Off-line signature verification using ensembles of local Radon transform-based HMMs

by

Mark Stuart Panton

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Applied Mathematics at the University of Stellenbosch

Supervisor: Dr. J. Coetzer

December 2010

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature: .....
MS Panton

Date: .....

Abstract

Off-line signature verification using ensembles of local Radon transform-based HMMs

MS Panton

Thesis: MSc (Applied Mathematics)

December 2010

An off-line signature verification system attempts to authenticate the identity of an individual by examining his/her handwritten signature, after it has been successfully extracted from, for example, a cheque, a debit or credit card transaction slip, or any other legal document. The questioned signature is typically compared to a model trained from known positive samples, after which the system attempts to label said signature as genuine or fraudulent.

Classifier fusion is the process of combining individual classifiers, in order to construct a single classifier that is more accurate, albeit computationally more complex, than its constituent parts. A combined classifier therefore consists of an ensemble of base classifiers that are combined using a specific fusion strategy.

In this dissertation a novel off-line signature verification system, using a multi-hypothesis approach and classifier fusion, is proposed. Each base classifier is constructed from a hidden Markov model (HMM) that is trained from features extracted from local regions of the signature (local features), as well as from the signature as a whole (global features). To achieve this, each signature is zoned into a number of overlapping circular retinas, from which said features are extracted by implementing the discrete Radon transform. A global retina, that encompasses the entire signature, is also considered.

Since the proposed system attempts to detect high-quality (skilled) forgeries, it is unreasonable to assume that samples of these forgeries will be available for each new writer (client) enrolled into the system. The system is therefore constrained in the sense that only positive training samples, obtained from each writer during enrolment, are available. It is however reasonable to assume that both positive and negative samples are available for a representative subset of so-called guinea-pig writers (for example, bank employees).

These signatures constitute a convenient optimisation set that is used to select the most proficient ensemble. A signature, that is claimed to belong to a legitimate client (member of the general public), is therefore rejected or accepted based on the majority vote decision of the base classifiers within the most proficient ensemble.

When evaluated on a data set containing high-quality imitations, the inclusion of local features, together with classifier combination, significantly increases system performance. An equal error rate of 8.6% is achieved, which compares favorably to an achieved equal error rate of 12.9% (an improvement of 33.3%) when only global features are considered.

Since there is no standard international off-line signature verification data set available, most systems proposed in the literature are evaluated on data sets that differ from the one employed in this dissertation. A direct comparison of results is therefore not possible. However, since the proposed system utilises significantly different features and/or modelling techniques than those employed in the above-mentioned systems, it is very likely that a superior combined system can be obtained by combining the proposed system with any of the aforementioned systems. Furthermore, when evaluated on the same data set, the proposed system is shown to be significantly superior to three other systems recently proposed in the literature.

Uittreksel

Statiese handtekeningverifikasie met behulp van ensembles van gelokaliseerde Radon-transform-gebaseerde HMMe

MS Panton

Tesis: MSc (Toegepaste Wiskunde)

Desember 2010

Die doel van 'n statiese handtekening-verifikasiestelsel is om die identiteit van 'n individu te bekragtig deur sy/haar handgeskrewe handtekening te analiseer, nadat dit suksesvol vanaf byvoorbeeld 'n tjek,'n debiet- of kredietkaattransaksiestrokie, of enige ander wettige dokument onttrek is. Die bevraagtekende handtekening word tipies vergelyk met 'n model wat afgerig is met bekende positiewe voorbeelde, waarna die stelsel poog om die handtekening as eg of vervals te klassifiseer.

Klassifiseerder-fusie is die proses waardeer individuele klassifiseerders gekombineer word, ten einde 'n enkele klassifiseerder te konstrueer, wat meer akkuraat, maar meer berekeningsintensief as sy samestellende dele is. 'n Gekombineerde klassifiseerder bestaan derhalwe uit 'n ensemble van basis-klassifiseerders, wat gekombineer word met behulp van 'n spesifieke fusie-strategie.

In hierdie projek word 'n nuwe statiese handtekening-verifikasiestelsel, wat van 'n multi-hipotese benadering en klassifiseerder-fusie gebruik maak, voorgestel. Elke basis-klassifiseerder word vanuit 'n verskuilde Markov-model (HMM) gekonstrueer, wat afgerig word met kenmerke wat vanuit lokale gebiede in die handtekening (lokale kenmerke), sowel as vanuit die handtekening in geheel (globale kenmerke), onttrek is. Ten einde dit te bewerkstellig, word elke handtekening in 'n aantal oorvleulende sirkulêre retinas gesoneer, waaruit kenmerke onttrek word deur die diskrete Radon-transform te implementeer. 'n Globale retina, wat die hele handtekening in beslag neem, word ook beskou.

Aangesien die voorgestelde stelsel poog om hoë-kwaliteit vervalsings op te spoor, is dit onredelik om te verwag dat voorbeelde van hierdie handtekeninge beskikbaar sal wees vir elke nuwe skrywer (kliënt) wat vir die stelsel registreer. Die stelsel is derhalwe beperk in die sin dat slegs positiewe afrigvoorbeelde, wat bekom is van elke skrywer tydens registrasie, beskikbaar is. Dit is egter redelik

om aan te neem dat beide positiewe en negatiewe voorbeelde beskikbaar sal wees vir 'n verteenwoordigende subversameling van sogenaamde proefkonynskrywers, byvoorbeeld bankpersoneel. Hierdie handtekeninge verteenwoordig 'n gereieflike optimeringstel, wat gebruik kan word om die mees bekwame ensemble te selekteer. 'n Handtekening, wat na bewering aan 'n wettige kliënt (lid van die algemene publiek) behoort, word dus verwerp of aanvaar op grond van die meerderheidstem-besluit van die basis-klassifiseerders in die mees bekwame ensemble.

Wanneer die voorgestelde stelsel op 'n datastel, wat hoë-kwaliteit vervalsings bevat, ge-evalueer word, verhoog die insluiting van lokale kenmerke en klassifiseerder-fusie die prestasie van die stelsel beduidend. 'n Gelyke foutkoers van 8.6% word behaal, wat gunstig vergelyk met 'n gelyke foutkoers van 12.9% ('n verbetering van 31.1%) wanneer slegs globale kenmerke gebruik word.

Aangesien daar geen standard internasionale statiese handtekening-verifikasiestelsel bestaan nie, word die meeste stelsels, wat in die literatuur voorgestel word, op ander datastelle ge-evalueer as die datastel wat in dié projek gebruik word. 'n Direkte vergelyking van resultate is dus nie moontlik nie. Desnieteenstaande, aangesien die voorgestelde stelsel beduidend ander kenmerke en/of modeleringstegnieke as dié wat in bogenoemde stelsels ingespan word gebruik, is dit hoogs waarskynlik dat 'n superieure gekombineerde stelsel verkry kan word deur die voorgestelde stelsel met enige van bogenoemde stelsels te kombineer. Voorts word aangetoon dat, wanneer op dieselfde datastel ge-evalueer, die voorgestelde stelstel beduidend beter vaar as drie ander stelsels wat onlangs in die literatuur voorgestel is.

Acknowledgements

I would like to express my sincere gratitude to the following people for enabling me to successfully complete this dissertation:

- My supervisor, Johannes Coetzer. This dissertation would not have been possible without his encouragement, guidance and support.
- My parents, for their financial support.
- Hans Dolfing, for granting us permission to use his signature data set.

Contents

List of Figures

List of Tables

List of Symbols

Data

$\boldsymbol{O} = \{\boldsymbol{O}^+, \boldsymbol{O}^-\}$	Optimisation set, containing both positive (+) and negative signatures (−)
$\boldsymbol{E} = \{\boldsymbol{E}^+, \boldsymbol{E}^-\}$	Evaluation set, containing both positive (+) and negative signatures (−)
$\boldsymbol{T}_O^+$	Training signatures belonging to the optimisation writers
$\boldsymbol{T}_E^+$	Training signatures belonging to the evaluation writers
N_w	Number of writers in a data set
N_T	Number of training signatures per writer

Feature Extraction

$\boldsymbol{G}_\mu = (G_x, G_y)$	Gravity centre coordinates of a signature image
Z_h	Set containing the horizontal intervals into which a signature is zoned
Z_v	Set containing the vertical intervals into which a signature is zoned
N_r	Number of retinas (and therefore candidate base classifiers) considered
γ	Radius of each local retina
$\mathbf{x}_i$	Feature vector
N_θ	Number of angles or projections used to calculate the discrete Radon transform (DRT)
T	Length of an observation sequence ($T = 2N_\theta$)

$\mathbf{X}_w^r = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$	Observation sequence extracted from retina r of writer w
$\mathbf{X}_{(w)}^r$ = $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$	Observation sequence extracted from retina r of a *claimed* writer w
I_i	Intensity of the ith pixel
Ξ	Total number of pixels in an image
R_j	The jth beam-sum
α_{ij}	Weight of the contribution of the ith pixel to the jth beam-sum
d	Dimension of each feature vector ($d = 2\gamma$)

