 1,$$

where

- $F(.)$ is an idealized template or "mean" signature for the writer,
- $A(.)$ is a 2 * 2 affine transformation matrix, and
- $e(.)$ is a stochastic vector of departures from the model.

Hastie and Kishon [53], performed checks on both the speed and shape of a signature. The estimation procedure discovers the parameters for these elements. These average segments are then concatenated to form a template signature. Verification is carried out at both the DTW stage and the affine transform stage. If the discrepancy at the DTW stage is not big enough to conclude the authenticity, the signature is segmented and the segments affinely transformed to match the template. The least squares distance between the template and suspect signature segments are then used to decide on the authenticity. This phase inspects a possible forgery for unacceptable shape variations. It is extremely unlikely that a forger can mimic both the shape and relative speed with which a person signs their name. This is even more so if the signature to be forged contains exotic which are often illegible but consistent in the victim's signature.

Munich and Perona [93] obtained signature dynamics from cameras. A normal pen is tracked by applying optimal signal detection techniques to images sampled from a digital camera. The tracking algorithm cannot distinguish between the up and down state of the pen so the entire pen trajectory is recorded. Various different parameterizations of the signatures were tested in this study. The affine arc-length parameterization has been found to be superior to arc-length and time parameterizations. The sequences are aligned by dynamic time warping even though the parameterization is not necessarily by time. A reference signature is obtained by finding the training signature which shows the least deformation during alignment with all the other samples. The average of this alignment with all the other signature samples represents a prototype signature in the system. The distance between a reference signature and a test signature is evaluated in various different ways in search for the optimal distance measure for this system. A harmonic mean measure was found to outperform residual distance, correlation and weighted correlation when establishing time correspondence between two curves. The study claims, somewhat contradictory to the general opinion, that dynamic information is of less importance than static information during verification.

The feature based approach is augmented by stroke-direction coding (SDC). With SDC, Kashi *et al.* [66] attempted to model hand movements that produce a signature. A signature is divided into a fixed number of time-ordered links called strokes, where each link is approximately of the same length. A stroke is described by a number indicating the general direction of pen movement within the stroke. A non-linear alignment through dynamic time warping is used to establish the deviation of the SDC vector of a test signature from a reference SDC vector. This deviation and the feature based error measure are combined for verification.

Dynamic time warping is quite often used to find an alignment between sequences. Other approaches do exist however. In Wu *et al.* [144], a signature was represented by a static feature sequence which is the sampled $(x; y)$ sequence and a dynamic feature sequence which is the velocity computed from the static sequence. To match an input signature with a reference signature, the two sequences have to be aligned. This study proposes a technique called split-and-merge. In contrast with dynamic time warping which is a piecewise advancing match algorithm, split-and-merge is a top-down approach. It proceeds in a recursive fashion by splitting the reference sequence in the middle and the test sequence at such a place that after refining the two subsequences and merging them, it best matches the reference sequence. A subsequence is refined by removing the non-uniform compression or spreading among sub-patterns relative to the reference sequence.

The refined subsequences are merged and interpolation is used to make the reference and test sequence the same length. After this, the distance between two sequences are measured and compared to a threshold value derived from the training set. An input signature is deemed genuine if both of its coordinate and velocity distance from the reference template are less than the respective coordinate and velocity thresholds. Results show that there is a split-and-merge recursion depth beyond which no performance gain is achieved.

We now brief summarize the results of a study which compares DTW with other approaches. Parizeau and Plamondon [100] discussed Dynamic Time Warping, Regional Correlation and Skeletal Tree Matching. The idea of regional correlation is to cut signals into regions and to correlate corresponding regions over different time lags to find the best possible match. The dynamic time warping variation used in this study is based largely on work done in the field of speech recognition. For skeletal tree matching, a tree representation is created for each of the two signals being compared. The tree representation seeks to capture peaks and valleys in the waveform together with their self-embedded structure. The methods are compared with respect to verification error rates, execution time and number and sensitivity of parameters. The comparisons are extended beyond normal signatures to handwritten passwords and initials. Furthermore, the tests are conducted using positional, velocity and acceleration signal representations, respectively. A variance analysis on the individual results shows that no algorithm consistently outperforms the other.

4.3 Hidden Markov models and their applications

Apart from automatic signature verification, hidden Markov models (HMMs) are also used with a great deal of success in automatic speech recognition and molecular biology. Essentially, they extend the well-known concept of Markov chains and are thus founded on solid statistical principals. HMMs comprise of a state graph connected by probabilistic transitions. Each state can accept an observation with some probability. The observation at each time instance need not be single-variate and can be either discrete or continuous. HMMs allow for the modelling of non-linear time variance in sequences of observations by dictating transition probabilities between states or imposing explicit state durations. This proves to be a handy feature when working with signatures which exhibit time warping amongst different samples originating from a single signer. Such a time-warping profile can serve as a distinguishing feature if captured by a model which is indeed the case for HMMs. Generally speaking, verification systems based upon HMMs are concerned with finding appropriate sequences of observations to represent a signature and to attach sensible semantics to model states. For a more in-depth discussion of HMMs, see Chapter 5 and references[110, 111].

The absolute angular direction of signature samples as a function of the distance along the signature trajectory is used to represent a sampled signature in[156]. This sequence of angles are divided into a fixed number of segments. A formula incorporating all the angles in a segment is used to calculate a discretization code representing a segment. This sequence of codes is then presented to a HMM. The theory provides for the calculation of a likelihood that a sequence was generated by the process being modeled by the HMM. This value for a test signature is compared against a threshold likelihood value to verify the authenticity of the signature. This approach in a sense counteracts the time warping ability of HMMs by implicitly assuming that an equal segmentation will group similar subparts of a writer's signature. The equal segmentation adopted from speech recognition cannot be applied with equal success to ASV due to the huge difference in the amount of samples available. Nalwa [95] agrees with our view as signature sequences are not long enough for a model to recover from segmentation errors.

In Dolfing *et al.* [30], samples are blocked into segments bounded by points where the velocity v_y in the y direction crosses zero. It is argued that segmenting on these points results in a size independent representation. A 32-component feature vector is derived for each segment. Linear discriminant analysis is performed on this feature vector and the N most discriminative features are selected to represent segments. A left-right hidden Markov model is used to model the sequence of feature vectors. An adaptive threshold for a signer is computed from the average likelihoods for the training set combined with a system dependent observation set.

Much the same as in Kashi *et al.* [65, 66] reported on a method combining global and local features. For a description of the global features. For the local feature based part, a hidden Markov model with explicit duration modelling is used. This results in a variable duration hidden Markov model. This model is also referred to as a hidden semi-Markov model (HSMM). In [65], Kashi *et al.* used a specific HMM configuration to approximate a HSMM. Each HSMM state is decomposed into a number of unit-duration substates, resulting in a HMM with a larger number of states than the HSMM. Each sample in a signature is represented by an inclination angle and the difference between adjacent inclination angles. These values are quantized for use with a discrete HMM. In such a HMM, no assumption about the distribution of the data needs to be made (as opposed to continuous HMMs).

The calculated likelihoods are divided by the number of sample points to reduce the effect of signing time variations on the algorithm. The difference between the likelihood for a test signature and the average likelihood for the training set is used as an error measure to determine the authenticity of a signature. The global and local errors are combined using a Euclidean distance measure to reach a conclusion. Results show that the combined use of global and local features perform better than any of the two parts on their own.

McCabe [91] extended the idea of signature verification to a system where a signature is substituted by a written password. This means that not only does a forger have to imitate the writing dynamics, but also guess the statics, i.e., the password. A written sequence is normalized to a horizontal baseline. It is then segmented into strokes delimited by consecutive minima of the absolute pen-tip velocity. Each stroke's net direction is obtained by placing the starting point of the stroke at the origin of the Cartesian plane and observing the quadrant of the end- point of the segment. This results in a discretization of size four. An element for recording pen-up events is added giving a codebook size of five elements. The sequence of discretized observations is then modified by repeating symbols proportional to the length of a segment. This modification enables the algorithm to make better use of the time warping ability of the HMM. To model the sequence of symbols, he used a very compact HMM with only five states.

4.4 Fourier transforms for signature modelling

The Fourier transform is probably the most widely used mathematical tool in signal processing applications today. It has found its way into signature verification as well. This section explores studies using what we deem to be more traditional signal processing techniques including the Fourier transform and spectral analysis made possible by it.

Different signature samples of a writer almost always exhibit instabilities of some kind. Wen *et al.* [140] introduces a distortion measure to to deal with this fact. This distortion measure is based on DTW and serves as a first step in the verification process. If this phase cannot decide conclusively on the authenticity of a suspect signature, a next phase based on spectral correlation is employed. For this, preprocessing of a signature consists of resampling a linear interpolation of the signal and including velocity information in the new signal. This signal is transformed into the frequency domain by a FFT. Linear correlation is used to find the similarity between the spectra of an input and reference signature. As usual, the correlation coefficient is compared to a threshold value. In calculating the correlation coefficient, the weight of each frequency component depends on the stability of the component as deduced from the training set.

In [145], Wu *et al.* sampled coordinates of a signature are converted to the frequency domain by the fast Fourier transform. To smooth out sharp spikes in this frequency spectrum, the log of the Fourier coefficients are taken to represent the signature as a logarithmic spectrum. Through principal component analysis based on scatter matrices, only a small amount of these coefficients are extracted to represent a signature. The similarity of the changing rate of coordinates between two signatures can be characterized by the similarity of the coefficients of logarithmic spectrum. A reference template for a signer is obtained by taking the mean values of the transformed training sequences.

Dimauro *et al.* [28] used a local verification strategy based on spectral analysis performed on fundamental components. Components are defined to be pieces of writing included between a pen-down movement and the successive pen-up movement (called pen-down singularity as opposed to pen-up singularities). It is claimed that these singularities can occur only in positions which are rather constant in the signatures of an individual. Stability in the positions of singularities allows identification of the finite set of fundamental components of each signer. The existence of a finite set of fundamental components in the signature of an individual makes forgery detection by a component-oriented verification system possible. During enrollment, a knowledge-base for an individual is created containing a component reference table and a structural description graph.

The component reference table presented by Dimauro *et al.* [28] contains the features representative of the classes of fundamental components of a signer. In this work, the structural description graph reported the acceptable sequences of fundamental components in the genuine signatures. For each component, they created a 5-dimensional topological feature vector to describe the component. Then they used a k-means clustering technique in three phases to detect different component clusters. They described these phases as Initial Clusters Recognition, Clusters Growing and Final Clustering. In their study, they found that small variation from these clusters confirm the stability of these topological features in the writing process and their effectiveness for the clustering of the fundamental components. The classification of each component of the reference signature permits the identification of the sequences of fundamental components. Their algorithm creates a graph allowing for the different component sequences as they occur within the training set. According to them, many differences may exist among the components within each cluster such as subtle shape variations or dynamics. To detect such differences among components belonging to the same class, they used a sub-clustering procedure through a particular Fourier descriptors. In this work, they used only the first few Fourier descriptors due to the band-limited nature of the signals produced buy the human writing system. A maximum distance algorithm was then used to split clusters into sub-clusters based on the differences in Fourier descriptors. Verification is done in a two-step fashion. The first step dictates that for a suspect signature to be classified as authentic, its sequence of components must match a possible sequence in the structural description graph of the claimed signer.

If this step is successfully completed, the second step verification is performed where each cluster is verified individually. The Fourier descriptors are used in a distance measure against a threshold value. If any component fails the test, the signature is classified a forgery. The threshold value for each cluster is automatically derived using the worst verification result obtained from the genuine components.

Velocity signals can be derived from positional signals. For the velocity signals v_x and v_y, the autocorrelation functions R_{v_x} and R_{v_y} are calculated. These signals are then regarded as the input and output, respectively, of a finite impulse response (FIR) filter in [88]. The impulse response is obtained by minimizing the least-square error between the autocorrelation signals. A reference vector of impulse responses is calculated from random samples from the training set. The distance between the impulse response of a suspect signature and the reference impulse response is compared to a threshold value to decide on the authenticity.

4.5 Artificial neural networks

Artificial neural networks (ANN) are used today in a wide variety of applications. Some of these include stock market prediction, medical diagnosis, seismic event prediction, speech recognition and artificial vision to name but a few. The ANNs are an active research field and automatic signature verification is no exception. For a gentle introduction to various different neural network architectures, see, [80].

The linear predictor coefficients (LPC) cepstrum is defined as the Fourier representation of the logarithmic amplitude spectrum of a signal. In [143], cepstral coefficients derived from LPCs of the writing trajectories are calculated as the features of signatures. These coefficients are fed into a multi-layer percpeptron (MLP) with multiple input nodes and a single output node. The MLP is selectively trained with backpropagation training meaning the weights are not updated if the desired output is closer than a certain predefined value from the network output. For authentic signatures, the desired output is set to one and for forgeries, it is set to zero. During verification, the LPC cepstrum features of a signature are presented to the trained network and if the output is larger than a threshold value (e.g., 0.5) the signature is accepted as authentic, otherwise it is rejected. A potential problem with this system is the need for negative examples, i.e., forgeries. These would be difficult to obtain for a large scale production system and the use of random forgeries might result in less than optimal performance.

The ANNs can learn from training examples and have the ability to compress information. Compression is an important consideration (see, Bromley *et al.* [20]) because an 80 byte restriction is imposed on the study by the fact that the model needs to be stored on a credit card magnetic strip. A signature is resampled to a fixed number of points by interpolation. Two such resampled signatures are then presented to two subnetworks based on the time delay neural network paradigm. The two subnetworks are joined at the output layer and the objective is to minimize the cosine distance of two feature vectors extracted by the subnetworks. The cosine distance is calculated as

$$\frac{f_1 . f_2}{|f_1||f_2|}.$$

Pairs of input are presented to the network. For pairs of genuine signatures, the desired cosine distance are desired to be 1.0 and for genuine-forge pairs -1.0. Once the network is trained it can be used for verification by presenting training signatures to one of the subnetworks and assuming the output of the network to be a multivariate normal distributed feature. The decision process then becomes a task of inspecting the likelihood value from such a density function.

A time delay neural network (TDNN) is an extension to the basic MPL. Tap-delay lines are added on the input layer to facilitate sequences of data rather than static patterns as is the case for the MLP. A signature is modeled by a TDNN in Schmidt [119]. Feature signals such as velocity, direction and curvature of the pen trajectory are added to the sampled signals. For a specific signer, a TDNN is trained by creating a network with default structure and input window size and applying the error backpropagation learning algorithm. Exemplars are presented in an iterative fashion. Regulated structural changes are imposed and network input window sizes changed according to a specific strategy until the network error ceases to decrease.