Hidden Markov Models

λ_w^r	A hidden Markov model (HMM) representing retina r of writer w
N	Number of states in a hidden Markov model (HMM)
s_j	The jth state
$\mathbf{S} = \{s_1, s_2, \ldots, s_N\}$	Set of states
π_j	The probability of entering an HMM at the jth state
$\boldsymbol{\pi} = \{\pi_1, \pi_2, \ldots, \pi_N\}$	Initial state distribution
$a_{i,j}$	The probability of transitioning from state s_i to state s_j
$\mathbf{A} = \{a_{i,j}\}$	State transition probability distribution
q_t	State at time t
$f(\mathbf{x}\|s_j, \lambda)$	The probability density function that quantifies the similarity between a feature vector $\mathbf{x}$ and the state s_j of an HMM λ
$f(\mathbf{X}\|\lambda)$	The probability density function, that quantifies the similarity between the observation sequence $\mathbf{X}$ and the hidden Markov model λ
$D(\mathbf{X}\|\lambda)$	The dissimilarity between the observation sequence $\mathbf{X}$ and the HMM λ ($D(\mathbf{X}\|\lambda) = -\ln f(\mathbf{X}\|\lambda)$)

Rotation Normalisation

$\mathbf{Q}^* = \{q_1^*, q_2^*, \ldots, q_T^*\}$	A universal reference state sequence
$\mathbf{Q} = \{q_1, q_2, \ldots, q_T\}$	Most probable state sequence determined using Viterbi alignment
$\mathbf{Q}^{'}$	Modified state sequence
μ_Q	Mean difference between $\mathbf{Q}^*$ and $\mathbf{Q}^{'}$
Δ	Correction angle

Thresholding and Score Normalisation

τ	un-normalised threshold parameter
ϕ	normalised threshold parameter
ρ	calibrated threshold parameter
$D_*(\mathbf{X}\|\lambda)$	The normalised dissimilarity value between the observation sequence X and the HMM λ
μ_w^r	Mean dissimilarity value for the training signatures for retina r of writer w
σ_w^r	Standard deviation of the dissimilarity values for the training signatures for retina r of writer w

Ensemble Selection and Classifier Combination

$C_w^r\{\sim\}$	Continuous classifier associated with retina r of writer w
$C_w^r\{\rho\}$	Discrete classifier associated with retina r of writer w, obtained by imposing the threshold ρ
ω_r	Class label (decision) for retina r
ω_f	Final class label (decision)
Υ	Fusion function
Ψ	Selected ensemble
N_S	Size of the selected ensemble
T_{N_S}	Total number of possible ensembles of size N_S
Ω_{ex}	Number of ensembles generated using the exhaustive generation approach
Ω_{pc}	Number of ensembles generated using the performance-cautious approach
Ω_{ec}	Number of ensembles generated using the efficiency-cautious approach

Experiments and Performance Evaluation

k	Number of folds
R	Number of repetitions in k-fold cross validation with data shuffling
L	Number of experimental iterations
ξ	Imposed operational criterion
ζ	Mean performance achieved across the L *optimisation* sets
μ_ξ	Mean operating value achieved across the L *evaluation* sets
σ_ξ	Standard deviation of the operating values across the L *evaluation* sets
μ_ζ	Mean performance achieved across the L evaluation sets
σ_ζ	Standard deviation of the performance achieved across the *evaluation* sets

List of Acronyms

AUC	Area under curve
CRF	Conditional Random Field
DRT	Discrete Radon transform
DTW	Dynamic time warping
EER	Equal error rate
FN	False negative
FNR	False negative rate
FP	False positive
FPR	False positive rate
GMM	Gaussian mixture model
HMM	Hidden Markov model
MAROC	Maximum attainable receiver operating characteristic
NN	Neural network
OP	Operating predictability
OPS	Operating point stability
OS	Operating stability
PDF	Probability density function
PP	Performance predictability
PS	Performance stability
ROC	Receiver operating characteristic
SVM	Support vector machine
TN	True negative
TNR	True negative rate
TP	True positive
TPR	True positive rate

Chapter 1

Introduction

1.1 Background

In many societies, handwritten signatures are the socially and legally accepted norm for authorising financial transactions, such as cheque, credit and debit payments, as well as providing evidence of the intent or consent of an individual in legal agreements. In practice, if a questioned signature is sufficiently similar to known authentic samples of the claimed author's signature, it is deemed to be *genuine*. Alternatively, if a questioned signature differs significantly from known authentic samples, it is deemed to be *fraudulent*.

In many practical scenarios, it is imperative that signatures are verified accurately and timeously. However, due to the time-consuming and cumbersome nature of manual authentication, handwritten signatures are typically only verified when a dispute arises, or when the value of a financial transaction exceeds a certain threshold.

The purpose of this research is therefore to develop a signature verification system that automates the process of signature authentication. For an automatic signature verification system to be viable, it should provide a substantial benefit over the utilisation of human verifiers, by offering superior speed and accuracy, and by minimising costs.

In the remainder of this chapter we emphasise some key issues and concepts relevant to this project (Section 1.2), state the project's objectives (Section 1.3), provide a brief synopsis of the system developed in this dissertation (Section 1.4), put the main results into perspective (Section 1.5), and list the major contributions to the field resulting from this study (Section 1.6).

1.2 Key issues, concepts and definitions

1.2.1 Human and machine verification

Although machines are superior to humans at performing fast repetitive calculations, automatic pattern recognition (that is, pattern recognition performed by a machine) is only achievable under very specific, controlled circumstances. A facial recognition system, for example, may only be able to successfully recognise a face when a user assumes a certain pose, or under controlled lighting conditions. The automatic recognition of text (optical character recognition) or handwriting, generally requires that said text or handwriting is much clearer than what is required for human legibility.

Despite the above-mentioned limitations of automatic recognitions systems, there are many applications in which the conditions can be controlled to such an extent that automatic recognition becomes more convenient and efficient that manual recognition−one such application is the automatic classification of handwritten signatures.

As an aside, a human-centric handwritten signature verification system is proposed in Coetzer and Sabourin (2007) that exploits the synergy between human and machine capabilities. The authors demonstrate that superior performance is achievable by implementing a hybrid (combined) human-machine classifier, when compared to the performance achievable by either using an unassisted human or an unassisted machine.

1.2.2 Biometrics

A biometric, or biometric characteristic, refers to a property of a human being that can be used to uniquely identify him/her, for the purposes of access control, surveillance, etc. Biometric characteristics can be broadly divided into two categories, namely behavioural and physiological characteristics. Physiological biometrics are attributes that constitute a physical trait of an individual, for example, facial features, fingerprints, iris patterns and DNA. Examples of behavioural biometrics include voice, handwritten signatures and gait. The distinction between physiological and behavioural biometrics is not always clear. An individual's voice, for example, is both a physiological attribute (determined by each individual's vocal chord structure) and a behavioural attribute (an individual may alter his/her voice to a certain extent).

Although a handwritten signature, which constitutes a pure behavioural biometric, is not the most secure or reliable biometric, it is the most socially and legally accepted biometric in use today (Plamondon and Shihari (2000)).

1.2.3 Recognition and verification

It is important to draw a distinction between what is meant by *recognition* and *verification* (or authentication). A verification system determines whether a claim that a pattern belongs to a certain class is true or false. A recognition system, on the other hand, attempts to determine to which class (from a set of classes known to the system) said pattern belongs.

It is worth emphasising that the terms *classifier* and *class* are applicable to both verification and recognition. A recogniser can be considered to be a multi-class classifier, whereas a verifier can be considered to be a two-class classifier, consisting of a positive ("true") class and a negative ("false") class. In the context of signature verification, we use the latter definition of a classifier throughout this dissertation.

1.2.4 Off-line and on-line signature verification

Signature verification systems can be categorised into off-line and on-line systems. Off-line systems use features extracted from a static digitised image of a signature that is typically obtained by scanning or photographing the document that contains said signature. On-line systems, on the other hand, require an individual to produce his/her handwritten signature on a digitising tablet that is capable of also recording dynamic information, like pen pressure, pen velocity and pen angle.

Since on-line systems also have dynamic signature information to their disposal, they generally outperform off-line systems, but are not applicable in all practical scenarios. Static signatures provide an explicit association between an individual and a document. A signature on a document therefore, in addition to providing a means for authenticating the writer's identity, also indicates that the writer consents to the content of the document, whether it be a cheque (stipulating that a specific amount is to be paid into a specific account) or a legal contract (stipulating that the writer agrees to certain terms and conditions). When a digitising tablet is used as part of an on-line signature verification system, the document-signature association is removed.

1.2.5 Forgery types

It is important to distinguish between different types of forgeries, since specific signature verification systems typically aim to detect specific forgery types. Since there is no standardised categorisation of forgery types, we adopt the definitions found in Dolfing (1998), Coetzer (2005), and Bastista *et al.* (2007).

Three basic forgery types are defined, in increasing order of quality. A *random* forgery is produced by an individual who has no prior knowledge of (or is making no attempt to imitate) the appearance of the victim's signature, or knowledge of the victim's name. A genuine signature belonging to any in-

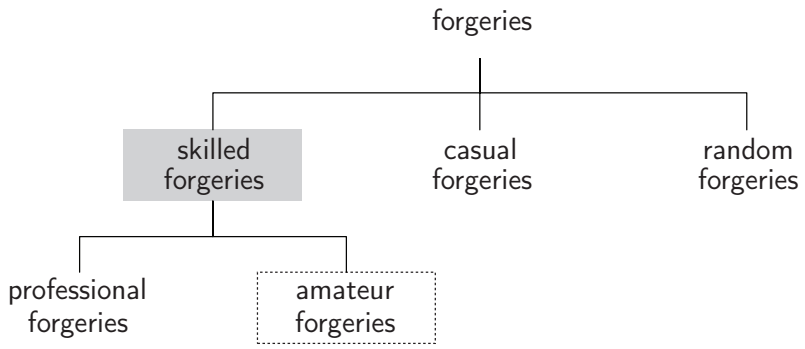


Figure 1.1: Forgery types. The quality of the forgeries decreases from left to right. In this dissertation we aim to detect only amateur-skilled forgeries (shaded block).

dividual, other than the victim, can therefore be considered to be a random forgery. *Casual* forgeries are produced when the forger has some prior knowledge of the victim's initials and surname only, and no further knowledge of the actual appearance of the victim's signature. Since there is generally only a very weak correlation between the appearance of an individual's signature and the individual's name, a casual forgery is typically of similar quality to a random forgery. *Skilled* forgeries are produced when the forger has prior information about the actual appearance of the targeted individual's signature.