A syntactic neural net is a connectionist architecture with the ability to infer grammars from training patterns. A strictly hierarchical context-free grammar is defined in Lucas and Damper [85] to be inferred by such a network. A signature's positional $(x; y)$ information is sampled from a tablet over time. The samples are quantised into an alphabet of eight direction vectors and a null vector for no movement. A non-temporal connectionist parser (NCP) is then used for learning and verification. In theory, the NCP learning and parsing time scale linearly with the pattern length.

Different neural architectures are compared in Lee [74]. A signature is normalized by resampling from a linear interpolation to obtain a sequence of a predefined fixed length. The absolute velocity is used as it is shift, rotation and translation invariant. It is related to $(x(t); y(t))$ as

$$|\upsilon(t)| = \sqrt{\triangle x(t)^2 + \triangle y(t)^2}.$$

Three different neural architectures are tested: time-delay neural network (TDNN), input-output neural network (IONN) and Bayes multilayer perceptron (BMP). These methods appeal to ASV since they act as single systems which automatically extract discriminant features and execute optimal classification in the sense of the Bayes decision rule. Only skilled forgeries are employed in the experiment as it is argued that the real nature of the forgery space is unknown and testing results for random forgeries hardly provides a high degree of reliability and robustness of a ASV system. The performance results reported in the study reveal that it is essential to have forgery training data for NN training. Their results showed that the BMP outperforms the other architectures suggesting it explores global features whereas the other explore local features. The sequence used in this study is fairly long making it difficult for TDNN and IONN to effectively discover discriminating evidence in local features if the dimension of the data is not high enough as is the case here.

Tseng and Huang [134] used ART1 neural network to do signature verification. The pressure pattern sampled from a digitizing tablet is quantized into a binary string of fixed length. A reference pattern is obtained by using the mean pattern for the training set. A vigilance parameter for the ART1 network is derived by inspecting the similarity of the reference pattern to the training patterns. The ART1 network is then trained in the normal sense. Verification is done by presenting a quantized pressure pattern under suspicion to the input nodes and comparing the output to the vigilance parameter to reach a verdict on the authenticity of a signature. The study states that the intended use is for a first stage screening only in a verification system.

This scheme is applied to Chinese signature verification where there is generally more pen-up/down transitions than in other languages. which makes this approach viable. As can be seen from the mentioned studies, a common problem is the need for negative examples (meaning forgeries) when training neural networks. ANNs function by positioning decision surfaces between classes of data rather than positioning model parameters on the data as is the case with for instance HMMs. This problem can be bridged by applying random affine transformations to authentic signatures within an acceptable threshold to fabricate forgeries. It remains to be explored though how effective this approach will be compared to using real forgeries.

4.6 Support vector machines

Support vectors is a technique developed by Vapnik [137] which tries to solve some of the problems that are inherent to all machine learning approaches, namely the problem of sparse data and generalization to unseen data.

The first problem usually occurs in real world applications when the data is of high dimension, many features. What happens is that there usually isn't enough data to represent the classification task at hand very well. This is because all machine learning applications act on empirical data which with all certainty has been sampled badly, i.e., not from an even distribution of the problem space. This makes the training set skewed and the performance of the system is downgraded accordingly.

Another problem is that if the data is sparse then how can the system even be expected to generalize well? In support vector machines (SVM) these problems are solved, somewhat, at the same time. Instead of trying to use all the data for construction of a hard border for choosing which class a new data sample is to be classified for, as is the case in artificial neural networks, the SVM pick the data which represents the task at hand the best, these data samples are called the support vectors, and use these to construct the decision border. The advantage of this approach is that the system will not be as sensitive to outliers that will distort the position and orientation of the border. The system constructed thus might not be the best, i.e., best at classifying unseen new data, that might exist but it is the best that can be achieved given data at hand and it constructs an optimal decision border where optimal is defined as there are no misclassifications, or the least amount if no misclassification is impossible, and that the support vectors closest to the hyper plane are maximal, i.e., the shortest distance to it.

In support vector machine's framework, one of the most widely used test is the Neyman-Pearson test. In [136] many physical situations it is difficult to assign realistic costs or a priori probabilities. A simple procedure to bypass this difficulty is to work with the conditional probabilities P_F and P_D. In general, we should like to make

P_F, as small as possible and P_D as large as possible. For most problems of practical importance these are conflicting objectives. An obvious criterion is to constrain one of the probabilities and maximize (or minimize) the other. A specific statement of this criterion is the following Neyman-Pearson criterion:
Constrain

$$P_F = \alpha' \leq \alpha$$

and design a test to maximize P_D (or minimize P_M) under this constraint. The solution is obtained easily by using Lagrange multipliers. We construct the function F,

$$F = P_M + \lambda[P_F - \alpha'] \tag{4.6.1}$$

or equivalently,

$$F = \int_{z_0} P_{r|H_1}(R|H_1)dR + \lambda\left[\int_{z_1} P_{r|H_0}(R|H_0)dR - \alpha'\right] \tag{4.6.2}$$

Clearly, if $P_F = \alpha'$, then minimizing F would minimize P_M. or

$$F = \lambda(1 - \alpha') + \int_{z_0} P_{r|H_1}(R|H_1) - \lambda P_{r|H_0}(R|H_0)dR. \tag{4.6.3}$$

Now observe that for any positive value of λ an LRT will minimize F. Note that a negative value of λ gives an LRT with the inequalities reversed. This follows directly, because to minimize F we assign a point R to Z_0. only when the term in the bracket is negative. This is equivalent to the test

$$\frac{P_{r|H_1}(R|H_1)}{Pr|H_0(R|H_0)} \leq \lambda, \ \text{ assign point to } \ z_0 \ \text{ or, say } \ H_0. \tag{4.6.4}$$

The quantity on the left is just the likelihood ratio. Thus F is minimized by the likelihood ratio test. To satisfy the constraint we choose λ so that $P_F = \alpha'$. If we denote the density of $\wedge$ when H_0 is true as $P_\wedge|H_0$ $(\bigwedge|H_0)$, then we require

$$P_F = \int_\lambda^\infty P_{\wedge|H_0}\left(\bigwedge|H_0\right) d\bigwedge = \alpha'. \tag{4.6.5}$$

To obtain the threshold value, we solve equation (4.6.5). The value of λ that satisfies equation (4.6.5) will be non-negative because $P_\wedge|H_0$ $(\bigwedge|H_0)$ is zero for negative values of h. Observe that decreasing h is equivalent to increasing Z_1, the region where we say H_1. Thus P_D, increases as α decreases. Therefore we decrease α until we obtain the largest possible $\alpha' < \alpha$. In most cases of interest to us P_F is a continuous function of h and we have $P_F = \alpha$. We shall assume this continuity in all subsequent discussions. Under this assumption the Neyman Pearson criterion leads to a likelihood ratio test.

Support vector machines are similar in architecture to multilayered artificial neural networks, both consists of an input layer, a hidden layer and an output layer. They both use the hidden layer to lift the input into a higher layer feature space so as to facilitate finding a decision border, a hyper plane in higher order dimensions, higher than 2. But the way that ANNs and SVMs find these are completely different as seen in the figure 4.6.1 below.

The reason for the difference of placement of the decision borders between the ANN and the SVM is in that the criterion for error minimization in the ANN is focused on minimizing the misclassification of data in the training set, empirical risk minimization (ERM) whereas the SVM will try to do this based on expected new data, structural risk minimization (SRM). The ERM in theory is good as it is based on the training set and will try to minimize the error of misclassification on this data set but in practice it only leads to over fitting this data and loss in generalization. If the training set was infinitely large, and rich enough, then the law of large numbers would lead to the convergence of the ERM with the actual risk.

The SRM on the other hand tries to set an upper bound on the expected risk; it analyses the data to try to find the structure of it, finding and choosing the support vectors. This risk is maybe not better than ERM if there is enough data, but it will

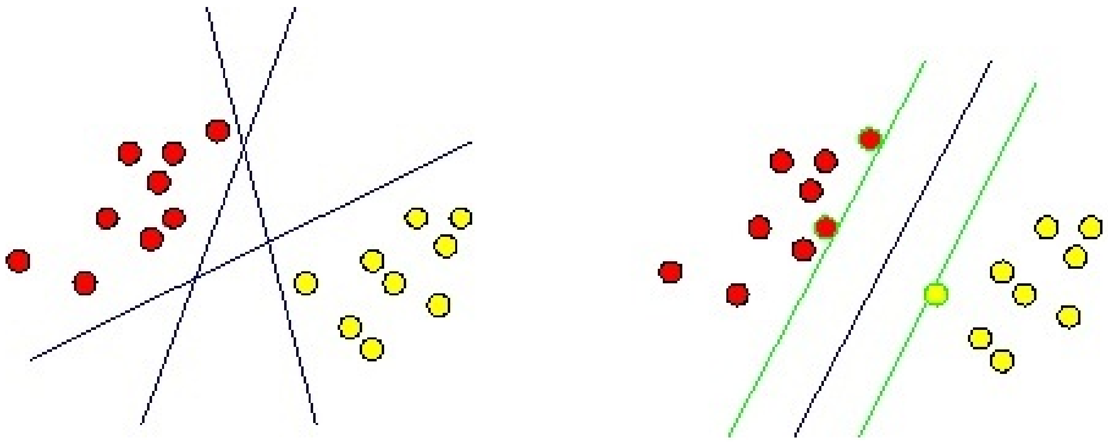

Figure 4.6.1: ANN decision borders vs SVM optimal decision borders

if the data is sparse. In general this approach will lead to better generalization. To better understand how this risk minimization scheme works one can imagine finding the data points that are closest to each other from the different classes, drawing two parallel lines between them and then imagine to push these lines out against the data points so that the distance between the line will be the greatest. The next step is to find the decision border and which is easy once the two other lines orientation and positions have been established. It is only a matter of finding a line parallel to the other two lines that is exactly in the middle of them to find the decision border.

Support Vector Machine Kernels

Now suppose that it's not possible to find a way of putting the first two lines in between the two classes, this is normal in real world data. One thing that can be done, and is done in SVMs, is to lift the data into a higher dimensional feature space, with the help of a transfer function, than the input space. This will hopefully make it easier to separate the two different classes. This is regularly done in artificial neural networks in the hidden layer, and also in SVMs, but due to the more complex risk minimization scheme in an SVM this has to be done with help of kernels.

A kernel is a function that takes a pair of transfer functions as its arguments.

$$K = (x, x') = [\phi(x), \phi(x')].$$

The inner product in feature space is an equivalent kernel in input space. There are however certain conditions that kernels function have to fulfil to be able to be used in a support vector machine. The kernel has to be symmetric positive definite and it also has to satisfy Mercer's conditions ([137]):

$$\int\int K(x, x')g(x)g(x')dxdx' > 0, \qquad g \epsilon L_2,$$

where

$$K = (x, x') = \sum_{m}^{\infty} a_m \phi_m(x)\phi_m(x'), \;\; a_m \geq 0.$$

In the above, ϕ is a mapping from the set of observations to the feature space.

Linear and Polynomial Kernel

The simplest kernel, the one described in figure 4.6.1, is the linear kernel.

$$K(x, y) = x \bullet y, \;\; where\, x \;\text{ and }\; y \;\text{ are input sequences.}$$

This kernel assumes that the time series are similar if they have been generated by the same autoregressive model where an autoregressive model is a model to predict the future of a time series. How does this relate to the linear kernel? The decision function of a linear kernel takes the form

$$f(x) = w \bullet x + b,$$

and when the decision function is used to predict time series

$$x_T = f(x_{T-1}, \ldots, x_{T-k}) = \sum_{t-1}^{k} w_{T-t} + b.$$

The result is an statistical autoregressive model of order k_9. A simple extension to the non-linearly separable case is to expand the linear kernel to a polynomial kernel.

4.7 Summary

In this chapter, we explained how Dynamic time wrapping, hidden Markov model, fourier tranformation and artifical neural network are used for signature modelling. We are also discussed the Neyman Pearson and support vector machines. In the next chapter, we are going to discuss concepts of Markov chains and hidden Markov model which were used in the implementation.

Chapter 5

Signature modelling using HMMs

A hidden Markov model is a statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters; the challenge is to determine the hidden parameters from the observable data. In this chapter we will explain the Markov chains and hiddin Markov models. Then it comes to the explanation of duration modelling and its possible extensions. Finally, concludes with the brief summary of the chapter.

5.1 Introduction

The theory of hidden Markov models (HMMs) was first introduced in a series of papers by Baum and colleagues in the late 1960's [6, 7, 8, 9, 10] at Institute for Defense Analysis (IDA). In the 1970's, Baker at carnegie-mellon univeristy, jelinek at IBM, and other applied HMMs to the problem of speech recognition. The success of these systems dramatically increased interest in applying HMMs to continous speech recognition and other difficult pattern recognition problems such as signature verification. As the work described in this thesis uses hidden Markov models extensively, we have included a chapter on the topic. It should be noted however that this chapter will not attempt to replace the excellent introductory work on HMMs found in papers such as the seminar tutorial by Rabiner [110].

HMMs are powerful tool for modelling time series data. They are used in speech recognition systems, biometrics, computational biology, and in the other areas of patter recognition and artificial intelligence. In present, HMMs have also been used in computer vision applications such as gesture recognition, image sequence modelling and object tracking. Hidden Markov models extend statistical models known as Markov chains. Markov chains arise naturally in biology, psychology, economics and many other sciences. There are two types of HMMs classified by their observation densities: discete-HMMs and continuous HMMs. We shall proceed with an overview of Markov chains and then extend the concept to hidden Markov models.

The rest of the chapter is organized as follows. We detailed about Markov chains and hidden Markov models in Section 5.2. Then explained about the duration modelling in section 5.3. Then we discussed about some other possible extentions in hiddne Markov models in section 5.4. Finally, we give a brief summary in Section 5.5.

5.2 Markov chains and HMMs

Markov chains

Real-world processes generally produce observable outputs which can be characterized as signals of either discrete (e.g., weather classified into sunny, cloudy and rainy) or continuous nature (e.g., features extracted from speech signals). The non-deterministic of the weather state is an example of a system which may be expressed (i.e., modelled) by a Markov chain. Such a model can then be used to predict the likelihood of a certain state some time in the future.