Skilled forgeries can be subdivided into two categories, namely *amateur* forgeries and *professional* forgeries. Amateur forgeries are typically produced by an individual who has access to one or more copies of the victim's signature and ample time to practise imitating them. Professional forgeries, on the other hand, are produced by a person, who, in addition to having access to the victim's signature, also has expert forensic knowledge of human handwriting, and is able to imitate the victim's signature with great precision.

Professional forgeries are difficult to detect by both humans and machines. The detection of professional forgeries therefore poses a challenging problem. Random and casual forgeries, on the other hand, are usually trivial to detect, by both humans and machines.

The system developed in this dissertation is designed to detect *amateur-skilled* forgeries, and is optimised and evaluated using a data set containing signatures of this type. Throughout this dissertation, we simply refer to amateur-skilled forgeries as skilled forgeries.

Since the data set utilised in this dissertation contains very few professional forgeries, we do not optimise and evaluate the proposed system using professional forgeries. It is important to note that a system that is proficient at detecting amateur forgeries is generally also proficient at detecting casual and random forgeries.

The categorisation of the different forgery types is depicted in Figure 1.1. Examples of different forgery types are shown in Figure 1.2.

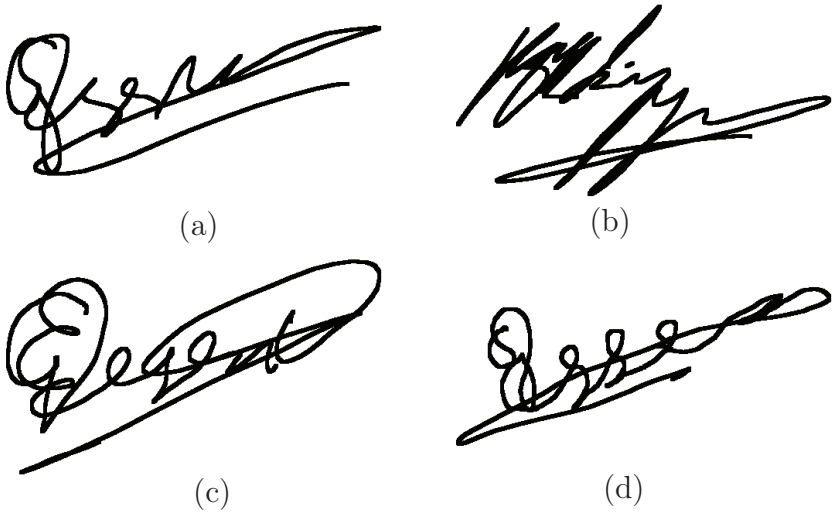


Figure 1.2: Examples of different forgery types. (a) A genuine signature. An example of (b) a random forgery, (c) an amateur-skilled forgery , and (d) a professional-skilled forgery of the signature in (a). In this dissertation, we aim to detect only amateur-skilled forgeries.

1.2.6 Generative and discriminative models

In the machine learning and statistics literature there are two types of models, namely *generative* models and *discriminative* models. Examples of generative models include hidden Markov models (HMMs), Gaussian mixture models (GMMs) and Naive Bayesian models. Discriminative models include support vector machines (SVMs), conditional random fields (CRFs) and neural networks (NNs).

A generative model is a full probabilistic model of all variables, whereas a discriminative model only models the conditional probability of a target variable, given the observed variables. The training of a discriminative model therefore requires (in the context of a verifier or two-class classifier) the availability of both positive and negative training samples, making the use of discriminative models unfeasible for signature verification systems that aim to detect skilled forgeries. A generative model, on the other hand, can be trained using positive training samples only.

A discrete classifier (crisp detector) is associated with a discriminative model and outputs only a class label. A continuous classifier (soft detector) is associated with a generative model and assigns a score (or dissimilarity value) to an input sample that can be converted into a discrete classifier by imposing a specific decision threshold on said score (or dissimilarity value).

Since only positive training samples are available for each writer enrolled into the system proposed in this dissertation, we model each writer's signature with *generative* hidden Markov models (HMMs).

1.2.7 Performance evaluation measures

When a questioned signature pattern is matched with an HMM (in this dissertation) a dissimilarity value is obtained. Positive (genuine) signatures should, in practice, have lower dissimilarity values than negative (fraudulent) signatures. An example of a dissimilarity value distribution for positive and negative signatures is shown in Figure 1.3a.

Throughout this dissertation we use the *false negative rate* (FNR) and the *false positive rate* (FPR) to evaluate the performance of a classifier. The false positive rate (FPR) is defined as

$$\text{FPR} = \frac{\text{number of false positives}}{\text{number of forgeries}}, \tag{1.2.1}$$

while the false negative rate (FNR) is defined as

$$\text{FNR} = \frac{\text{number of false negatives}}{\text{number of genuine signatures}}. \tag{1.2.2}$$

The true positive rate (TPR), where TPR = 1 - FNR, and the true negative rate (TNR), where TNR = 1 - FPR, are also considered.

By lowering the decision threshold of a generative classifier (thereby making it "stricter") the FPR can be decreased, however this is invariably at the expense of an increased FNR (see Figure 1.3b). A trade-off therefore exists between the FPR and the FNR for a generative classifier. The decision threshold can be chosen in such a way that the FPR is equal to the FNR. This error rate is referred to as the *equal error rate* (EER), and the associated threshold value is referred to as the *EER threshold*.

The trade-off that exists between the FPR and FNR can be conveniently depicted as a receiver operating characteristic (ROC) curve in ROC space, with the FPR on the horizontal axis, and TPR on the vertical axis (see Figure 1.3c). Each point on a ROC curve is referred to as an operating point and is associated with a specific decision threshold.

The performance evaluation measures are discussed in more detail in Chapter 6.

1.2.8 Local and global features

In order to achieve successful off-line signature verification, appropriate features have to be extracted from a static digitised image of each signature. We distinguish between two types of features, namely *local* features and *global* features. Global features are extracted from the *entire* signature image. Any change to a local region of the signature image will therefore influence all global features. This is in contrast to *local* features, which are extracted from local regions of the signature image. In order to extract local features, a signature image is typically *zoned* into local regions, using a grid-based zoning scheme.

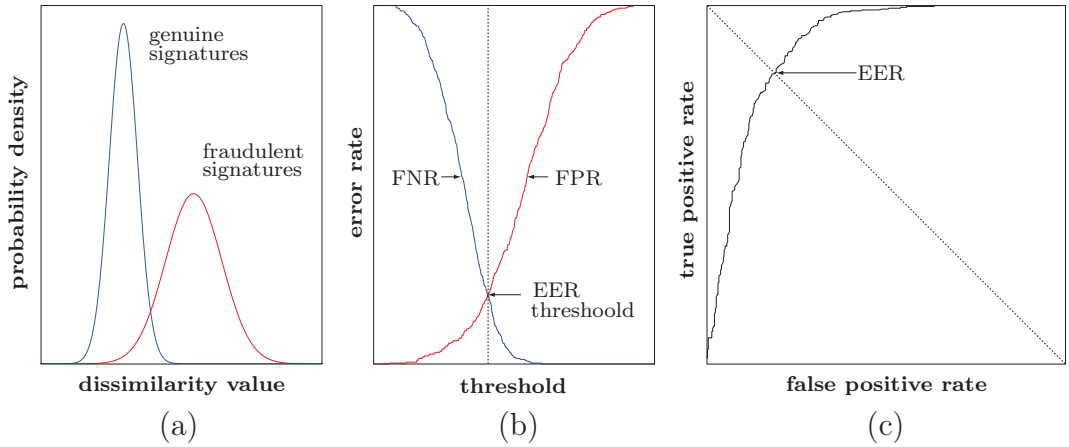


Figure 1.3: Performance evaluation measures. (a) A hypothetical distribution of dissimilarity values for positive and negative signatures. (b) The FPR and FNR plotted against the decision threshold. Note that a decrease in the FPR is invariably associated with an increase in the FNR, and visa versa. (c) The ROC curve corresponding to the FPRs and FNRs depicted in (b).

Any change to a local region of a signature will therefore only influence the features extracted from said region.

1.2.9 Classifier fusion

Classifier fusion, or classifier combination, is the process of combining individual classifiers (base classifiers), in order to construct a single classifier that is more accurate, albeit more computationally complex, than its constituent parts. A *combined classifier* therefore consists of an *ensemble* of *base classifiers* that are combined using a specific *fusion* strategy. In this dissertation we investigate two ensemble generation techniques in order to produce a pool of *candidate* ensembles, after which the optimal ensemble is selected based a specific operating criterion.

Classifier combination is often referred to as a *multi-hypothesis* approach to pattern recognition. Classifier fusion can be employed at two fundamentally different levels, namely at the score level (score-level fusion) or at the decision level (decision-level fusion). In score-level fusion the scores generated by the individual base classifiers are combined (for example, by averaging the individual scores), after which a decision threshold is imposed on the combined score in order to reach a final decision. In decision-level fusion, on the other hand, a decision is made by each individual base classifier (for example, by imposing a threshold on the score (or dissimilarity value) generated by each base classifier), after which the individual decisions are combined in order to reach a final decision. In this dissertation, a decision-level fusion strategy (namely majority voting) is employed. Classifier fusion is discussed in more detail in

Chapter 8.

1.3 Objectives

It was recently shown in Coetzer (2005) that the utilisation of the discrete Radon transform (DRT) (for feature extraction) and a ring-structured continuous observation HMM (for signature modelling) provides an efficient and robust strategy for proficient off-line signature verification. Since (in Coetzer (2005)) the DRT of the *entire* signature image is extracted, only *global* features are considered. The main objective of this dissertation is to investigate whether a significant improvement in system performance is possible by also utilising *local* DRT-based features.

1.4 System overview

In this section we provide a brief overview of the system proposed in this dissertation. A data partitioning protocol, that accounts for the limitations of the available signature data, is detailed in Section 1.4.1. A brief overview of each of the proposed system's components is given in Sections 1.4.2−1.4.6, with references to the chapters in which said components are discussed in detail. The system outline is further clarified by providing the pseudocode in Section 1.4.7 and a flowchart in Figure 1.6. In Section 1.5 a synopsis of the results achieved for the proposed system is provided. The major contributions of this dissertation are discussed in Section 1.6, and the layout of this dissertation is provided in Section 1.7.

1.4.1 Data set and data partitioning

The data set ("Dolfing's data set") used in this dissertation was originally captured on-line for Hans Dolfing's Ph.D. thesis (Dolfing (1998)). This on-line signature data was converted into static signature images in Coetzer *et al.* (2004), and has subsequently been used to evaluate several off-line signature verification systems. Dolfing's data set contains the signatures of fifty-one writers. For each writer, there are fifteen training signatures, fifteen genuine test signatures and sixty *skilled* forgeries (with the exception of two writers, for which there are only thirty skilled forgeries). Dolfing's data set and the data partitioning protocol utilised in this dissertation are discussed in detail in Chapter 9.

It is important to be cognisant of the limitations on available signature data that a system designer would face in a real-world scenario, and to enforce these same limitations when designing and testing a system in an artificial research environment. In this section we discuss these limitations, and introduce terminology and notation that is used throughout this dissertation.

An important, albeit trivial, distinction is that between genuine and fraudulent signatures. In this dissertation, genuine signatures are referred to as *positive* signatures, and fraudulent signatures as *negative* signatures. We refer to the process of labelling a *questioned* signature (that is, a signature of which the authenticity is not yet decided) as classifying said signature as either positive or negative. We may therefore *accept* (classify as positive) or *reject* (classify as negative) said signature.

We distinguish between two disjoint signature data sets. The first set, referred to as the *optimisation* set, is denoted by $\boldsymbol{O}$, and represents signature data that is available to the system designer, before the system is deployed. This data set is used for the purpose of optimising the system parameters and should therefore contain representative signatures typically encountered in the general population. We assume that a group of so-called *guinea-pig* writers (for example, bank employees) are able to provide this data.

The *evaluation* set (denoted by $\boldsymbol{E}$), on the other hand, represents the signature data of actual clients enrolled into the system, *after* the system is deployed. The evaluation set therefore contains *unseen* data and plays no role in the design or optimisation of the system; it is used solely to evaluate the system's performance. The results achieved using the evaluation set therefore provide an *estimation* of the system's potential real-world performance.

Typically, when a user/client (referred to as a *writer*) is enrolled into a system, he/she is expected to provide several positive examples of his/her signature. These signatures are primarily used to train a model of said writer's signature, and are therefore referred to as *training* signatures. We use the symbols $\boldsymbol{T}_O^+$ and $\boldsymbol{T}_E^+$ to denote the training subsets of the optimisation set (that is, the guinea-pig writers) and evaluation set (actual clients), respectively. Since the signature verification system developed in this dissertation aims to detect skilled forgeries (as opposed to only random forgeries), it is highly impractical to also acquire negative examples (skilled forgeries) for *each* new writer enrolled into the system. The system is therefore limited in the sense that no training set that also contains negative examples is available.

The optimisation and evaluation sets therefore contain three groups of signatures: positive training signatures ($\boldsymbol{T}_O^+$ and $\boldsymbol{T}_E^+$), positive testing signatures ($\boldsymbol{O}^+$ and $\boldsymbol{E}^+$) and negative testing signatures ($\boldsymbol{O}^-$ and $\boldsymbol{E}^-$). The functionality of the above-mentioned data sets is summarised in Figure 1.4.

In a research environment, a single, unpartitioned data set, containing both positive and negative signatures (which are correctly labelled), across a set of writers, is typically available. A data set containing the signatures of N_w writers, with J positive and negative signatures for each writer is depicted in Figure 1.5a. Each column represents an individual writer, where the symbols "+" and "−" represent positive and negative signatures respectively. Figure 1.5b shows a partitioned data set with the appropriate labels.

The appropriate separation of the optimisation and evaluation sets ensures

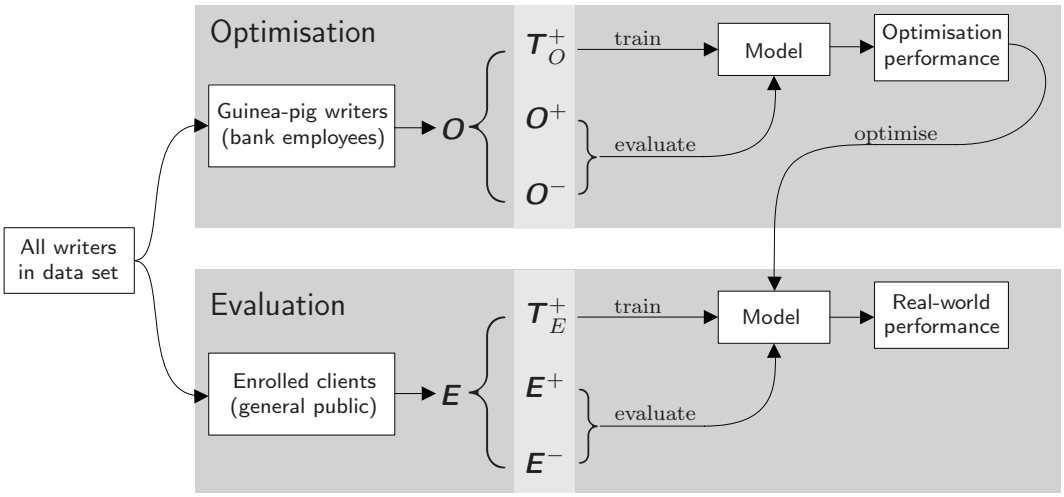


Figure 1.4: Data partitioning. The writers in the data set are divided into optimisation writers and evaluation writers. The notation used throughout this dissertation, as well as the role of each partition, are shown.

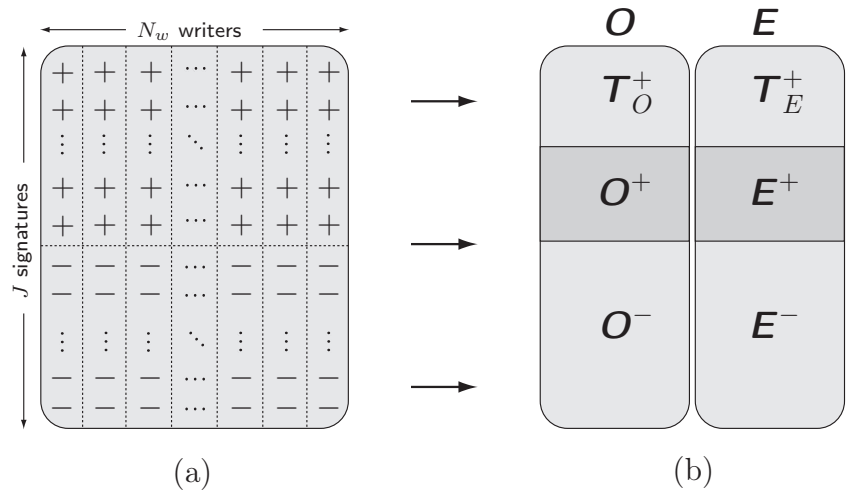


Figure 1.5: Data partitioning. (a) Unpartitioned data set. Each column represents an individual writer. A "+" indicates a positive signature, while a "−" indicates a negative signature. (b) Partitioned data set.

that the reported results provide a reliable indication of the system's generalisation potential (real-world performance). When (inappropriately) optimising the system parameters using the *evaluation* set, it is not possible to detect overfitting, and the results obtained may therefore be optimistically biased (Kuncheva (2004)).

The actual data set used in this dissertation, as well as the partitioning and evaluation protocols, are discussed in more detail in Chapter 9.

1.4.2 Preprocessing and signature zoning

Each signature is represented by a binary image, where 1 represents a pen stroke, and 0 the background. A flexible grid-based zoning scheme is employed in order to define $N_r - 1$ different coordinate pairs for each signature image. Each of these coordinate pairs constitute the centre of a circular local subimage (referred to as a *retina*). A global "retina", which encompasses the entire signature image, is also defined. We therefore define N_r retinas in total for each signature image. Signature zoning is discussed in more detail in Section 3.3.