More precisely, Markov chains are used to model phenomena exhibiting a sequence (i.e., chain) of fixed length periods during which any one of a set of N distinct states, $S = (S_1, S_2, \dots, S_N)$ can be assumed. Transitions between states occur over time and are expressed by probabilistic means. A matrix of transition probabilities links states by giving the probability of being in a state for the next time period, given the current state of the system

$$a_{ij} = P(q_t = s_j | q_{t-1} = s_i), \quad 1 \leq i, j \leq N.$$

The entries a_{ij} need not necessarily be non-zero for all i and j; if they are, then the model is said to be a fully connected model or ergodic model. The assumption that the state at a certain time is dependent only on the previous state, is called the first order Markov assumption. This need not necessarily be the case. If n is the length of the state history in the choice of the next state, the model is said to be an n-th order Markov model but, as is most often the case, $n = 1$ implying first order Markov models. The Markov assumption simplifies matters significantly, however, for many complex processes the first order assumption may lead to a less than accurate expression by the model. Nevertheless, since such simplified systems may often be more readily subjected to analysis, we bring ourselves to live with the shortages, baring in mind the possible inaccuracy of the results. We may perhaps compensate for them in other ways through domain specific knowledge in order to tap from the sound formalism of Markov models and especially hidden Markov models. Recently, it has been shown how efficient higher order hidden Markov models can be realized [32]. Figure 5.2.1 depicts all possible first order transitions between our chosen weather states. In a fully connected (ergodic) model configuration with N distinct states, there are N^2 transition probabilities which, as previously stated, can be collected into a state transition matrix A. For the weather example with

$$S = \{S_1 = \text{sunny}, \;\; S_2 = \text{cloudy}, \;\; S_3 = \text{rainy}\},$$

the transition probability matrix becomes

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}.$$

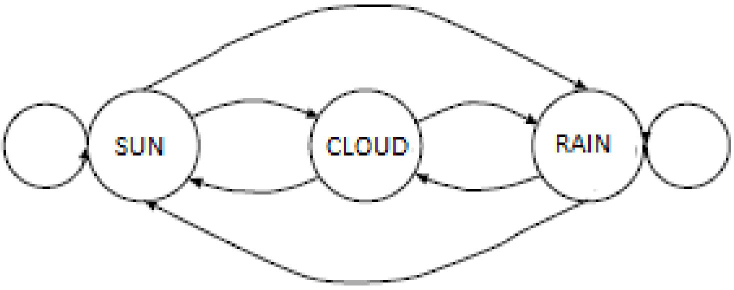

Figure 5.2.1: A case of first order HMM (Weather example)

Entry a_{ij} , with i indicating the row and j indicating the column, is the probability of making the transition from state i to state j. Thus, a_{22} is the probability of the weather remaining cloudy given that it was cloudy for the previous time period. Because a a_{ij} is interpreted as a probability, the row entries must adhere to stochastic constraints and

$$a_{ij} \geq 0, \quad 1 \leq i, j \leq N,$$

with

$$\sum_{j=1}^{n} a_{ij} = 1, \quad 1 \leq i \leq N.$$

The probabilities remain stationary over time which often proves to be an unrealistic assumption.

There is still one missing part of information in defining the weather Markov model; which is the vector

$$\pi = (\pi_1,\ \pi_2,\ \pi_3),$$

that denotes what the probable state of the weather was at time t_1.

We have now fully defined a first order Markov model

$$M = (S, \pi, A),$$

with

- S : states,
- π : starting state probabilities, and
- A : transition probabilities.

Such a system is called a Markov process.

We could now ask questions regarding the model for instance: What is the probability that the observation sequence

$$O = \{S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3\},$$

was generated by the model M? That is, what would the model say is the chance of having the weather start out as rainy and then be rainy-rainy-sunny-sunny-rainy-cloudy-rainy. This can be expressed as

$$\begin{aligned} P(O|M) &= P(S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3|M) \\ &= P(S_3).P(S_3|S_3).P(S_3|S_3).P(S_1|S_3).P(S_1|S_1).P(S_3|S_1).P(S_2|S_3).P(S_3|S_2) \\ &= \pi_3.a_{33}.a_{33}.a_{31}.a_{11}.a_{32}.a_{23}. \end{aligned}$$

Another question which leads to the notion of a probability density function is: Given the model is in a known state, what is the probability that it would stay in that state for exactly d days? The observation sequence would be

$$O = \{S_i^1, S_i^2, S_i^3, ..., S_i^d, S_{j \neq i}^{d+1}\},$$

where the superscripts merely indicate the time instance. We express the probability as

$$P(O|M, q_1 = S_i) = 1.(a_{ii})^{d-1}(1 - a_{ii}) = p_i(d),$$

with the 1 referring to the certainty of our knowledge about what the state at t_1 is and $(1 - a_{ii})$ referring to the mandatory transition out of state i. We call this quantity $p_i(d)$ the discrete probability density function of duration d in state i. Using $p_i(d)$ we can now calculate the expected duration of remaining in state i, denoted by $\bar{d}_i$, given the system started in state i by

$$\begin{aligned}
\bar{d}_i &= \sum_{d=1}^{\infty} d p_i(d) \\
&= \sum_{d=1}^{\infty} d(a_{ii})^{d-1}(1 - a_{ii}) \\
&= (1 - a_{ii}) \sum_{d=1}^{\infty} \left[(d-1)(a_{ii})^d - 1 + (a_{ii})^d - 1\right] \\
&= (1 - a_{ii}) \left(\frac{a_{ii}}{(1 - a_{ii})^2} + \frac{1}{1 - a_{ii}} \right) \\
&= \frac{a_{ii}}{1 - a_{ii}} + 1 \\
&= \frac{a_{ii} + 1 - a_{ii}}{1 - a_{ii}} \\
&= \frac{1}{1 - a_{ii}},
\end{aligned}$$

using a well known infinite series identity

$$\sum_{n=1}^{\infty} n x^n = \frac{x}{(1 - x)^2}.$$

The above model is called an observable Markov model since the output of the process is a set of observations at each instance of time where each model state corresponds to an observable event. There is however instances where this modeling technique is too limited to model the process under inspection. This leads us to the notion of a hidden Markov model.

Hidden Markov models

When listening to a voice, the sound one hears is the product of state changes in the vocal and nasal tracts, respectively. Variables include the size of the throat, position of the tongue and radiation effects at the lips. We thus have an observable speech signal and a hidden vocal system which are non-trivialy related.

A hidden Markov model (HMM) extends the concept of a Markov chain by attaching an observation probability distribution to each state in the model. Within the framework of the theory, this effectively hides the states previously visible in Markov chains as each state has an 'opinion' about any observation value. We now have the ability to attach arbitrary semantics to model states which might not be readily accessible through observation in the process being modelled. This means that we can better model processes where we cannot directly observe the process states but instead have access to the outputs resulting from the internal state changes in the process.

Model states do not necessarily have to correspond to some physical quantization of process states. Instead, they can be any abstraction with sensible semantics within the context of the process being modelled and the observations sampled from such a process. It is up to the modeller to decide on the semantics of a model. Sensible semantics will assist in initializing the model to startup values which will converge faster during discovery of the model parameters. The theory provides for a means to infer model parameters from a training set of observation sequences in a way which maximizes the likelihood of the sequences being generated by the model. Furthermore, we can calculate the likelihood that a sequence was generated by a model and derive the most probable state sequence corresponding to an observation sequence.

As stated, a hidden Markov model augments a Markov chain by coupling observation symbol distributions to the model states. As with Markov chains, we distinguish between

- discrete models where the process observations assumes one of a finite set of possible values (this calls for a probability distribution of the observations at each state),

- continuous models where the process observations are of continuous nature. This calls for a probability density function for the observations at each state.

A HMM is thus defined as

- S : hidden states,
- π : starting state probabilities,
- A : transition probabilities i.e., $a_{ij} = P(q_t = S_j | q_{t-1} = S_i)$,
- B : observation symbol of probability distributions, i.e., $P(o|S_i)$ or probability density, and functions $p_s(o)$ where o is either a discrete or continuous variable.

The complete parameter set of the model is indicated by the compact notation $\lambda = (S, \pi, A, B)$.

To extend the weather example we might imagine a scenario where the weather state is not observable any more. Instead we have access to readings from a barometer which measures atmospheric pressure. We can quantify such a reading into low, medium and high pressure to obtain discrete pressure values. We denote such a set of discrete observed values as $V = (v_1, v_2...., v_M)$ with $M = |V|$, *i.e.*, $V_{\text{weather}} = (\text{low, medium, high})$.

Figure 5.2.2 shows a graphical representation of the Markov model extended to a HMM with the previously visible weather states now hidden and the barometer readings observable. The figure illustrates how probability distributions and density functions for discrete and continuous models, respectively, are coupled to states. The connection from a hidden state to an observable value represents the likelihood of generating the observable value, given that the model is in the hidden state. Thus, the likelihood of rain at a given time depends both on the weather state at the previous time instance and the barometer reading for the current time instance. The observation probability

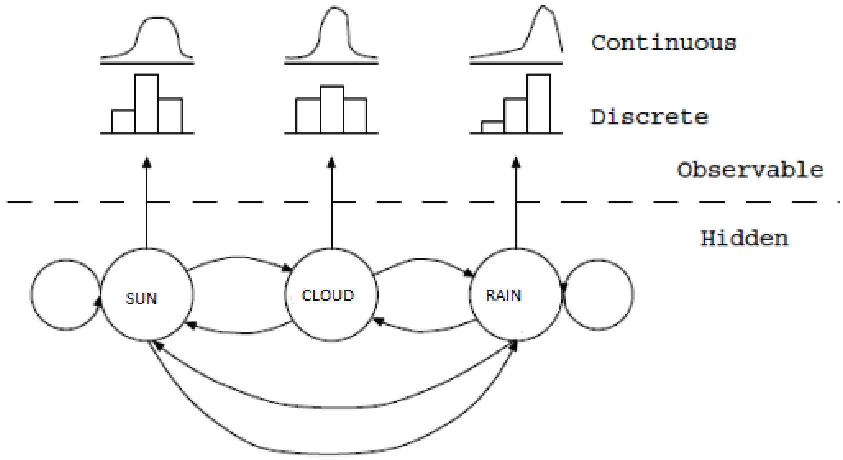

Figure 5.2.2: A case of higher order HMM (Weather example)

distribution in state j can be expressed as

$$b_j(o_t) = P(o_t|q_t = S_j); \quad 1 \leq j \leq N, \quad 1 \leq k \leq M.$$

In the discrete case, probabilities can be arranged in matrix form called a confusion matrix with a row assigned to each hidden state S_i and a column to each observable value o_t. In general, we will refer to the probability of generating symbol o_t when in state S_i as $b_i(o_t)$.

The observable event for a particular hidden state is a stochastic variable and therefore the entries in any row of matrix B satisfy the condition

$$\sum_{j=1}^{M} b_{ij} = 1, \quad 1 \leq i \leq N.$$

Many signals are continuous in nature and although a continuous signal can be quantized by one of the many vector quantization techniques available, information is likely to be lost in the process which could affect modelling performance to some extent. It is therefore advantageous to have HMMs with continuous observation density functions such as Gaussian mixtures. In this case, we then have the condition

$$\int_{-\infty}^{\infty} b_j(x)dx = 1, \;\; 1 \leq j \leq N.$$

When working with continuous HMMs, an assumption about the form of the density function has to be made. In some studies, see, e.g., Kashi *et al.* ([65]), prefer to discretize continuous values to avoid making this assumption. Another approach is to employ neural networks to learn the form of the density functions [43].

This weather model can be used to answer questions regarding various weather related issues. For instance, we want to use the model in such a way as to determine what the most probable season is in which a certain string of barometer measurements were made. This brings us to three problems associated with HMMs as generally itemized in the literature:

Problem 1 (Observation sequence likelihood): Given an observation sequence $O = o_1 o_2 .....o_T$ and a HMM λ, how do we efficiently compute $P(O|\lambda)$, i.e., the probability of the observation sequence being generated by the model?

Problem 2 (Most Probable State Sequence (Viterbi Algorithm)): Given an observation sequence $O = o_1....o_T$ and a model λ, what state sequence $Q = q_1......q_T$ best explains the observations? Several optimality criteria exist and the appropriate one to use depends on the problem at hand.

Problem 3 (Baum-Welch Parameter Re-estimation): Finding a suitable set of parameters to model a process. The question now arises what the word suitable suggests. We would like to maximize the probability $P(O|\lambda)$ where O is an observation sequence sampled from the process being modelled and forms part of a set of training sequences to deduce λ from.

Below we discuss each of the above mentioned problems in detail.

Problem 1: We want to compute the likelihood of the observation sequence O of length T being generated by the model λ, i.e., $P(O|\lambda)$.

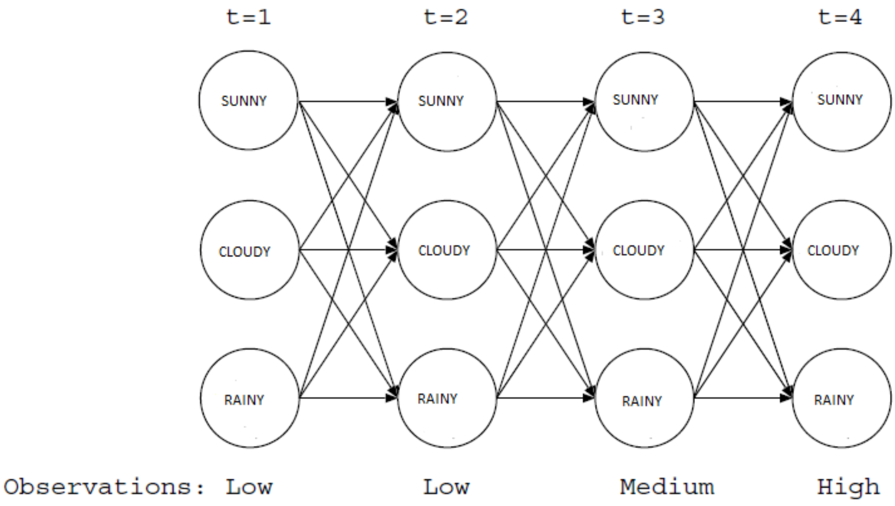

Figure 5.2.3: Forward procedure for the partial observation sequence

Figure 5.2.3 illustrates the observation sequence low low medium high and the possible hidden states at each time instance as a trellis. The most straightforward way of calculating $P(O|\lambda)$ is through enumerating every possible state sequence of length T and summing. Without documenting this process, we conclude by stating that such an approach is computationally unfeasible due to the complexity being in the order of $O(TN^T)$. This led to the development of what is known as the forward procedure. The procedure is made possible by the time invariance of the probabilities in HMMs. We define a variable (called the forward variable) as

$$\alpha_t(i) = P(o_1, o_2, \ldots, o_t;\ q_t = S_i|\lambda),$$

which is the probability of the partial observation sequence $o_1, o_2, \ldots, o_t$ and the model λ being in state S_i at time t. An inductive calculation of $\alpha_t(i)$ is

1. Basis for induction:

$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N.$$

The forward probabilities are initialized to be the joint probability of starting out in state S_i and the first observation symbol being o_1.

2. Induction step:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)a_{ij}\right] b_j(o_t+1); \;\; 1 \leq t \leq T-1, \;\; 1 \leq j \leq N.$$

3. Termination step:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha T(i).$$

The desired probability is given by the sum of the terminal forward variables $\alpha T(i)$. The time complexity of this procedure is in the order of $O(N^{2T})$ which is a huge improvement over $O(TN^T)$.

Above mentioned second step expresses the fact that any hidden state in the vertical columns of the trellis of Figure 5.2.3 can only be reached via the N hidden states in the previous column, i.e., previous time instance. From this, we deduce that to any of the N hidden states at time t, $1 \leq t \leq T$, there exist N^{t-1} distinct state sequences or paths leading to this state from the states at time $t = 1$. This is illustrated in Figure 5.2.4. Due to the time invariance of A and B, we can use induction to calculate a forward variable for a specific state at a specific time rather than trace all the possible sequences the entire way back to time $t = 1$. We therefore calculate $\alpha_{t+1}(j)$ as the sum of the forward variables for the partial observation sequences up to time t over all the hidden states, each time multiplying by the transition probability for being in the hidden state and making a transition to state S_j (which is the terminal state for the value being calculated). This sum is then multiplied by the observation probability for symbol o_{t+1} observed when in state S_j.

Problem 2: We want to find a hidden state sequence which best explains the observation sequence. There is no exact solution to this problem due to the uncertainty of which optimality criterion to use. The most popular approach is called the Viterbi

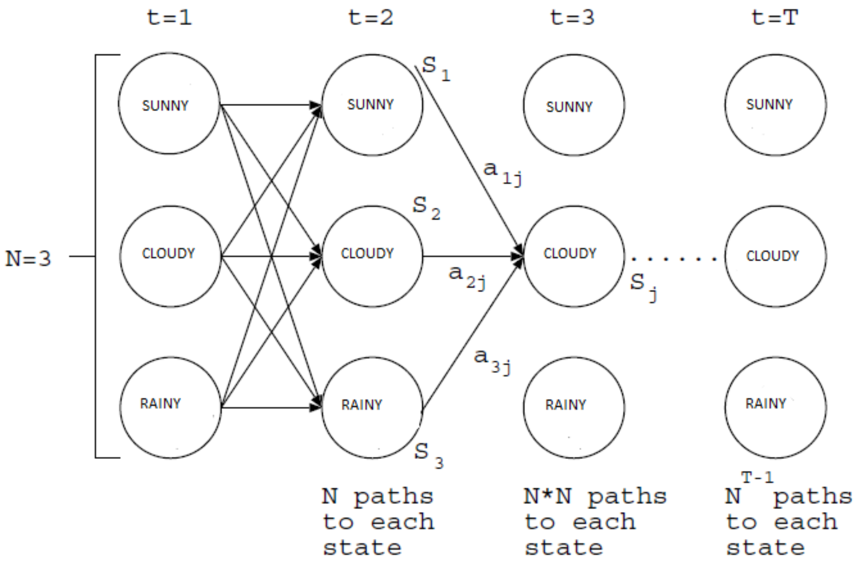

Figure 5.2.4: Most probable state sequence

algorithm which stems from dynamic programming methods. This algorithm seeks to find a single best hidden state sequence through the graph of Figure 5.2.3 every time inspecting $b_i(o_t)$ to make its decisions. To classify a sequence as being best, it needs to be enumerated and score a higher value than all other sequences in the graph. A hidden state sequence Q is enumerated for observation sequence O through

$$\pi q_1 b_{q_1}(o_1) \prod_{t=2}^{T} a_{q_{t-1}q_t} b_{q_t}(o_t).$$

This action is equivalent to maximizing $P(Q|O, \lambda)$. For each intermediate and terminating state in the trellis of Figure 5.2.3 there is a most probable path to that state. So, for example, each of the three states at $t = 4$ will have a most probable path to it, perhaps as in Figure 5.2.5. The Viterbi algorithm, rather than enumerating all possible sequences in a brute force fashion and selecting the maximum scoring one, goes about recursively to find the sequence as follows.

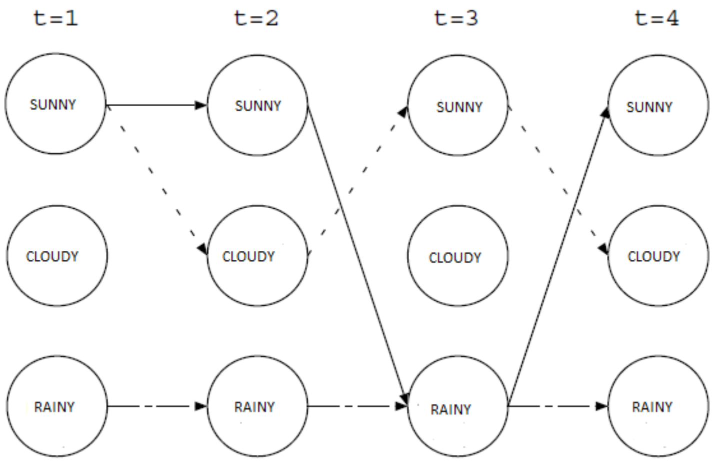

Figure 5.2.5: Possible state sequences with the highest likelihood

We define the quantity

$$\delta_t(i) = \max_{q_1,\ldots,q_t-1} P[q_1,\ldots,q_t = S_i; o_1,\ldots,o_t|\lambda],$$

which is the best scoring sequence of length t ending in state S_i. By induction we have

$$\delta_{t+1}(j) = [\max_i \delta_t(i)a_{ij}]b_j(o_t + 1).$$

This quantity will facilitate in finding the highest score; however, the state chosen to maximize the quantity at each time instance, needs to be remembered if the actual sequence is to be reconstructed after the search terminates. For this purpose, an array $t(j)$ is used. The complete recursive algorithm is then

1. Basis for induction

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N, \tag{5.2.1}$$

$$\psi_1(i) = 0. \tag{5.2.2}$$

2. Inductive step

$$\delta_t(j) = \max_{1\le i\le N}[\delta_{t-1}(i)a_{ij}]b_j(O_t), \;\; 2 \le t \le T, \quad 1 \le j \le N, \tag{5.2.3}$$

$$\psi_t(j) = \arg\max_{1\le i\le N}[\delta_{t-1}(i)a_{ij}], \;\; 2 \le t \le T, \quad 1 \le j \le N. \tag{5.2.4}$$

3. Termination

$$P^* = \max_{1\le i\le N}[\delta_T(i)], \tag{5.2.5}$$

$$q_t* = \arg\max_{1\le i\le N}[\delta_T(i)]. \tag{5.2.6}$$

4. Sequence reconstruction

$$q*_t = \psi_t + 1(q*_{t+1}), \quad t = T_1, \;\; T_2, \ldots, 1. \tag{5.2.7}$$

The above algorithm, known as Viterbi algorithm, provides a computationally efficient way of analyzing observations of HMMs to recapture the most likely underlying state sequence. It exploits recursion to reduce computational load, and uses the context of the entire sequence to make judgements, thereby allowing a good analysis.

Problem 3: We want to adjust the model parameters λ as to maximize the probability $P(O|\lambda)$ for a given O. This is a learning problem and for any finite observation sequence as training data, we can estimate λ so that, at best, $P(O|\lambda)$ will be locally maximized. This section will present a re-estimation procedure for iteratively updating and improving λ, called the Forward-Backward or Baum-Welch algorithm. This re-estimation procedure is based on the principal of maximum likelihood estimation (MLE). With MLE, the parameter(s) describing a likelihood function (i.e., the parameters of the HMM) are discovered by holding fixed the underlying random variable and varying the function parameters in such a way as to maximize the function. The Baum-Welch re-estimation formulae described here converges to a parameter set λ

for an HMM which maximizes the function [10]. Any good mathematical statistics textbook can be consulted for a description of MLE [33].

We define a variable (called the backward variable) as

$$\beta_t(i) = P(o_{t+1}, \ldots, o_T | q_t = S_i, \lambda),$$

which is the probability of the partial observation sequence from $t+1$ to the end, given the model λ is in state S_i at time t. As with the forward variable, we can solve for $\beta_t(i)$ inductively as follow

1. Basis for induction

$$\beta_T(i) = 1, \quad 1 \leq i \leq N.$$

2. Inductive step

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \quad t = T_1, \ldots, 1, \; 1 \leq i \leq N.$$

Furthermore, we define

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda),$$

which is the probability of being in state S_i at time t and state S_j at time $t+1$ given the model λ and observation sequence O. We can write $\xi_t(i,j)$ in terms of the forward variable introduced during the solution to problem 1 and the backward variable as follow

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}.$$

The denominator serves to normalize the term as to make it a probability measure.

Figure 5.2.6 graphically illustrates the calculation with $a_{ij}b_j(o_{t+1})$ providing the link between the partial sequence probability up to time t ending in state S_i and the partial sequence probability from time $t+1$ onwards starting in state S_j. Having obtained $\xi_t(i,j)$, we define

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j),$$

which is the probability of being in state S_i at time t, given O and λ. Summing $\gamma_t(i)$ over time gives the expected number of times that state S_i will be visited or when time $t = T$ is excluded, the expected number of transitions from state S_i

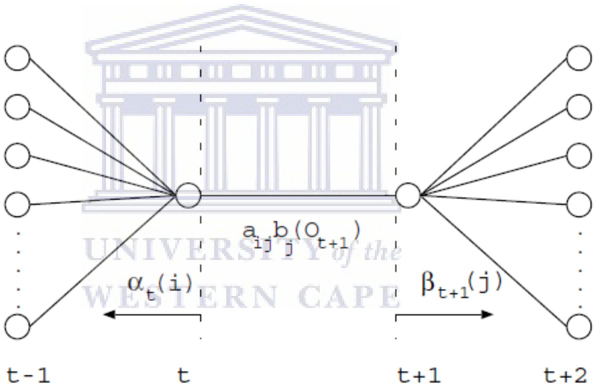

Figure 5.2.6: Forward-Backward procedure to find the likelihood

$$\Gamma(i) = \sum_{t=1}^{T-1} \gamma_t(i).$$

Likewise, summing $\xi_t(i,j)$ over time gives the expected number of transitions from state S_i to S_j:

$$\Xi(i,j) = \sum_{t=1}^{T-1} \xi_t(i,j).$$

Using the defined quantities, the Baum-Welch algorithm re-estimates the HMM parameters as follows

$$\begin{aligned}
\overline{\pi} &= \gamma(i), \\
\overline{a}_{ij} &= \frac{\Xi(i,j)}{\Gamma(i)}, \\
\overline{b}_i(k) &= \frac{\sum_{t=1}^{T} \xi_t(i,j) \text{ and } o_t = \upsilon_k \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}.
\end{aligned}$$

If λ did not already define a critical point of the likelihood function, in which case the re-estimation will have no effect, the new parameter set $\overline{\lambda}$ obtained from the procedure, describes a model more likely to have produced the observation sequence O. The procedure preserves stochastic constraints for the HMM parameters, viz.,

$$\begin{aligned}
\sum_{i=1}^{N} \overline{\pi}_i &= 1, \\
\sum_{j=1}^{N} \overline{a}_{ij} &= 1, \;\; 1 \le i \le N, \\
\sum_{k=1}^{M} \overline{b}_j(k) &= 1, \;\; 1 \le j \le N.
\end{aligned}$$

Unfortunately, some difficulty arise when implementing HMMs in finite computing systems. The calculations involved when working with HMMs, often multiplies probabilities which are by definition less than 1. When the length of a sequence is large enough, the computed values generally decrease beyond the precision range of most computing machines. A procedure does exist which scales calculated values to fall within a computable range and results in probabilities in the log domain. This procedure can be found in [110] and will not be presented here. Instead, we adopt another training procedure based on the Viterbi algorithm which provides an easier way to overcome this problem. The algorithm has the added advantage that it enables faster training than the Baum-Welch re-estimation procedure.

Viterbi training procedure

The Viterbi algorithm described previously in the above, finds the most probable state transition sequence a process undergoes whilst generating a particular observation sequence. The forward and Viterbi algorithms together with a hidden Markov model configuration both define probability density functions over a space of observation sequences. The algorithm can thus calculate the likelihood that an observation sequence was generated by a model. The likelihood calculated by the Viterbi algorithm differs from that calculated by the forward algorithm as explained in the above section. This suggests that the overall shape of the probability density function defined by the Viterbi procedure differs slightly from that defined by the forward procedure. Thus, the Viterbi training algorithm maximizes the likelihood of the training set for a different density function than does the Baum-Welch procedure. Experience has shown, however, that the results obtained by using this approach do not significantly differ from those obtained by using the Baum-Welch algorithm; yet, they require far fewer computations.

Before explaining the training algorithm, we show how the computations of the Viterbi algorithm are modified to overcome the problem of values exceeding the precision range of computers [65, 110]. The idea is to perform calculations in the log domain. The Viterbi algorithm is modified by changing Equation (5.2.1) to

$$\delta_1(i) = \log(\pi_i) + \log(b_i(O_1)), \;\; 1 \leq i \leq N,$$

Equation (5.2.3) becomes

$$\delta_t(j) = \max_{1\leq i\leq N} \left[\delta_{t-1}(i) + \log(a_{ij})\right] + \log(b_j(O_1)), \;\; 2 \leq t \leq T, \;\; 1 \leq j \leq N.$$

This results in the calculation of a log-likelihood with Equation (5.2.5) becoming

$$\log(P^*) = \max_{1\leq i\leq N} \left[\delta_T(i)\right].$$

Once a model configuration has been decided on, the model parameters need to be discovered from a training set of sequences. Prior knowledge about the problem domain and the semantics of the model states assist in initialization of the model parameters before training commences. This often proves to be crucial in achieving a good representation of the modelled process. It is fairly common for transition variables to be initialized to random values maintaining stochastic constraints. However, applying this approach to set initial values of the probability distributions/density functions is most likely to result in far less than optimal parameter discovery. In Chapter 6, various initialization schemes are employed prior to training models. Initialization can be seen as biasing the training procedure and the challenge thus resides in finding a good model bias for the problem at hand. A description of the Viterbi training algorithm is now presented.