1.4.3 Discrete Radon transform

The discrete Radon transform (DRT) is subsequently used to extract a preliminary observation sequence from each retina. The DRT is obtained by calculating projections of each signature image from different angles. A number of modifications are subsequently made to each preliminary observation sequence in order to obtain a final, *periodic* observation sequence. The DRT is discussed in more detail in Section 3.4.

1.4.4 Signature Modelling

Observation sequences extracted from the writer-specific positive training signatures are used to initialise and train a continuous, ring-structured HMM for each retina, by implementing Viterbi re-estimation.

The ring-structured topology of the HMM, associated with each global retina, in conjuction with the periodic nature of the observation sequences, ensure that each global HMM is invariant with respect to the rotation of the signature in question (see Chapters 4 and 5 for a detailed discussion).

For each writer, the above-mentioned training signatures are also used to estimate the distribution of typical dissimilarity values, associated with positive signatures. In order to obtain such a dissimilarity value, an observation sequence (extracted from a specific retina) is matched with the corresponding trained HMM though Viterbi alignment, so that a high dissimilarity value is associated with a low confidence of authenticity (see Chapter 6 for a detailed discussion).

The parameters of each writer-specific dissimilarity value distribution is used to normalise the dissimilarity values for each writer, so that they are comparable across writers.

1.4.5 Verification

A retina is henceforth accepted or rejected by imposing a threshold on the normalised dissimilarity value. If this dissimilarity value is less than said threshold, the retina is accepted, otherwise it is rejected (see Chapter 6 for a detailed discussion).

1.4.6 Ensemble generation and selection

A questioned signature is classified as positive (using the majority voting rule) if at least half of the decisions made by the optimal ensemble of selected base classifiers are positive, where each base classifier is associated with a specific retina.

In this dissertation we employ two different ensemble generation techniques in order to produce a pool of candidate ensembles. The performance of each ensemble (where fusion of the decisions of the constituent base classifiers is achieved by majority voting) is evaluated using the optimisation set, after which the most proficient ensemble is selected, based on a specific operating criterion (see Chapter 8 for a detailed discussion).

1.4.7 System outline and flowchart

A basic overview of the proposed system, in pseudocode form, is given below.

1 For each writer w:
  1.1 Define N_r retinas, that include a global retina, for each signature using the zoning procedure outlined in Chapter 3.
  1.2 For each retina r:
    1.2.1 Extract an observation sequence $\mathbf{X}_w^r = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$, using the DRT-based feature extraction technique discussed in Chapter 3.
    1.2.2 Use the relevant training set, $\boldsymbol{T}_O^+$ or $\boldsymbol{T}_E^+$, to train an HMM λ_w^r, as discussed in Chapter 4.
2 Use $\boldsymbol{O}^-$ and $\boldsymbol{O}^+$ to select the best performing ensemble of size N_S, where $1 \leq N_S \leq N_r$, amongst all of the optimisation writers (Chapter 8).
3 Combine the decisions of the selected base classifiers (in 2) using majority voting and the evaluation writers (in $\boldsymbol{E}$), in order to gauge the generalisation potential of the system (Chapters 9 and 10).

A flowchart of the system is provided in Figure 1.6.

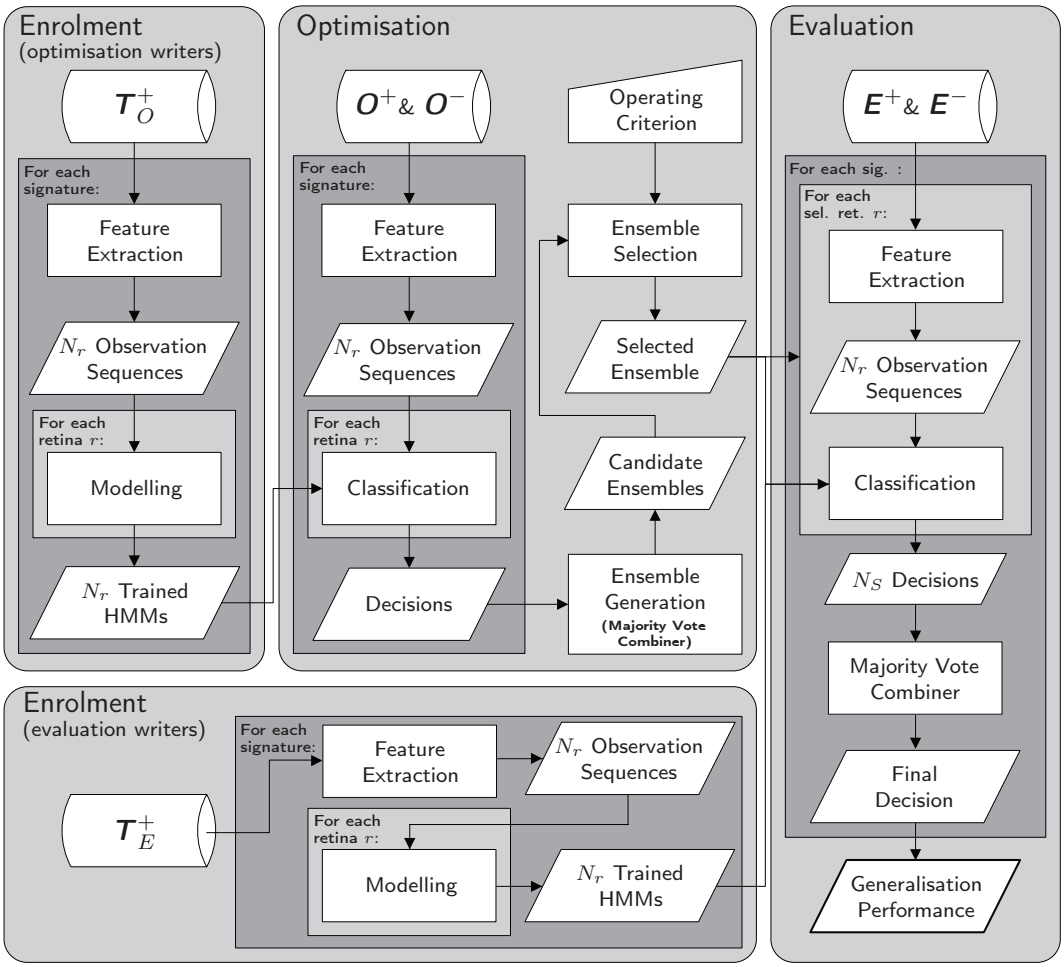


Figure 1.6: System flowchart.

1.5 Results

We show that, by employing the optimal ensemble, where each individual base classifier constitutes a continuous observation HMM that is trained using the DRT-based features extracted from *local* signature regions, an EER of 0.086 (or 8.6%) is achievable. This compares favourably with the system proposed in Coetzer (2005), for which an EER of 12.2% is reported when evaluted on the same data set. Note that the system proposed in Coetzer (2005) is similar to the system developed in this dissertation, with the exception that only global features are considered in Coetzer (2005). We therefore show that the inclusion of local features, when evaluated on the same data set, improves the EER by 33.3%. The results achieved in this dissertation are discussed in detail in Chapter 10.

1.6 Contributions

The major contributions of this dissertation are listed below. The first three items relate to the realisation of the *specific* objectives set out in Section 1.3. The last four item constitute *general* contributions made to the field of pattern recognition/machine learning as a result of this study.

1. **The development of a novel off-line signature verification system**. Since the proposed system utilises significantly different features and/or modelling techniques than those employed in existing systems, that were evaluated on *different* data sets than the one considered in this dissertation, it is very likely that a superior combined system can be obtained by combining the proposed system with any of the aforementioned systems.

2. **The development of a proficient off-line signature verification system.** When the performance of the proposed system is compared to the performance of existing systems (see Dolfing (1998), Coetzer *et al.* (2004) and Swanepoel and Coetzer (2010)), it is clearly demonstrated that, when evaluated on the *same* data set (that contains high-quality imitations), the proposed system is significantly superior (see Chapter 10).

3. **The benefit of also utilising local features.** When the performance of the proposed system (that utilises global *and* local features) is compared to the performance of an existing system (see Coetzer (2005)), that utilises similar features—but only global features and no local features—it is clearly demonstrated that, when evaluated on the *same* data set (that contains high-quality imitations), the inclusion of local features, together with classifier combination, leads to a significant increase in system performance (see Chapter 10).

4. **Rotational invariance.** A novel strategy for ensuring rotational invariance amongst signatures belonging to the same writer is proposed. The proficiency and robustness with respect to various noise intensities, variations in pen stroke-width, and variations in aspect ratio of said strategy is clearly demonstrated (see Chapter 5).

5. **Score normalisation.** A threshold parameter calibration strategy, that reconciles the operational criteria (as imposed by an operator, like a bank manager) with the practical constraints (that is, the abundance/scarcity, and class label(s) of the available training data) is introduced and demonstrated. This issue is rarely addressed in the existing literature (see Chapter 7).

6. **Efficiency-cautious ensemble selection.** Instead of utilising a brute force approach, like an exhaustive search (which is computationally infeasible) or a genetic search algorithm (see Mitchell (1998)), a more subtle approach of selecting optimal ensembles along several radial lines in ROC space is introduced and demonstrated. This approach leads to a considerable reduction in the computational requirements during the optimisation stage. Since a genetic search algorithm is not implemented in this dissertation (and is considered to fall outside the scope of this project), a possible gain in performance, by utilising the aforementioned strategy, is not investigated. This is considered to be part of possible future work (see Chapter 8).

7. **Bias-cautious performance evaluation in multi-iteration experiments.** In multi-iteration experiments (where disjoint optimisation and evaluation data sets are employed), for example in k-fold cross-validation (with or without shuffling), the traditional approach (adopted in the existing literature) entails the simple averaging of the reported performance across all of the experimental iterations for a certain imposed operational criterion. This approach is optimistically biased−more markedly so in cases where the evaluation data is scarce. This is especially evident in the leave-one-out scenario, for which the optimistic bias can be clearly demonstrated by drawing attention to the connection between the aforementioned scenario and score normalisation. An alternative performance evaluation protocol is therefore adopted and implemented in this dissertation. Given a certain imposed operational requirement, for example a desired error rate, this performance evaluation protocol also takes the accuracy and precision of the attained operational error rate (obtained on the evaluation set) across all of the experimental iterations into account.

1.7 Layout of dissertation

The dissertation is presented as follows.
Chapter 2: Related Work. We discuss some recent work on off-line signature verification focussing specifically on systems proposed in the literature that utilise similar feature extraction or modelling techniques. Signature verification systems that employ classifier combination are also discussed.
Chapter 3: Feature Extraction. We introduce the zoning strategy employed in this dissertation, and explain how the DRT is used to extract features from each local retina (as well as from the global "retina") defined during the zoning process.
Chapter 4: Signature Modelling. We introduce the HMM topology employed in this dissertation and discuss how each retina can be modelled using a ring-structured, continuous observation HMM.

Chapter 5: Invariance and Normalisation Issues. We address issues relating to scale, translation and rotation invariance of the signature verification system developed in this dissertation. A novel algorithm for normalising the rotation of each signature is also introduced.
Chapter 6: Verification and Performance Evaluation. We introduce the concept of thresholding and discuss the performance evaluation measures utilised in this dissertation.
Chapter 7: Score Normalisation. We introduce the concept of score normalisation and present a brief overview of some of the strategies employed in the literature. We also introduce the concept of threshold parameter transformation.
Chapter 8: Ensemble Selection and Classifier Combination. We present a brief overview of ensemble selection and combination before discussing the ensemble generation and selection strategies employed in this dissertation.
Chapter 9: Data and Experimental Protocol. The data partitioning protocol utilised in this dissertation, as well as the data set, are discussed in detail in this chapter. The performance evaluation protocol is also specified.
Chapter 10: Results. In this chapter, the results achieved for the system developed in this dissertation are presented and discussed.
Chapter 11: Conclusion and Future Work. We consider possible future work, as well as a number of outstanding issues.

Chapter 2

Related Work

2.1 Introduction

In this chapter we discuss some relevant, existing off-line signature verification systems. The literature on off-line signature verification is expansive. We do not, therefore, aim to provide a comprehensive literature study here, and only provide a brief summary of signature verification systems that are *related* to the work presented in this dissertation by either the feature extraction technique utilised (DRT or projection-based features) or the modelling technique employed (HMMs). We also discuss several recent signature verification systems that utilise classifier combination techniques.

Since most existing off-line signature verification systems are not evaluated on the same data set that is utilised in this dissertation (that is, Dolfing's data set), we are unable to directly compare our system's results with the results reported by most other authors. Several systems have, however, *also* been evaluated on Dolfing's data set and we include a detailed analysis of the results reported for each of these systems.

Since the system developed in this dissertation improves upon the system proposed in Coetzer *et al.* (2004), we first discuss Coetzer's system in detail (Section 2.2), and then discuss other relevant systems proposed in the literature based on the modelling technique (Section 2.3) and feature extraction strategy (Section 2.4) utilised.

Since human classifiers provide an important benchmark against which to measure the proficiency of an automated system, we discuss research aimed at investigating the proficiency of a typical human being at verifying the authenticity of a signature in Section 2.5.

Related work that utilises classifier combination is detailed in Section 2.6.

2.2 Coetzer's system

In Coetzer (2005) two off-line signature verification systems are proposed: an HMM-based system and a dynamic time warping (DTW)-based system. Both systems utilise features based on a modified DRT. The final observation sequence in each case is periodic. The HMM-based system utilises a ring-structured HMM, which, when used in conjuction with periodic observation sequences, ensures that the system is invariant with respect to the rotational orientation of each questioned signature. The DTW-based system uses DTW techniques to match each observation with a template created for each writer's signature. When evaluated on the "Stellenbosch data set", both the HMM-based system and the DTW-based system achieve an EER of approximately 18%. This suggests that the performance of the two systems are equivalent, although the HMM-based system is significantly more computationally efficent.

As is the case for the system developed in this dissertation, the HMM-based system proposed in Coetzer (2005) is also evaluated on Dolfing's data set, and an EER of 12.2% *is reported. This EER provides a very important benchmark against which to compare the performance of the system developed in this dissertation, since the feature extraction and modelling techniques are similar, except that in Coetzer (2005), only a global retina (that is, global features) is considered.*

2.3 Modelling techniques

In this section we discuss a few modelling techniques commonly used in signature verification systems.

Support vector machines (SVMs) were originally developed by Vapinik (1998). An SVM uses a set of training examples, each labelled as belonging to one of two classes, to train a model that predicts to which class an unknown sample belongs. In the context of signature verification, both genuine signatures and forgeries are required for *each* writer to train the SVM. SVMs are therefore generally limited to signature verification systems aimed at detecting only random or casual forgeries. Examples of off-line signature verification systems that utilise SVMs include Bortolozzi and Sabourin (2004), Martinez *et al.* (2004) and Ozgunduz *et al.* (2005). In Ozgunduz *et al.* (2005), however, skilled forgeries are (unrealistically) assumed to be available for each writer for the purpose of training the SVM.

A neural network (NN) is a non-linear statistical data modelling tool that is used to model complex relationships between input and output data. Neural Networks have been used extensively for off-line signature verification (see Velez *et al.* (2003) and Armand *et al.* (2006)).

Many signature verification systems have been developed that utilise template matching techniques (for example, DTW) instead of a discriminative or

generative model. Examples of systems utilising DTW can be found in Deng *et al.* (1999), Guo *et al.* (2001) and Fang *et al.* (2003).

2.3.1 Hidden Markov models

An HMM (see Rabiner (1989)) is an example of a generative stochastic model, well suited for representing temporal data, such as human speech, handwriting and dynamic signature data. HMMs have, however, also been successfully utilised in many off-line signature verification systems. We provide a few examples below.

El-Yacoubi *et al.* (2000) describes an off-line signature verification system based on HMMs. Each signature image is segmented into local square regions by superimposing a grid on said image. The pixel density of each cell is calculated and constitutes a feature. Each feature vector represents the pixel densities of a column of cells. Multiple classifiers are trained using grids with different resolutions. The decisions of the individual classifiers are combined using majority voting. Since the system was evaluated using only random forgeries, average error rates of less than 1% are reported.

In Justino *et al.* (2001), the above system is improved by extracting several features from each cell. In addition to the pixel density, the pixel distribution and axial slant of each cell are also computed. The system is evaluated using random, casual and skilled forgeries.

The above system is further improved in Swanepoel and Coetzer (2010), in which a flexible grid is used. Each cell is dilated so that overlapping cells are defined. The optimal degree of overlap is determined using an optimisation set. Pixel density, pixel slant and pixel distribution features are extracted from each cell. Features obtained from each column of cells therefore constitute a feature vector. Individual discrete observation HMMs are trained for each feature type. Two classifier combination strategies are investigated, namely majority voting (decision-level fusion) and score averaging (score-level fusion). The system is optimised and evaluated on Dolfing's data set, and an EER of 10.23% is reported for the majority voting system, while an EER of 11.21% is reported for the score averaging system. Since these results are also obtained using Dolfing's data set, a direct comparison with the results reported for our system is possible.

Hidden Markov models have also been utilised in Batista *et al.* (2009) and Mehta *et al.* (2010).

2.4 Features

Projection-based feature extraction techniques, similar to the DRT-based technique employed in this dissertation are also used in, for example, Deng *et al.* (1999), El-Yacoubi *et al.* (2000), Fang *et al.* (2001), Baltzakis and Papamarkos

(2001), Fang *et al.* (2002) and Mizukami *et al.* (2002). In addition to Coetzer *et al.* (2004), the discrete Radon transform has also been utilised in Shapiro and Bakalov (1993) and Maiorana *et al.* (2007).

2.5 Human classifiers

Despite being the standard against which all off-line signature verification systems should be compared, the proficiency of a typical human at classifying off-line signatures has not been studied extensively.

In Coetzer (2005) and Coetzer and Sabourin (2007), experiments are conducted on human classifiers using Dolfing's data set. Faculty members, graduate students and departmental secretaries at Stellenbosch University constituted the test group.

Each individual was provided with a training set containing fifteen signatures, and a corresponding test set also containing fifteen signatures, for all fifty-one writers in Dolfing's data set. Each test set contained a randomly selected number (between zero and fifteen) of skilled forgeries. The remaining signatures in each set were genuine, and were randomly selected from the test signatures for each writer.

Most individuals spent approximately 3.5 to 4.7 seconds classifying each signature. Of the twenty-two individuals who participated in the experiment, only four performed better than the HMM-based system developed in Coetzer *et al.* (2004). The system developed in this dissertation outperforms each of the above-mentioned twenty-two human beings.