The Viterbi algorithm is conducted in a batch fashion meaning that all the training sequences are used during a single re-estimation iteration. From the training sequences O_K, we re-estimate λ by

- the starting probability for state S_i as

$$\overline{\pi}_i = \frac{\sum q_1^k = S_i}{K}, \quad 1 \leq i \leq N,$$

i.e., the number of times a computed state sequence starts in state S_i as a fraction of the number of training patterns

- the transition probability for state S_i to S_j as

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} q_t^k = S_i \text{ and } q_{t+1}^k = S_j}{\sum_{t=1}^{T-1} q_t^k = S_i}, \quad 1 \leq i, j \leq N,$$

i.e., the number of times a transition is made from state S_i to S_j as a fraction of the number of times state S_i was visited the observation symbol probability (in the discrete case) for state S_i and observation symbol v_j as

$$\bar{b}_{ij} = \frac{\sum_t q_t^k = s_i o_t = v_j}{\sum_{t=1}^{T-1} q_t^k = S_i}, \;\; 1 \leq i \leq N, \;\; 1 \leq j \leq M,$$

i.e., the total number of times the model was in state S_i and generated observation symbol v_j as a ratio of the number of times S_i was visited. In the continuous case the observation value v_j contributes to an average observation value for state S_i and a second traversal of the sequences is needed to determine the standard deviation from this average. These values are then used to define a probability density function.

During training, we have the option of imposing a restriction on the allowed terminal state used by the alignments. This means we can say that an alignment may not terminate further to the left from the rightmost state (in the case of left-right models that is) than a preset distance. Our experience have shown that this approach results in models with a better ability to distinguish between true and false exemplars. The Viterbi training algorithm is also described briefly in [65].

As can be seen, the Viterbi training algorithm is straight forward and contains less calculations than the Baum-Welch algorithm. It should be noted that this algorithm is based on the assumption that a most probable state sequence can be matched to an observation sequence. This means that prior to training the model needs to be initialized in such a way that from the outset, training sequences are close enough to model state observation distributions to prevent underflow due to uncomputable likelihood values. This stresses the importance of a good biasing initialization prior to the re-estimation training procedure.

State transition configurations

As explained in the above Section, a model is considered ergodic when any state in the model can be reached from any other state in a single transition. This however need not always be the case. In a *left-right* (Bakis) HMM [65, 110, 156] states are numbered in ascending order; a system either remains in the same state, i.e., a self-transition, or it transitions to a state with a higher index. In such a model, the initial state probabilities, Π , has the property that only the left-most state has a non-zero starting probability, i.e., $\pi_0 = 1$ and $\{\pi_1, \ldots, s\} = 0$. This type of model has been found to account for the observed properties of certain types of signals better than does the ergodic model. In a variant of this model, parallel paths through the model are also allowed.

Figure 5.2.7 shows a graphical depiction of these types of models. Conceptually, any configuration is possible and will have no in on the re-estimation procedure. State transitions set to zero when re-estimation commences, will remain zero. The converse is not true however, meaning non-zero transitions could very well become zero as the re-estimation procedure iterates.

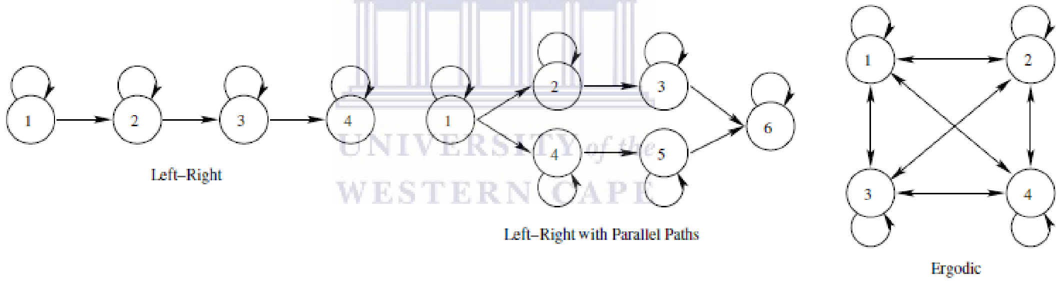

Figure 5.2.7: Types of hidden Markov models

Figure 5.2.7 helps us to understand the conceptual difference between left-right and ergodic models. With Left-right models, the observation density functions for states at certain offsets from the left of the state graph, correspond closely to actual sequence values at similar offsets from the beginning of the sequence. A left-right model thus act as a sequence memory allowing only for marginal deviations from a representative sequence both in sequence values (vertical deviations) and timing information (horizontal deviations). We will thus use left-right models when we want to model a signal source which produces fairly similar sequences, i.e., stationary processes. Ergodic models, on the other hand, allow for self-similarities within the signal as any state in the model can be reached at any stage. This allows for better generalization; however, the set of sequences which will match the model is not as intuitively predictable as is the case with left-right models. Ergodic models can thus match unseen sequences disparate from those in the training set.

5.3 Duration modelling

As described earlier in the section 5.2, the state duration probability density function is

$$P(O|M, q_1 = S_i) = 1.(a_{ii})^{d-1}(1 - a_{ii}) = p_i(d),$$

based on the value of the self-transition probability a ii of a state. This density function decays exponentially and is not suitable for many physical signals. An explicit way of modelling state duration is presented in [110]. This, however, results in a quadratic increase in computational cost. An alternative heuristic which alters the computed log-likelihood in a postprocessing phase according to duration probability histograms derived from a segmental k-means procedure can be used instead.

Duration modelling is achieved, Kashi *et al.* [65] by introducing a number of unit-duration sub-states replacing each original model state. The transitions from one state to the sub-states of the next state approximates the wanted transition probability density function. This approach has the disadvantage that it significantly increases the number of model parameters. Figure 5.3.1 shows an example of such a substitution.

we explore a simple way of limiting the maximum number of self-transitions in a state by restricting the Viterbi algorithm to allow only a specified number of self-transitions in each separate state. This is realized by maintaining a duration count for each state and updating this parameter along with the other model parameters during re-estimation. With each re-estimation iteration, the duration count for a given state is decremented by one if none of the training sequences result in a Viterbi alignment which remains in that state for exactly

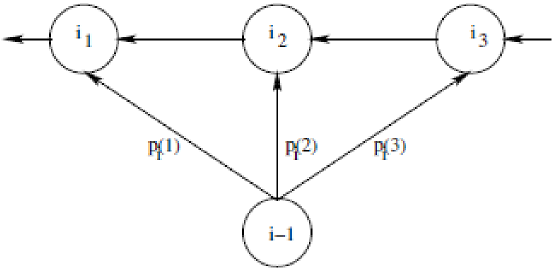

Figure 5.3.1: Duration modelling sub-states

the allowed count. If, however, there is a state sequence which remains in the state for the maximum allowed count, the allowed duration count is incremented by one. This does not achieve the same effect as the duration modelling described in the previous paragraphs. It does however enable a model to disallow sequences which scores high due to the entire sequence consisting of values close to the mean of a single state's observation density function.

With regards to our application of HMMs that we will be presenting in chapter 6, the possibility to restrict large deviations from the training sequences used to build the models, is an important factor. The ability to limit the number of times a state may be repeated in a highest probability state sequence, prevents an alignment with sequences which match the observed values in certain states for excessive periods of time. Together with the restriction on the allowed terminal state, this form of duration restriction forces a sequence to more or less conform to the profile of the entire length of training sequences. It will prohibit sequences which deviate from such a model to score high likelihoods when the alignment is calculated.

5.4 Possible extensions

A number of other extensions to HMMs have also been proposed particularly as far as recognition of complex gestures are concerned [15, 142]. Many hybrid combinations of HMMs and Artificial Neural Networks (ANNs) have also been proposed in the field of automatic speech recognition [12, 13, 19, 63, 122, 135, 161]. It is shown how ANNs can be used to learn the observation symbol distributions in states, relieving us from having to make assumptions about the shape of the distributions. Furthermore, ANNs are used to perform input transformations on the data before presenting it to a HMM. Gradient descent learning rules adapted from have also been applied to HMMs in an attempt to create a unifying learning method to better integrate HMMs and ANNs.

5.5 Summary

In this chapter, we have discussed important notions of signature modelling using Hidden Markov models. These HMMs are statistical modelling tools for time series modelling. They have been applied to various fields of research with a great deal of success. In this chapter therefore, we provided the necessary background on hidden Markov models to understand the next chapter which applies HMMs to automatic signature verification. In the next chapter, we will discuss about the implementation of hidden Markov model for online signature verification. We will also present different experiments to evaluate the performance of our proposed modelling approach.

Chapter 6

Data training and experimental results

In this chapter, different experiments on online signature verification with hidden Markov model have been carried out. The objectives of experiments are to investigate the reliability of online signature verification system with hidden Markov model and find the out the relationship and dependence between different extracted features and to find out the best combination of features to represent a signature are discussed.

6.1 Introduction

Automatic signature verification has been studied in the last tow decades. Nowadays, the performance of this approach is being challenged, as skillful forgers can imitate the shape of a signature easily. For possibilities of improvements, automatic signature verification has being studied. Automatic signature verification includes both static and dynamic features for verification. Static features refer to the information that cab be extracted from the signature shape such as point coordinates, angles between pint pairs, width and height. Dynamic features refer to the information that describes the signing process such as pressure change, speed, pen-tilt stroke-order and stroke-direction. Dynamic features are difficult to imitate, because they are hidden in the sense that they cannot be revealed by simply observing the shape of the signature. Also different experiments are performed using dynamic features can improve the accuracy of verification.

To measure the accuracy of a verification systems are usually used two types of errors. False acceptance rate describes the error rate that forged patterns are accepted incorrectly, while false rejection rate describes the error rate that genuine pattern are rejected incorrectly. There is no fixed relationship of importance between FAR and FRR. A highly secure system prefer a low FAR and a high FRR, while a user-friendly system prefers a high FAR and a low FRR. In order to compare the accuracy between different verification systems. The term equal error rate, for which FAR equal to FRR, is usually used to measure the overall accuracy of a verification system.

A Wacom Intous graphical tablet is used to capture signature through out the experiment. This digitized tablet is built with orthogonal sensors, which collect the current x and y position of the pen tip on the surface of the tablet, pressure, x and y-coordinate and time are captured through this digitized tablet. The objectives of the experiments are to accept as many genuine signatures as possible and reject as many as forged signatures as possible.

The rest of the chapter is organized as follows. We discuss about the signature processing and representation in Section 6.2. Explanation about the signature model training is given in section 6.3. Then we discussed about experimental results in section 6.4. Finally, we give a brief summary in Section 6.5.

6.2 Processing and representation

Handwritten signature can be considered as “fast handwriting”, and it can be represented by a sequence of point movements. As a result, the hidden Markov model approach, which is powerful for recognizing time-varying patterns, can be applied to the problem of online signature verification.

For online signature verification, different features like pen pressure, pen tilt, stroke direction, etc, are extracted along every point in a signature. The extracted features form a feature sequence that can be modeled with a hidden Markov model and can perform stochastic matching with any test signature.

In this chapter different experiments on online signature verification with the hidden Markov model have been carried out. The objectives of experiments are to

1. investigate the reliability of online signature verification with hidden Markov model,

2. to find out the relationship and dependence between different extracted features, and

3. to find out the best combination of features to represent a signature.

The objectives of the experiment are to accept as many genuine signatures as possible and reject as many forged signatures as possible.

Signature database

We obtained permission from J.G.A. Dolfing, the author of amongst others [30], to use a database compiled at his institution for our research for which we are greatly appreciative. This database contains 1530 authentic signatures for 51 individuals of who 45 are males and 6 females. Each individual donated 30 signatures which are divided into two sets of equal size for training and validation purposes. Each stored signature comprises of a sequence of 5-tuples (1) pen x, (2) pen y, (3) pressure, (4) x tilt and (5)y tilt sampled from the PAID tablet described in Section 2.3.1. Three types of forgeries are included in the database (1) home-improved, (2) over-the-shoulder and (3) professional. The home-improved and over-the-shoulder forgeries were created by the 51 individuals which contributed the authentic signatures. Each data acquisition session involved both a signer and a forger. While the signer contributed signatures, the forger closely watched the dynamics of the signing process. The forger then attempted to recreate the dynamics of the observed signature to create the over-the-shoulder forgery. The roles were then reversed to produce another set of signatures and forgeries. Each individual was provided with a paper copy of another signature to take home for practice. A set of home-improved forgeries were subsequently donated by those individuals. These forgeries can be regarded as amateur and the final home-improved set consists of 1530 forgeries whereas the over-the-shoulder set contains 1470 forgeries. Additionally, four forensic document examiners provided a total of 270 forgeries of 20 individuals from the database. The signatures of these individuals were evenly divided over complexity classes easy, moderately easy and difficult to forge.

Preprocessing

As described in chapter 3, signature captured through a digitized tablet are usually interfered by signing environment such as the weight of pen provided for signing as well as the orientation of the tablet. As a results, the captured signatures should be preprocessed in order to minimize such interferes. In this experiment are mapped to an area of 100*100 pixels square. Additionally, dynamic features such as the signing pressure and pen-tilt are normalized to a range from 0 to 100. Details preprocessing can be found in the Table 6.2.1.

Table 6.2.1: Details of Preprocessing

Dimension	Feature	Original range	Range
1	x-coordinate	0 to 12700	0 to 100
1	y-coordinate	0 to 9700	0 to 100
1	Pressure	0 to 1024	0 to 100
1	x pen-tilt	0 to 1710	0 to 100
1	y pen-tilt	0 to 840	0 to 100

Feature extraction

Feature extraction methods for online signature are similar to the methods used for handwriting recognition. Both static and dynamic features are extracted from a signature. Static features include coordinate,delta angle, while dynamic features include velocity acceleration, pressure, pen-tilt, time, stroke directions and air movement. Extracted features form a 16 dimensional feature vector, and the whole signature is represented by a sequence of these feature vectors.

Table 6.2.2: Details of Feature Vector

Dimension	Feature	Range
2	x y coordinate	0 to 100
2	$\sin\theta, \cos\theta$	-1 to 1
2	$\Delta\sin\theta, \Delta\cos\theta$	-1 to 1
1	Pressure	0 to 100
1	Δ pressure	No range
2	Δ x,Δ y pen-tilt	No range
1	x,y pen-tilt	0 to 100
1	θ	0 to 100
1	$\Delta\theta$	No range
1	Velocity	No range
1	x,y Acceleration	No range
1	Time	No range

First select the each feature from the dimensional feature vector and different combinations of features are also selected from the same dimensional feature vector for modelling. Select the features with the highest representation power is evaluated in this experiment. Additionally, as the signing process is easily interfered by the signing environment, we are interested in investigating whether using standardized features can improve the verification accuracy. Standardized features are the features values standardized by their corresponding mean and variance, i.e.,

$$f_i^{\wedge} = \frac{f_i - \mu_i}{\sigma_i},$$

where f_i is the i^{th} component in the feature vector, μ_i and σ_i are the mean and variance of the i^{th} feature component for the signature respectively.