2.6 Classifier combination

In addition to Swanepoel and Coetzer (2010), El-Yacoubi *et al.* (2000) and Justino *et al.* (2001), classifier combination techniques (sometimes referred to as the multi-hypothesis approach to pattern recognition) have also been utilised in the following papers.

In Bertolini *et al.* (2008) and Bertolini *et al.* (2009), a graphometric feature set is extracted that considers the curvature of the main strokes of a signature. An optimal ensemble of classifiers is built using a standard genetic algorithm and different fitness functions are assessed to drive the search. It is shown that the combined classifier is more proficient than the constituent classifiers when considered individually.

In Batista *et al.* (2009), an approach based on the combination of discrete Hidden Markov models (HMMs) in the ROC space is proposed. The system selects, from a set of different HMMs, the most suitable solution for a given input sample. Experiments performed using a real-world off-line signature verification data set, with random, casual and skilled forgeries, indicate that

the multi-hypothesis (classifier combination) approach can reduce the average error rates by more than 17%. This is consistent with the system developed in this paper, in which the EER is reduced by 33.3% when classifier combination is employed.

2.7 Conclusion

A brief synopsis of related off-line signature verification systems has been provided. Since no standardised signature data set is available on which all systems are evaluated, a direct comparison of performances achieved is generally not possible. A detailed comparison of the results achieved for several systems that were evaluated on the *same* data set utilised in this dissertation is provided in Section 11.1.

Chapter 3

Image Processing, Zoning and Feature extraction

3.1 Introduction

In this chapter we focus on the process utilised by the system developed in this dissertation for extracting features from a static, binary image of a signature. We assume that said signature has already been successfully extracted from the document background. Features are extracted both globally as well as from *local* regions within the signature. The zoning scheme, that is, the process of dividing a signature into local regions, is discussed in Section 3.3. Section 3.4 describes the DRT, while the specific implementation of the DRT, used to generate observation sequences, is discussed in Section 3.5.

3.2 Signature preprocessing

Each signature is represented by a binary image, where 1 represents pen strokes and 0 represents the background. In binary image processing, black typically represents 0, and white represents 1. This convention is reversed in this dissertation, so that a signature is shown in black on a white background. The signature image is assumed to be free of noise and any other background artefacts. The largest rectangular dimension of the signature image is rescaled to 512 pixels, while maintaining the aspect ratio. Examples of typical signature images are shown in Figure 3.1.

The data set used in this dissertation was originally captured on-line for Hans Dolfing's Ph.D. thesis (Dolfing (1998)). The dynamic signatures are converted into static signature images using only the pen position data. The signature images therefore contain no background noise, or variation in pen stroke-width; in this sense they are ideal. See Coetzer (2005) for a detailed discussion on the process of rendering static signature images from dynamic data. The signatures are therefore ideal in the sense that they have been ex-

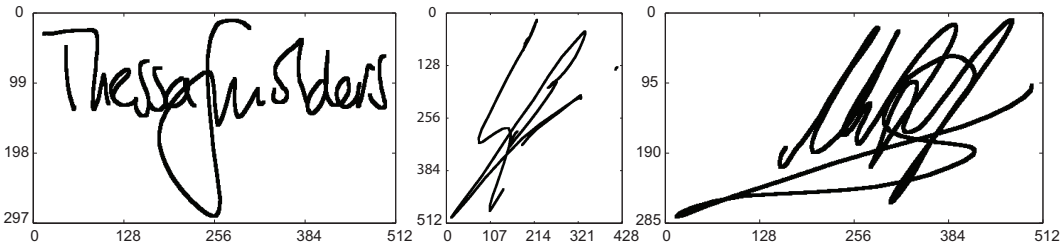


Figure 3.1: Examples of typical signatures belonging to three different writers. The largest rectangular dimension of each signature is 512 pixels.

tracted successfully from the document background. Strategies for extracting signatures from bank cheques, historic documents, and legal documents are well-documented (see Madusa *et al.* (2003)) and fall outside the scope of this dissertation. Noise removal may be achieved by applying an adaptive median filter to the signature (see Gonzalez and Woods (2002)). A more detailed discussion on the signature data set used in this dissertation can be found in Chapter 9.

The consistent rotational orientation of signatures belonging to the same writer is essential to the success of the zoning scheme described in the next section. Since the normalisation of each signature's rotational orientation is achieved by utilizing a trained HMM, the discussion of this normalisation process is necessarily postponed until Chapter 5. For the remainder of this chapter, consistent rotational orientation of all signatures belonging to the same writer is assumed.

3.3 Signature zoning

3.3.1 Background

Signature zoning is the process of dividing a signature into regions, primarily to define areas from which local features can be extracted. A number of zoning schemes have been described in the literature, and can be broadly classified as either *signal-dependent* or *fixed*, as discussed in Bastista *et al.* (2007).

Fixed zoning schemes define constant regions for *all* signatures, independent of any characteristic of the signature being processed. Fixed zoning schemes typically use vertical or horizontal strips of a fixed width or height, although radial-based zoning schemes have also been used. A combination of both vertical and horizontal strips may be used to form a grid layout. Examples of systems employing grid-based zoning can be found in Ferrer *et al.* (2005), Armand *et al.* (2006), Justino *et al.* (2001) and Swanepoel and Coetzer (2010).

Signal-dependent zoning schemes define different regions for each signature, based on one or more characteristics of the signature being processed.

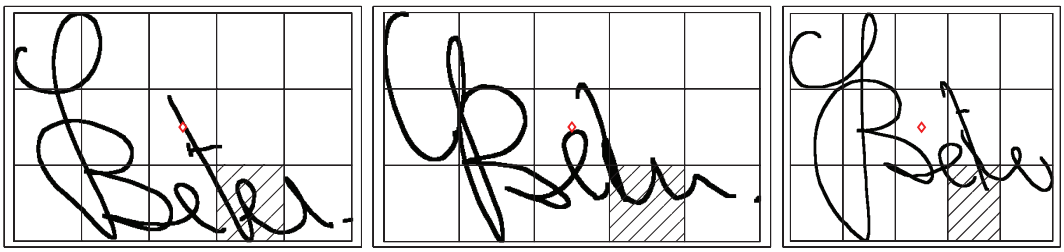


Figure 3.2: Rigid grid-based zoning illustrated using three positive samples of the same writer. The bounding box of each signature is uniformly divided into horizontal and vertical strips. The geometric centre of each bounding box is indicated with a ⋄.

A flexible, grid-based zoning scheme is employed in this dissertation. This strategy is discussed in Section 3.3.3, but a few key concepts are introduced in the next section, where rigid, grid-based zoning schemes are introduced.

3.3.2 Rigid grid-based zoning

A simple implementation of a rigid grid-based zoning scheme is shown in Figure 3.2, using three positive samples belonging to the same writer. A bounding box is defined around the signature and said box is divided into a number of vertical and horizontal strips of uniform width and height. This scheme is fixed in the sense that every signature is divided into a grid containing the same number of cells, however, it can still be considered signal-dependent, as the width and height of the strips are dependent on the size of the signature's bounding box.

By examining the shaded regions in Figure 3.2, the inadequacy of this basic zoning scheme becomes evident, since *different* regions of the signatures are zoned as *corresponding regions* (see shaded grid cells). Ideally, all positive samples belonging to the same writer should be zoned in such a way that corresponding zones contain corresponding regions of the signature.

An improvement to the basic grid-based zoning scheme is shown in Figure 3.3. The bounding box is again used to divide each signature into uniform vertical and horizontal strips, but the centre of the grid is now translated to correspond to the gravity centre of the signature (indicated by "○"). If, for example, the shaded regions are again considered in Figure 3.3, it is clear that centring the grid on the gravity centre of the signature improves upon the basic scheme, as the shaded zones now represent similar regions of each signature, namely the lower section of the letters "*et*". A comparison of Figures 3.2 and 3.3 shows that this improvement is evident for all zones.

In conclusion, the gravity centre of a signature is more stable than the geometric centre, and should therefore be preferred as a reference point when zoning a signature. The actual zoning scheme used in this dissertation, which is a more adaptive version of the above grid-based zoning scheme, is discussed

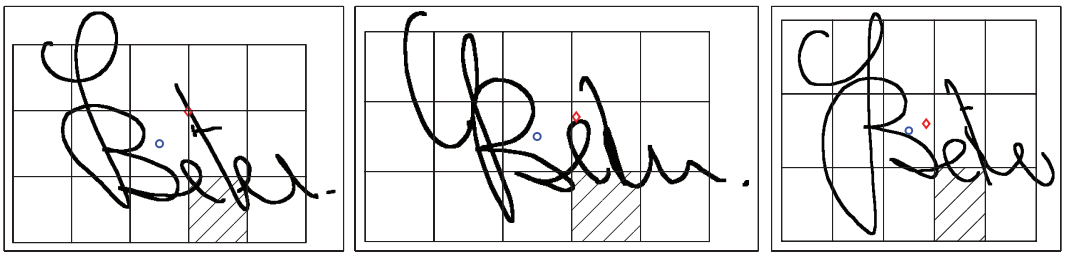


Figure 3.3: Grid-based zoning illustrated using three positive samples of the same writer. The grid constructed in Figure 3.2 has now been translated to align with the gravity centre of each signature. The gravity centre of each signature is indicated with a ○. The geometric centre (◇) is shown for reference.

in the next section.

3.3.3 Flexible grid-based zoning

The system developed for this dissertation uses overlapping, circular retinas. The purpose of zoning in this context therefore differs from how zoning is typically used. The standard grid-based zoning scheme can be modified by defining a point at the intersection of each pair of vertical and horizontal dividing lines. These points are then used as centroids for circular retinas.