Model representation

As we discussed in chapter 5, the left-right hidden Markov model is used for signature modelling throughout the experiment. shows an example of left-right hidden Markov mode:

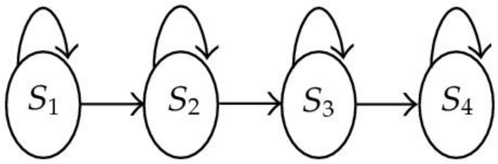

Figure 6.2.1: Example of left-right hidden Markov model

Similar to other types of model, a let-right hidden Markov model consists of a set of N states, a set of M observable symbols, a state-transistion probability matrix $A = a_{ij}$, a set of observation symbol probability distributions, and a set of initial state probabilitiesΠ. But a left-right hidden Markov model only allows states sequence start at state 1(and end at state N), i.e.,:

$$\pi_i = \begin{cases} 0, & i \neq 1, \\ 1, & i = 1, \end{cases}$$

is the initial state probability for state i and a left-right hidden Markov model only allows left-right state-transitions, i.e.,

$$a_{ij} = 0, \quad \text{for} \quad j < i,$$

often, additional constraints that prevent large changes in state indices usually places on the state-transition coefficient, i.e;

$$a_{ij} = 0, \quad \text{for} \quad j > i + \Delta i.$$

Left-right hidden Markov model is usually applied to speech recognition. It is because this type of model contains desirable properties for modelling time-varying sequences. In addition, discrete probability density function is used for states probabilities density function. Discrete probability density function is more practical than continues probability density function like Gaussian mixtures for model training because less training data is required for state probability density function training.

6.3 Model training

In the training process the codebook is used to map vectors from continues spaces to discrete number of predefined symbols. For codebook training, an efficient Vector Quantizer (VQ) can be constructed using a top-down approach K-means algorithm (binary split approach). Hight dimensional feature vectors are subdivided in to different streams to reduce dimension, and each stream is clustered in to M groups by the K-means algorithm. One subdivided stream results in one single codebook, and multiple codebook is a collection of different single codebooks. Table 6.3.1 show the codebook side for different features:

Table 6.3.1: Codebook sizes of different features

Feature	Codebook size
x y coordinate	32
$\sin\theta, \cos\theta$	32
$\Delta\sin\theta, \Delta\cos\theta$	16
Pressure	4
Δ pressure	8
Δ x,Δ y pen-tilt	16
x,y pen-tilt	8
θ	32
$\Delta\theta$	32
Velocity	32
x,y Acceleration	16
Time	32

In the second step, feature vectors are encoded with the codebook constructed in the first phase. The encoded feature vectors are used for discrete hidden Markov model training. For model training, t-uniform state segmentation method is used to initialize model stage setting. After, EM algorithm is applied in for model training. In the final step, classifier for the constructed model is being constructed. Two approached are concerned for classifier training; the first approach constructs a classifier based on Gaussian distribution of average log-likelihood obtained from genuine signatures, while the second approach constructs a classifier based on both genuine and forged signatures with Neyman-Pearson classification.

In the first approach, average log-likelihood of a testing signature is obtained by dividing the log-likelihood with the length of the testing signature sequence, i.e.,

$$p^* = -\frac{\log p}{L},$$

where $\log p$ refers to the log-likelihood computed by the reference model, and L is the length of the signature.

The decision boundary of the classifier is defined as

$$\overline{p} = \alpha.\sigma,$$

where $\overline{p}$ is the expected value for p^* and σ is its deviation scaled by a constant factor α.It should be noted that $\overline{p}$ and σ are person-specific values obtained from each individual's HMM.

In the second approach as different forged samples are available for model training,we can define decision boundary by applying neyman pearson classification on both genuine and forged samples.

6.4 Experimental results

We are conducted many experiments on online signature using hidden Markov model with combination of Neyman Pearson and support vector machine.

In the first experiment investigates the various features using each feature with separately in a single stream hidden Markov model. The second experiment investigates the representation power of different combinations of features and compares the representation power between standardized features and un-standardized features vectors. And in the final experiment studies the trained features with the combination of neyman pearson and support vector machine.

For signature verification, several error rates are of importance. The False Acceptance Rate indicates the percentage of forged signatures that have been accepted. The False Rejection Rate determines the percentage of valid signatures which have been rejected. If one evaluates the number of correctly accepted original signatures and the number of correctly rejected forgeries, this results are in the total agreement which is given in all the following experiment. The information content of the various features have been investigated using each feature separately in a single stream hidden Markov model. This results are in the accuracy which is given in Table 6.4.1.

Table 6.4.1: Verification results for each feature used in single stream HMMs

Feature	Accuracy
x y coordinate	65%
$\sin\theta, \cos\theta$	60%
$\Delta\sin\theta, \Delta\cos\theta$	55%
Pressure	80%
Δ pressure	70%
Δ x,Δ y pen-tilt	66%
x,y pen-tilt	69%
θ	55%
$\Delta\theta$	45%
Velocity	81%
x,y Acceleration	58%

Modelling with different combinations of features

Various combinations of features are investigated in this experiment. Tables 6.4.2 shows the selections of different features: The error rates of different feature combinations with different decision boundary widths. Different decision boundaries are obtained by

Table 6.4.2: Different combinations of features

Combinations	Accuracy
All	61%
3	70%
5	75%
7	65%

changing the value of α in the

$$p^* = \overline{p} + \alpha.\sigma,$$

function where p^* represents the decision boundary, $\overline{p}$ is expected values for P, and σ is the deviations of P scaled by a constant factor α.

The feature combination using 5 is the best, for all decision boundary widths from $\alpha = 1$ (1 standard deviation) to $\alpha = 6$ (6 standard deviations). The second best feature combination was 3 features. The third feature combination was 7. The experiment results show that the decision boundary width that minimizes the error for using all features s around 5 standard deviations, with equal-error rate 2%. The decision boundary width for using coordinate, pressure, velocity, pen-tilt is 2.7 standard deviations, with equal-error rate 3%. The decision boundary width for using coordinate, pressure, velocity, path tangent and time is 4.8 standard deviations, with equal-error rate of 2% experiments results show that the combination of all features and the combination of coordinate, pressure velocity and time features can obtains lower error rate. This implies that the two feature combinations have the highest representation power over the other combination features for online signatures.

Modelling with un-standardized and standardized features

The other experiment examines the representations ability of standardized features and un-standardized features. Un-standardized features are the exact values of features, while standardized features are those features values standardized by the feature's mean and variance.

We noticed that using un-standardized features could achieve better results than using standardized features. The observed standardized features could reduce the false rejection rate. However, it also significantly increased the false acceptance rate of the system. As a result, using standardized features reduced the overall performance of the system.

Model training with forged samples

This section covers the design and implementation of the signature feature discrimination experiments that were conducted to determine the effect that varying amounts of significant features has on the performance level of the network. The features used in the various features sets are as follows: for 3 features; x, y co-ordinates and pentip pressure, for 5 features; x, y co-ordinates, pen-tip pressure and the two pen-tip angles θ_x and θ_y, for 7 features; x, y co-ordinates, pen-tip pressure, the two pen-tip angles θ_x and θ_y, pen-tip velocity $V(t)$ and path-tangent T_θ. Three different experiments were conducted for the three signature feature sets, i.e., 3, 5 and 7 features. The data set consisted of 15 users of varying signature complexities. Within the 3 feature sets the x and y co-ordinates represented the static features while the remaining features were dynamic in nature. Also of note is the fact that within the 7 feature set both velocity and path-tangent are derived features where velocity is computed using the x and y co-ordinates and path-tangent is then calculated based on the velocity.

Table 6.4.3: Performance summary of signature feature sets

Forgery Type	FAR	FRR
Casual	70%	35%
Skilled	65%	25%
Forensic	60%	26%

Classification with SVM:

Using the second approach, we construct a classifier based on both genuine and forged signatures using Neyman-Pearson and Support vector machines.

As we have discussed in chapter 4, Neyman-Pearson algorithm efficiently computes the optimal operating points on the receiver operating characteristic (ROC) curves, i.e., for a given selection of features and a chosen acceptable false alarm rate (i.e., the probability of rejecting a genuine signature), the algorithm computes the optimally achievable probability of detecting a fraudulent signature. This gives the user control over the system performance in a rigorous fashion. This allows us to rank features using support vector machines.

We use support vector machines to classify the final acceptance and rejection states.Training were performed 10 fold cross validation on the complete dataset with 51 genuine and 27 forged signatures. The SVM algorithm was used to classify the signatures. The kernel function of the SVM was the Linear and Polynomial Kernel.For testing we used the test signature image dataset with 51 genuine and 25 forged signatures.

Procedure followed in testing:

(a) Test dataset was fed into the SVM model. At the beginning, all of the features were used. Recognition Rate, False Rejection Rate and False Acceptance Rate were observed.

(b) Then number of features tested with the SVM model was decreased step by step according to the weightage of features. Features with least weightage were gradually removed every time and Recognition Rate, FRR and FAR were observed.

(c) Above step (b) was continued till we reached a single feature.

Observations: Maximum recognition rate of 70% with FRR = 5% and FAR = 1% was achieved when we used 3 features (acceleration, velocity, pressure). When number features used in testing was increased beyond four, recognition rate reduced.

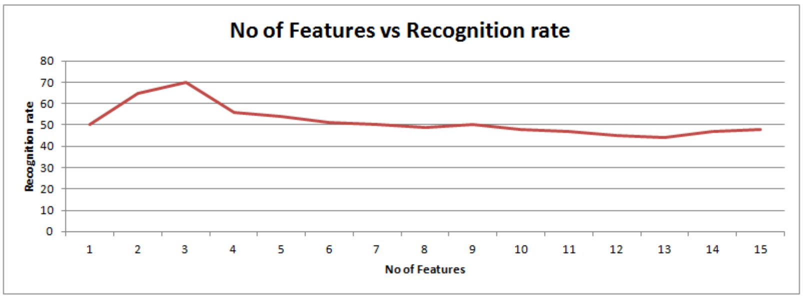

Figure 6.4.1: Number of features vs recognition rate

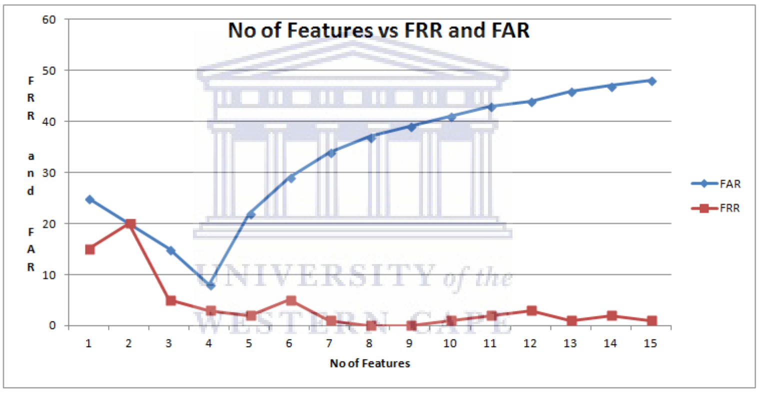

Figure 6.4.2: Number of features vs FRR and FAR

In our approach, we tried to find the effectiveness of some commonly used global features in online signature verification. It is observed that out of those fifteen features, four features were not selected for any of the datasets. Another five features were also found to be less significant for classification as they were selected for maximum 3 datasets. Out of the remaining six features, three top ranked features could give a maximum recognition rate of 70%. Signature verification is a very sensitive problem. Cost of misclassification associated with signature verification is very high.

6.5 Summary

In this chapter, we have examined the problem using hidden Markov model on Automatic signature verification. Different experiments have been performed to evaluate the performance of modelling approaches. By the experimental results it is concluded that the best combination of 5 features set when compare with the other feature sets. Results also show that using un-standardized features can achieve better results than using standardized features. Finally, results show that using different forged samples for model training can improve the accuracy of the constructed signature model.

Chapter 7

Concluding remarks and scope for future research

In this thesis, we have studied different aspects of online signature verification. A digitized tablet with orthogonal sensors used for collecting signature samples. It can collect axial pen tilt and pen pressure in x and y-directions with 64 level with maximum signal report rate 200Hz. We have collected about 1530 authentic signatures for 51 individuals were collected in which 45 were males and 6 were females used for this experiment. From this experiment we noticed that we trained individually trained each feature and as well trained in the group. We developed a authentication system that

1. scales easily as new features are added,
2. allows to built separate, possibly different HMMs for the different features, and
3. optimally combines features to compute the likelihood that a given signatures is genuine from the likelihoods for the individual features.

We used left-right HMMs for modeling the individual dynamic features act as they allow only for marginal deviations from a representative sequence both in sequence values (vertical deviations) and timing information (horizontal information). We then applied the Neyman-Pearson algorithm to efficiently compute the optimal operating points on the receiver operating characteristic curves, i.e., for a given selection of features and a chosen acceptable false alarm rate (i.e., the probability of rejecting a genuine signature), the algorithm computes the optimally achievable probability of detecting a fraudulent signature. This gives the user control over the system performance in a rigorous fashion. It also allows us to rank features according to their power of discrimination.

As far as the **scope for future research** is concerned, we would like to mention the following:

- Proposed method (which is based on the combination of simple HMMs and with Neyman Pearson) can be extended for other applications.
- Extraction of known and perhaps novel measures of complexity from signatures.
- Training of hybrid RNN-HMM to predict the threshold value for HMM likelihood measures to accept or reject.

Bibliography

[1] T. Ahmad, J. Hu and S. Wang, Pair-polar coordinate-based cancelable fingerprint templates, *Pattern Recognition* **44(10-11)** (2011) 2555-2564.

[2] J.L.M. Amaral, A.J. Lopes, J.M. Jansen, A.C.D. Faria and P.L. Melo, An improved method of early diagnosis of smoking-induced respiratory changes using machine learning algorithms, *Computer Methods and Programs in Biomedicine* **112(3)** (2013) 441-454.

[3] R.M.L. Baena, D. Elizondo, E.L. Rubio, E.J. Palomo and T. Watson, Assessment of geometric features for individual identification and verification in biometric hand systems *Expert Systems with Applications* **40(9)** (2013) 3580-3594.

[4] G. Bailador, C. Avila, J. Guerra-Casanova and A. Santos, Sierra Analysis of pattern recognition techniques for in-air signature biometrics, *Pattern Recognition* **44(10-11)** (2011) 2468-2478.

[5] R. Baron and R. Plamondon, Acceleration measurement with an instrumented pen for signature verification and handwriting analysis, *IEEE Transactions on Instrumentation and Measurement* **38(6)** (1989) 1132-1138.

[6] L. Baum, An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes, *Inequalities* **3** (1972) 1-8.

[7] L. Baum and J. Egon, An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology, *Bulletin of Amererican Mathematical Society* **73(3)** (1967) 360-363.

[8] L. Baum and T. Petrie, Statistical inference for probabilistic functions of finite state Markov chains, *Annals of Mathematical Statistics* **37** (1966) 1554-1563.

[9] L. Baum, T. Petrie, G. Soules and N. Weiss, A maximization technique occuring in thestatistical analysis of probabilistic functions of Markov chains, *Annals of Mathematical Statistics* **41(1)** (1970) 164-171.

[10] L. Baum and G. Sell, Growth functions for transformations on manifolds, *Pasific Journal of Mathematics* **27(2)** (1968) 211-227.

[11] L. Batista, E. Granger and R. Sabourin, Dynamic selection of generativediscriminative ensembles for off-line signature verification, *Pattern Recognition* **45(4)** (2012) 1326-1340.

[12] Y. Bengio and P. Frasconi, Input output hmms for sequence processing, *IEEE Trans-actions on Neural Networks* **7(5)** (1996) 1231-1249.

[13] H. Bourlard and N. Morgan, Hybrid HMM/ANN Systems for Speech Recognition, *Overview and New Research Directions* Springer- Verlag 1997.

[14] K.W. Bowyer, The results of the NICE.II Iris biometrics competition , *Pattern Recognition Letters* **33(8)** (2012) 965-969.

[15] M. Brand, N. Oliver and A. Pentland, Coupled hidden Markov models for complexaction recognition, *MIT Media Lab Perceptual Computing Learning and Common Sense Technical Report 407*, 1996.

[16] J. Brault and R. Plamondon, A complexity measure of handwritten curves: Modelling of dynamic signature forgery, *IEEE Transactions on Systems, Man and Cybernetics* (1993) 400-413.

[17] J. Brault and R. Plamondon, How to detect problematic signers for automatic sig-nature verification,*Proceedings. 1989 International Carnahan Conference on Security Technology* (1989) 127-132.

[18] J. Breebaart, I. Buhan, K. Groot and E. Kelkboom, Evaluation of a template protection approach to integrate fingerprint biometrics in a PIN-based payment infrastructure, *Electronic Commerce Research and Applications* **10(6)** (2011) 605-614.

[19] J.S. Bridle, Alpha-nets: A recurrent 'neural' network architecture with a hidden Markov model interpretation, *Speech Communication* **9(1)** (1990) 8392.

[20] J. Bromley, J. Bentz, L. Bottou, I.G.Y. Lecun, C. Moore, E. Sackinger and R. Shah, Signature verification using a siamese time delay neural network, *International Journal of Pattern Recognition and Artificial Intelligence* **7(4)** (1993) 669-688.

[21] E.L.V. Broek, Beyond biometrics, *Procedia Computer Science* **1(1)** (2010) 2511-2519.

[22] F.M.V. Castaldi and E.S. Gomez, A new spontaneous pupillary oscillation-based verification system, *Expert Systems with Applications* **40(13)** (2013) 5352-5362.

[23] S. Chakraborty, V. Balasubramanian and S. Panchanathan, Generalized batch mode active learning for face-based biometric recognition, *Pattern Recognition* **46(2)** (2013) 497-508.

[24] M.C Chuang and M.C. Chen, An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics *Expert Systems with Applications* **41(4)** (2014) 14111418.

[25] H. Crane and J. Ostrem, Automatic signature verification using a three-axis force sensitive pen, *IEEE Transactions on Systems, Man and Cybernetics* **13(3)** (1994) 749-770.

[26] H. Crane, D. Wolf and J. Ostrem, The SRI pen system for automatic signature verification, in proceedings of the *NBS Trends and Applications* **32** 1977.

[27] A.H. Cummings, M.S. Nixon and J.N. Carter, The image ray transform for structural feature detection, *Pattern Recognition Letters* **32(15)** (2011) 2053-2060.

[28] G. Dimauro, S. Impedovo and G. Pirlo, Component-oriented algorithms for signature verification, *International Journal of Pattern Recognition and Artificial Intelligence*, **8(3)** (1994) 771-793.

[29] A.D. Dinkar and S.S. Sambyal, Person identification in Ethnic Indian Goans using ear biometrics and neural networks, *Forensic Science International* **223(1)** (2012) 373.e1-373.e13

[30] J.G.A. Dolfing, E.H.L. Aarts and J.J.G.M. van Oosterhout, On-line signature verification with hidden Markov models, In: Proceedings of the 14th International Conference, *Fourteenth International Conference on Pattern Recognition* **2** (1998) 1309-1312.

[31] D.T. Toledano, R.F. Pozo, A.H. Trapote and L.H. Gomez, Usability evaluation of multi-modal biometric verification systems, *Interacting with Computers* **18(5)** (2006) 1101-1122.

[32] J.A. du Preez, Effcient high-order hidden Markov modelling, *PhD thesis, University of Stellenbosch, Stellenbosch, South Africa* 1998.

[33] E. Dudewicz and S. Mishra, *Modern Mathematical Statistics* (1988) 347-367.

[34] E.P. Eernisse, C.E. Land and J.B. Snelling, Piezoelectric sensor pen for dynamic signature verification, *Proceedings of the IEEE International Electronic Devices Meeting* **30** 1977.

[35] M. Fairhurst and P. Brittan, An evaluation of parallel strategies for feature vector con- struction in automatic signature verification systems, *International Journal of Pattern Recognition and Artificial Intelligence* **8(3)** (1994) 661-678.

[36] M.C. Fairhurst, New perspectives in automatic signature verification, *Information Security Technical Report* **3(1)** (1998) 52-59.

[37] S.-C. Fang and H.-L. Chan, QRS detection-free electrocardiogram biometrics in the reconstructed phase space, *Pattern Recognition Letters* **34(5)** (2013) 595-602.

[38] C. Feher, Y. Elovici, R. Moskovitch, L. Rokach and A. Schclar, User identity verification via mouse dynamics, *Information Sciences: an International Journal* **201** (2012) 19-36.

[39] A.J. Fierrez, S. Krawczyk, G.J. Ortega and A.K. Jain, Fusion of local and regional approaches for on-line signature verification, *Proceedings of the International Workshop on Biometric Recognition Systems, IWBRS, Springer LNCS-3781*, Beijing, China, (2005), pp 188-196.

[40] A.J. Fierrez, L. Nanni, L. Penalb, J.O. Garcia and D. Maltoni, An on-line signature verification system based on fusion of local and globalinformation, *Proceedings of the 5th International Conference on Audio and Video-based Biometric Person Authentication*, AVBPA, Springer LNCS-3546, New York, USA, (2005), pp 523-532.

[41] L. de Figueredo, Adaptive sampling of parametric curves, *Graphics Gems V* **5** (1995) 173-178.

[42] M. Fons, F. Fons and E. Canto, Biometrics-based consumer applications driven by reconfigurable hardware architectures, *Future Generation Computer Systems* **28(1)** (2012) 268-286.

[43] H. Franco, M. Cohen, N. Morgan, D. Rumelhart and V. Abrash, Context-dependent connectionist probability estimation in a hybrid hidden Markov model

neural net speech recognition system, *Computer Speech and Language* **8(3)** (1994).

[44] J. Galbally, J. Fierrez, J. Ortega and R. Plamondon, Synthetic online signature generation. Part II: Experimental validation, *Pattern Recognition* **45(7)** (2012) 2622-2632.

[45] J. Gil and D. Keren, New approaches to the arc length parameterization problem: In *Proceedings of the 13th Spring Conference on Computer Graphics* (1997) 27-34.

[46] R. Giot and C. Rosenberger, Genetic programming for multibiometrics, *Expert Systems with Applications: An International Journal* **39(2)** (2012) 1837-1847.

[47] T.A.F. Gomes, R.B.C. Prudencio, C. Soares, A.L.D. Rossi and A. Carval, Combining meta-learning and search techniques to select parameters for support vector machines, *Neurocomputing* **75(1)** (2012) 3-13.

[48] M.E. Gonalves and M.I. Gameiro, Security, privacy and freedom and the EU legal and policy framework for biometrics, *Computer Law and Security Review* **28(3)** (2012) 320-327.

[49] J. Gupta and A. McCabe, *A review of dynamic handwritten signature verification*, Technical Report **9**, James Cook University, 1997.

[50] G. Gupta and R. Joyce, A study of some pen motion features in dynamic handwritten signature verification, Unpublished Report, Department of Computer Science, James Cook University, Townsville, Australia, 1997.

[51] W. Haberman and A. Fejfar, Automatic identification of personnel through speaker and signature verification system description and testing, *Proceedings of the Carnahan Conference on Crime Countermeasures* (1976) 20-30.

[52] M. Hanmandlu, J. Grover, A. Gureja and H.M. Gupta, Score level fusion of multimodal biometrics using triangular norms, *Pattern Recognition Letters* **32(14)** (2011) 1843-1850.

[53] T. Hastie and E. Kishon, A model for signature verification, Conference Proceedings 1991, *IEEE International Conference on Systems, Man, and Cybernetics* **1** (1992) 191-196.

[54] B. Herbst and D. Richards, On an automated signature verification system, *IEEE International Symposium on Industrial Electronics* **2** (1998) 600-604.

[55] N. Herbst and C. Liu, Automatic signature verification based on accelerometry, *IBM Journal Research and Development* **21(3)** (1977) 245-253.

[56] G. Hermosilla , D.S.J. Ruiz , R. Verschae and M. Corre, A comparative study of thermal face recognition methods in unconstrained environments, *Pattern Recognition* **45(7)** (2012) 2445-2459.

[57] D.J. Hurley, M.S. Nixon and J.N. Carter, Force field feature extraction for ear biometrics, *Computer Vision and Image Understandin* **98(3)** (2005) 491-512.

[58] Y. Imamverdiyev, A.B.J. Teoh and K. Jaihie, Biometric cryptosystem based on discretized fingerprint texture descriptors, *Expert Systems with Applications* **40(5)** (2013) 1888-1901.

[59] S.M.S. Islam, R. Davies, M. Bennamoun, R.A. Otheyns and A.S. Mian, Multibiometric human recognition using 3D ear and face feature Pattern Recognition, *Pattern Recognition* **46(3)** (2013) 613-627.

[60] Y. Iwashita, A. Stoica and R. Kurazume, Gait identification using shadow biometrics, *Pattern Recognition Letters* **33(16)** (2012) 2148-2155.

[61] Z. Jin, A.B.J. Teoh, T.S. Ong and C. Tee, Fingerprint template protection with minutiae-based bit-string for security and privacy preserving, *Expert Systems with Applications* **39(6)** (2012) 6157-6167.

[62] Y. Ji and S. Sun, Multiclass support vector machines: Model and experiments, *Pattern Recognition* **46(3)** (2013) 914-924.

[63] F.T. Johansen and M.H. Johnsen, Non-linear input transformations for discriminative hmms, *ICASSP* (1994).

[64] H. Kameya, S. Mori and R. Oka, A segmentation-free biometric writer verification method based on continuous dynamic programming, *Pattern Recognition Letters* **27(6)** (2006) 567-577.

[65] R. Kashi, J. Hu, W. Nelson and W. Turin, On-line handwritten signature verification using hidden Markov model features, *Proceedings of the Fourth International Conference on Document Analysis and Recognition* **1** (1997) 253-257.

[66] R. Kashi, W. Turin and W. Nelson, On-line handwritten signature verification using stroke direction coding, *Optical Engineering* **(9)** (1996) 2526-2533 .

[67] H. Ketabdar, J. Richiardi and A. Drygajlo, Global Feature Selection for On-line Signature Verification, *Proceedings of the 12th Conference of the International Graphonomics Society*, Solerno, Italy, 2005.

[68] T. Kohonen, The self-organizing map, *Proceedings of the IEEE* **78(9)** (1990) 1464-1480.

[69] A. Krogh, M. Brown, I. Mian, K. Sjolander and D. Haussler, Hidden Markov models in computational biology. Applications to protein modeling, *Journal of Molecular Biology* **235(5)** (1994) 1501-1531.

[70] J. Kruppa, A. Schwarz, G. Arminger and A. Ziegler, Consumer credit risk: Individual probability estimates using machine learning, *Expert Systems with Applications* **40(13)** (2013) 5125-5131.

[71] H. Lai, J. Xiao, L. Li and Y. Yang, Recursive hiding of biometrics-based secret sharing scheme using adversary structure, *Information Processing Letters* **112(17-18)** (2012) 683-687.

[72] F. LeClerc and R. Plamondon, Automatic signature verification: The state of the art 1989-1993, *International Journal of Pattern Recognition and Artificial Intelligence* **8(3)** (1994) 643-660.

[73] J. Lee, H. Yoon, J. Soh, B.T. Chun and Y.K. Chung, Using geometric extrema for segment-to-segment characteristics comparison in online signature verification, *Pattern Recognition* **37(1)** (2004) 93-103.

[74] L. Lee, Neural approaches for human signature verification, *3rd International Conference on Signal Processing Proceedings* **2** (1996) 1346-1349.

[75] L. Lee, *Online Systems for Human Signature Verification* Ph.D. Thesis, Cornell University, 1992.

[76] L. Lee, T. Berger and E. Aviczer, Reliable on-line human signature verification, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18(6)** (1996) 643-647.

[77] Y. Lei, M. Bennamoun, M. Hayat and Y. Guo, An efficient 3D face recognition approach using local geometrical signatures, *Pattern Recognition* **47(2)** (2014) 509-524.

[78] X. Li, J.W. Niu, J. Ma, W.D. Wang and C.L. Liu, Cryptanalysis and improvement biometrics-based remote user authentication scheme using smart cards, *Journal of Network and Computer Applications* **34(1)** (2011) 73-79.

[79] X. Li, Y. Xiong, J. Ma and W. Wang, An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards, *Journal of Network and Computer Applications* **35(2)** (2012) 763-769.

[80] R. Lippman, An introduction to neural networks, *Optical Engineering* **35(9)** (1996) 2526-2533.

[81] Y. Liu, The principle of proportionality in biometrics:Case studies from Norway, *Computer Law and Security Review* **25(3)** (2009) 237-250.

[82] F. Liu and D. Zhang, 3D fingerprint reconstruction system using feature correspondences and prior estimated finger model, *Pattern Recognition* **47(1)** (2014) 178-193.

[83] Y. Liu, Privacy regulations on biometrics in Australia, *Computer Law and Security Review* **26(4)** (2010) 355-367.

[84] H. Liu, Force field convergence map and Log-Gabor filter based multi-view ear feature extraction, *Neurocomputing* **76(1)** (2012) 2-8.

[85] S. Lucas and R. Damper, Signature verification with a syntactic neural net, *IJCNN International Joint Conference on Neural Networks* **1** (1990) 373-378.

[86] I.R. Lujan, G. Bailador, C.S. Avila, A. Herrero and G. Miguel, Analysis of pattern recognition and dimensionality reduction techniques for odor biometrics, *Knowledge-Based Systems* **52** (2013) 279-289.

[87] R. Madhusudhan and R.C. Mittal, Dynamic ID-based remote user password authentication schemes using smart cards, *Journal of Network and Computer Applications* **35(4)** (2012) 1235-1248.

[88] T. Matsuura and S. Okamura, On FIR filter for signature verification,*38th Midwest Symposium on Circuits and Systems* **1**(1996) 366-369.

[89] T. Matsuura and H. Sakai, On stochastic representation of handwriting process and itsapplication to signature verification, *ICSP '96. 1996 3rd International Conference on Signal Processing Proceedings* **2** (1996) 1330-1333 .

[90] T. Matsuura and H. Togiishi, Two dimensional AR model of signing process and its application to online signature verification, *IEEE International Conference on Electronics*, Circuits and Systems **2** (1998) 545-548.

[91] A. McCabe, Hidden Markov modelling with simple directional features for effective and efficient handwriting verification, 6th Pacific Rim International Conference on Artificial Intelligence, Melbourne, Australia, 2000.

[92] D. Mital and K. Lau, A microprocessor-based signature verification system, *IEEE Transactions on Consumer Electronics* **35(4)** (1989) 845-851.

[93] M. Munich and P. Perona, Visual signature verification using an arc-length, *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2** (1999) 180-186.

[94] E. Mostafa, R. Hammoud, A. Asem and A. Farag, Face recognition in low resolution thermal images, *Computer Vision and Image Understanding* **117(12)** (2013) 1689-1694.

[95] V. Nalwa, Automatic on-line signature verification, *Proceedings of the IEEE* **85(2)** (1997) 215-239.

[96] L. Nanni, E. Maiorana, A. Lumini and P. Campisi, Combining local, regional and global matchers for a template protected on-line signature verification system, *Expert Systems with Applications: An International Journal* **37(5)** (2010) 3676-3684.

[97] M. Nappi and H. theychsler, Robust re-identification using randomness and statistical learning: Quo vadis, *Pattern Recognition Letters* **33(14)** (2012) 1820-1827.

[98] W. Nelson, W. Turin and T. Hastie, Statistical methods for on-line signature verifi- cation, *International Journal of Pattern Recognition and Artificial Intelligence* **8(3)** (1994) 749-770.

[99] B. Oh, K. Toh, K. Choi, A.J. Teoh and J. Kim, Extraction and fusion of partial face features for cancelable identity verification, *Pattern Recognition* **45(9)** (2012) 3288-3303.

[100] M. Parizeau and R. Plamondon, A comparative analysis of regoinal correlation, dynamic time warping and skeletal tree matching for signature verification, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12(7)** (1990) 710-717.

[101] J. Parks, D. Carr and P. Fox, *Apparatus for Signature Verification*, US Patent Number **4** (1985) 495-644.

[102] M. Parodi and J.C. Gomez, Legendre polynomials based feature extraction for online signature verification. Consistency analysis of feature combinations *Pattern Recognition* **47(1)** (2014) 128-140.

[103] M. Paulik, N. Mohankrishnan and M. Nikiforuk, A time varying vector autoregressive model for signature verification, *Proceedings of the 37th Midwest Symposium on Circuits and Systems* **2** (1994) 1395-1398.

[104] R. Phelps, A holistic approach to signature verification, *Proceedings of the 6th International Conference on Pattern Recognition* **2** (1982) 1187.

[105] R. Plamondon, The design of an on-line signature verification system: From theory to practice, *International Journal of Pattern Recognition and Artificial Intelligence* **8(3)** (1994) 795-811.

[106] R. Plamondon, C. Reilly, J. Galbally, A. Almaksour and . Anquetil, Recent developments in the study of rapid human movements with the kinematic theory:

Applications to handwriting and signature synthesis, *Pattern Recognition Letters* **35(1)** (2014) 225-235.

[107] A.Prieto, M. Atencia and F.Sandoval, Advances in artificial neural networks and machine learning, *Neurocomputing* **121** (2013)

[108] B. Purgason and D. Hibler, Security Through Behavioral Biometrics and Artificial Intelligence, *Procedia Computer Science* **12** (2012) 398-403.

[109] J. Qian, J. Yang and G. Gao, Discriminative histograms of local dominant orientation (D-HLDO) for biometric image feature extraction, *Pattern Recognition* **46(10)** (2013) 2724-2739.

[110] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE* **77(2)** (1989) 257-285.

[111] L.R. Rabiner, Fundamentals of Speech Recognition, Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1993.

[112] S. Rashidi, A. Fallah and F. Towhidkhah, Feature extraction based DCT on dynamic signature verification, *Scientia Iranica* **19(6)** (2012) 1810-1819.

[113] D.A. Reid, S. Samangooei, C. Chen, M.S. Nixon and A. Ross, Chapter 13 - Soft Biometrics for Surveillance: An Overview, *Handbook of Statistics* **31** (2013) 327-352.

[114] J. Richiardi, H, Ketabdar and A. Drygajlo, Local and global feature selection for on-line signature verification, *Proceedings of the Eighth International Conference on Document Analysis and Recognition* (ICDAR 2005), (2005) 625-629.

[115] W. Rong, B. Bhanu, Ninad and S. Thakoor, Learning small gallery size for prediction of recognition performance on large populations, *Pattern Recognition* **46(12)** (2013) 3533-3547.

[116] M. Saeed, K. Javed and H. A. Babri, Machine learning using Bernoulli mixture models: Clustering, rule extraction and dimensionality reduction, *Neurocomputing* **119** (2013) 366-374.

[117] N. Saini and A. Sinha, Biometrics based key management of double random phase encoding scheme using error control codes, *Optics and Lasers in Engineering* **51(8)** (2013) 1014-1022.

[118] H. Sakoe and S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, *IEEE Transactions on Acoustics, Speech and Signal Processing* **26(1)** (1978) 43-49.

[119] C. Schmidt, Signature verification using time-delay neural networks, *Proceedings of the 37th Midwest Symposium on Circuits and Systems* **2** (1994) 1395-1398.

[120] A. Serrano, I.M. Diego, C. Conde and E. Cabello, Recent advances in face biometrics with Gabor wavelets: A review, *Pattern Recognition Letters* **31(5)** (2010) 372-381.

[121] J. Shin and T. Okuyama, Detection of alcohol intoxication via online handwritten signature verification, *Pattern Recognition Letters* **35** (2014) 101-104.

[122] E. Singer and R. Lippmann, A speech recognizer using radial basis function neural networks in an HMM framework, *Acoustics, Speech, and Signal Processing* **1** (1992) 629-632.

[123] R. Singh, M. Vatsa, A. Ross and A. Noore, Biometric classifier update using online learning: A case study in near infrared face verification, *Image and Vision Computing* **28(7)** (2010) 1098-1105.

[124] Y.N. Singh, S.K. Singh and P. Gupta, Fusion of electrocardiogram with unobtrusive biometrics: An efficient individual authentication system, *Pattern Recognition Letters* **33(14)** (2012) 1932-1941.

[125] SMARTPEN biometric authentication system website, www.smartpen.net.

[126] D. Smeets, J. Keustermans, D. Vandermeulen and P. Suetens, meshSIFT: Local surface features for 3D face recognition under expression variations and partial data, *Computer Vision and Image Understanding* **117(2)** (2013) 158-169.

[127] J. Sternberg, Automated signature verification using handwriting pressure, *Wescon Technical Papers* **31(4)** (1975).

[128] D. Stefan, X. Shu and D. Yao, Robustness of keystroke-dynamics based biometrics against synthetic forgeries,*Computers and Security* **31(1)** (2012) 109-121.

[129] S.P. Tankasala, P. Doynov and R. Derakhshani, Visible Spectrum Bi-Modal Ocular Biometrics, *Procedia Technology* **6** (2012) 564-573.

[130] A. Teoh, B. Jin and T. Connie, Remarks on BioHashing based cancelable biometrics in verification system, *Neurocomputing* **69(16-18)** (2006) 2461-2464.

[131] A.B.J. Teoh, Y.W. Kuan and S. Lee, Cancellable biometrics and annotations on BioHash, *Pattern Recognition* **41(6)** (2008) 2034-2044.

[132] T. Tirelli, M. Gamba and D. Pessani, Support vector machines to model presence/absence of Alburnus alburnus alborella (Teleostea, Cyprinidae), *in Norththeystern Italy: Comparison with other machine learning techniques Comptes Rendus Biologies* **335(10-11)** (2012) 680-686.

[133] D.T. Toledano, R.F. Pozo, A.H. Trapote and L.H. Gomez, Usability evaluation of multi-modal biometric verification systems, *Interacting with Computers* **18(5)** (2006) 1101-1122.

[134] L. Tseng and T. Huang, An online chinese signature verification scheme based on the ART1, *Neural network, IJCNN, International Joint Conference on Neural Networks* **3** (1992) 624-630.

[135] S.K.V. Valtchev and S. Young, Recurrent input transformations for hidden Markov models, *Acoustics, Speech, and Signal Processing* **2** (1993) 287-290.

[136] H.L. van Trees, *Detection, Estimation, and Modulation Theory, Part I*, John Wiley and Sons, New York, USA, (2004).

[137] V. Vapnik, The Nature of Statistical Learning Theory, *Information Science and Statistics* Spinger-Verlag, Berlin, (2010).

[138] J.F. Vargas, M.A. Ferrer, C.M. Travieso and J.B. Alonso, Off-line signature verification based on grey level information using texture features, *Pattern Recognition* **44(2)** (2011) 375-385.

[139] S.L. Wang and A.W.C. Liew, Physiological and behavioral lip biometrics: A comprehensive study of their discriminative power, *Pattern Recognition* **45(9)** (2012) 3328-3335.

[140] C. Wen, M. Chang, B. Jeng and H.F. Yau, Signature verification based on distortion measure and spectral correlation, *Proceedings of the International Society for Optical Engineering* **2564** (1995) 252-260.

[141] J. Wen, B. Fang, Y.Y. Tang and T. Zhang, Model-based signature verification with rotation invariant features, *Pattern Recognition* **42(7)** (2009) 1458-1466.

[142] A. Wilson and A. Bobick, Nonlinear parametric hidden Markov models, *MIT Media Lab Perceptual Computing Section Technical Report 424* (1996).

[143] Q. Wu, S. Lee and I. Jou, Online signature verification using LPC cepstrum and neural networks, *IEEE Transactions on Systems, Man and Cybernetics - Part B* **27(1)** (1997) 148-153.

[144] Q. Wu, S. Lee and I. Jou, Online signature verification based on split-and-merge match-ing mechanism, *Pattern Recognition Letters* **18(7)** (1997) 665-673.

[145] Q. Wu, S. Lee and I. Jou, Online signature verification based on logarithmic spectrum, *Pattern Recognition* **31(12)** (1998) 1865-1871.

[146] S. Wu, M. Sun and J.Yang, Stochastic neighbor projection on manifold for feature extraction, *Neurocomputing* **74(17)** (2011) 2780-2789.

[147] B. Xianye, W. Meng, R. Yan and W. K. Kernel, coupled distance metric learning for gait recognition and face recognition, *Neurocomputing* **120** (2013) 577-589.

[148] B. Xianye, W. Meng, R. Yan and W. Kejun, An improved biometrics technique based on metric learning approach, *Neurocomputing* **97(15)** (2012) 44-51.

[149] C. Xiaojun, W. Zicheng, P. Yiguo and S. Jinqiao, A Continuous Re-Authentication Approach Using Ensemble Learning, *Procedia Computer Science* **17** (2013) 870-878.

[150] Y. Xuhua, T. Furuhashi, K. Obata and Y. Uchikawa, Selection of features for signature verification using the genetic algorithm, *Computers and Industrial Engineering* **30(4)** (1996) 1037-1045.

[151] Y. Xu, D. Zhang and Y. Jing-Yu, A feature extraction method for use with bimodal biometrics, *Pattern Recognition* **43(3)** (2010) 1106-1115.

[152] Y. Xu, Q. Zhu and D. Zhang, Combine crossing matching scores with conventional matching scores for bimodal biometrics and face and palmprint recognition experiments, *Neurocomputing* **74(18)** (2011) 3946-3952.

[153] Y. Xu, Z. Fan, M. Qiu, D. Zhang and J. Yang, A sparse representation method of bimodal biometrics and palmprint recognition experiments *Neurocomputing* **103** (2013) 164-171.

[154] Y. Xu, Q. Zhu, Z. Fan, Y. Wang and J. S. Pan, From the idea of sparse representation to a representation-based transformation method for feature extraction, *Neurocomputing* **113(3)** (2013) 168-176.

[155] N. Yager and A. Amin, Dynamic registration selection for fingerprint verification, *Pattern Recognition* **39(11)** (2006) 2141-2148.

[156] L. Yang, B. Widjaja and R. Prasad, Application of hidden Markov models for signature verification, *Pattern Recognition* **28(2)** (1995) 161-170.

[157] K. Yang, E.Y. Du and Z. Zhou, Consent biometrics, *Neurocomputing* **100(16)** (2013) 153-162.

[158] Y.F. Yao, X.Y. Jing and H.S. Wong, Face and palmprint feature level fusion for single sample biometrics recognition, *Neurocomputing* **70 (7-9)** (2007) 1582-1586.

[159] M. Yasuhara and M. Oka, Signature verification experiment based on nonlinear time alignment: A feasibility study, *IEEE Transactions on Systems, Man and Cybernetics* **7(3)** (1977) 212-216.

[160] K. Yasuda, D. Muramatsu, S. Shirato and T. Matsumoto, Visual-based online signature verification using features extracted from video, *Journal of Network and Computer Applications* **33(3)** (2010) 333-341.

[161] J.M.G. Zavaliagkos, Y. Zhao and R. Schwartz, A hybrid segmental neural net/hidden Markov model system for continuous speech recognition, *IEEE Proceedings* (1994).

[162] B. Zavar and M.S. Nixon, On guided model-based analysis for ear biometrics, *Computer Vision and Image Understanding* **115(4)** (2011) 487-502.

[163] D. Zhang, Z. Liu and J. Yan, Dynamic tongueprint: A novel biometric identifier, *Pattern Recognition* **43(3)** (2010) 1071-1082.