We therefore aim to zone the signature so that said centroids are located in similar regions across all signatures belonging to the same writer. The concentration of retinas on denser areas of the signature, as well as the avoidance of retinas containing no signature information, is also desirable.

The flexible grid-based zoning scheme calculates the location of the vertical and horizontal dividing lines based on the percentage of the total number of black pixels contained in each strip. We define two parameter sets, Z_h and Z_v, which contain the intervals for the horizontal and vertical divisions respectively. The gravity centre, $\boldsymbol{G}_\mu = (G_x, G_y)$, is used as a reference point. The zoning process is best explained by illustration. First, the signature image is divided vertically into two sections at G_x, as shown in Figure 3.4. Each section is then divided into strips, based on the percentage of black pixels defined in Z_v. For the example shown in Figure 3.4, $Z_v = \{20, 60\}$. The first strip therefore contains 20% of the total number of black pixels in the respective section, while the second division contains 60% of the total number of black pixels. The same process is used to create horizontal divisions, as shown in Figure 3.5. In this example, $Z_h = \{33\}$.

The sample signatures shown in Figures 3.2 and 3.3 are repeated in Figure 3.6 with a flexible grid-based zoning scheme applied.

Further examples of the flexible grid-based zoning scheme, with $Z_v = \{0, 40, 95\}$ and $Z_h = \{0, 60\}$, are shown in Figure 3.7. Note that the retina cen-

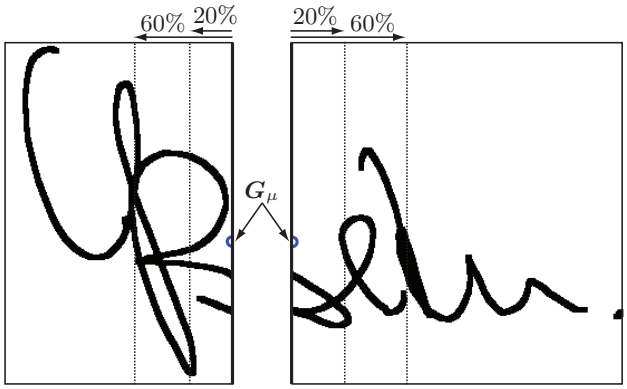


Figure 3.4: Flexible grid-based zoning illustrated for $Z_v = \{20, 60\}$. The signature is vertically divided at G_x into two sections. Each section is then zoned based on the values in Z_v.

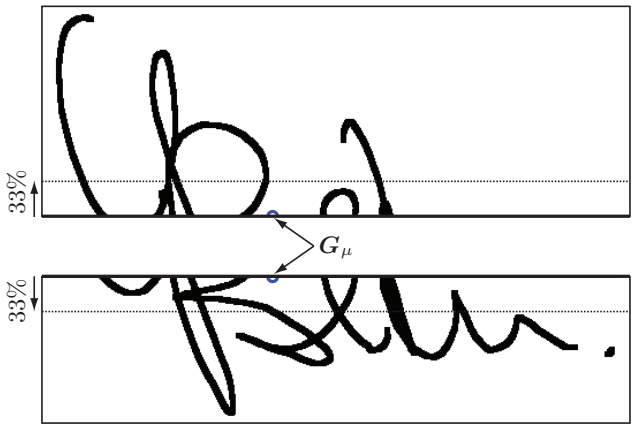


Figure 3.5: Flexible grid-based zoning illustrated for $Z_h = \{33\}$. The signature is horizontally divided at G_y into two sections. Each section is then zoned based on the values in Z_h.

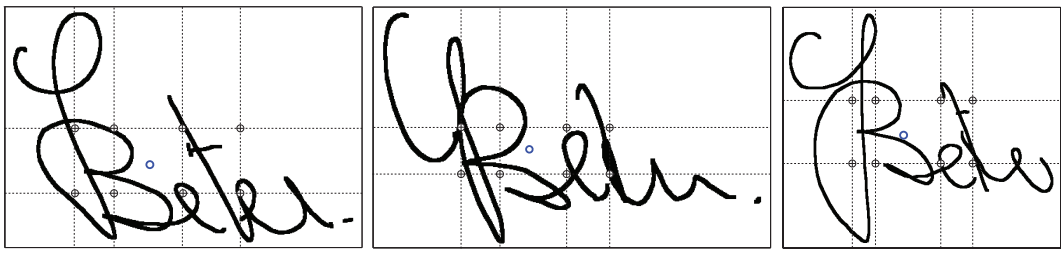


Figure 3.6: Flexible grid-based zoning, with $Z_v = \{20, 60\}$ and $Z_h = \{33\}$, illustrated using three positive samples belonging to the same writer. The gravity centres ($\circ$) are shown for reference. The $\oplus$ symbol indicates the location of the retina centroids.

Figure 3.7: Examples of flexible grid-based zoning applied to signatures belonging to four different writers. In these examples, $Z_v = \{0, 40, 95\}$ and $Z_h = \{0, 60\}$. The centroid locations are indicated by the $\oplus$ symbol.

troids are concentrated on areas of the signature image that contain significant signature information.

3.3.4 Retina construction

The points defined at the intersection of each pair of dividing lines are used as centroids to construct $N_r - 1$ circular retinas with radii γ, as shown in Figure 3.8. The number of retinas (including the global retina), N_r, is determined by the lengths of Z_h and Z_v, that is

$$N_r = 2|Z_h| \times 2|Z_v| + 1, \tag{3.3.1}$$

or, if Z_h and Z_v include a zero-valued entry,

$$N_r = (2|Z_h| - 1) \times (2|Z_v| - 1) + 1. \tag{3.3.2}$$

The "+1" term in each of the above equations accounts for the global retina.

3.4 The discrete Radon transform

The Radon transform is an integral transform consisting of integrals of a function over straight lines. The DRT of an image (or any matrix) is calculated by taking the d-dimensional projection of the image from N_θ equally distributed angles, ranging from 0° to 180°. The DRT is therefore a $N_\theta \times d$ image, where each column of the DRT image represents the projection of the original image at a certain angle. The DRT, when converted to a greyscale image, is known as a sinogram. A typical signature and its sinogram are shown in Figure 3.10.

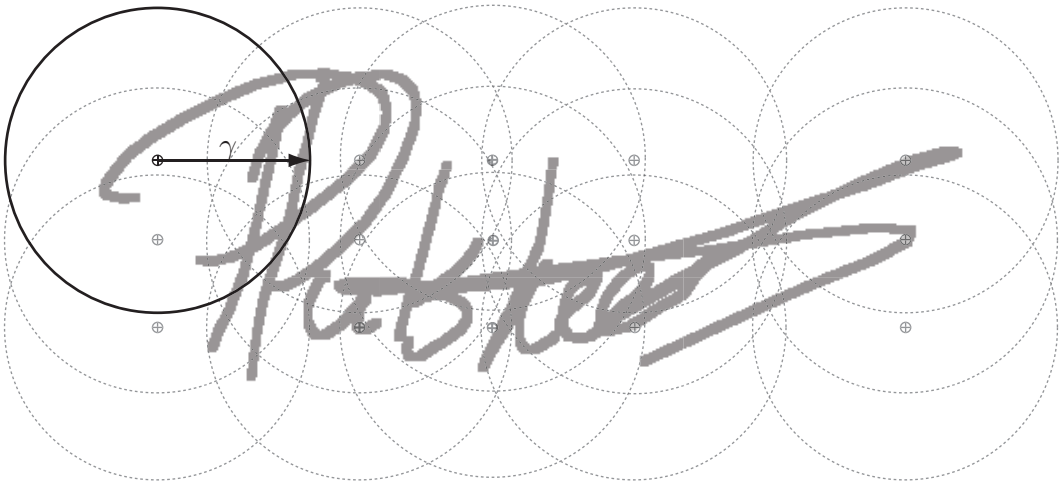


Figure 3.8: A signature image showing the location of fifteen retinas with radii γ. Each circular retina is centred on a centroid defined in the zoning process.

The basic algorithm for the calculation of the DRT follows. A more detailed discussion on the theory and implementation of the DRT can be found in Toft (1996) and Coetzer (2005).

The DRT of an image consisting of Ξ pixels, where the intensity of the ith pixel is denoted by I_i, for $i = 1, \ldots, \Xi$, is calculated using d non-overlapping beams per angle, and N_θ angles in total. The jth beam-sum, which is the cumulative intensity of the pixels that are within the jth beam, is denoted by R_j, where $j = 1, \ldots, N_\theta \cdot d$. The DRT can therefore be expressed as

$$R_j = \sum_{i=1}^{\Xi} \alpha_{ij} I_i, j = 1, 2, \ldots, N_\theta \cdot d. \tag{3.4.1}$$

The symbol α_{ij} denotes the weight of the contribution of the ith pixel to the jth beam-sum, as shown in Figure 3.9. The values N_θ (the number of angles) and d (the number of beams per angle) determine the accuracy of the DRT.

3.5 Observation sequence generation

An observation sequence is a set of feature vectors obtained during feature extraction. The system designed in this dissertation uses both local and global features. Each signature is therefore modelled using N_r observation sequences, where N_r is the number of retinas used. The dimension and length of each observation sequence is determined directly from the parameters d and N_θ used in calculating the DRT.

The parameter d, which is the number of beams used to calculate the DRT, is equal to 2γ, where γ is the radius of the retina from which the DRT is

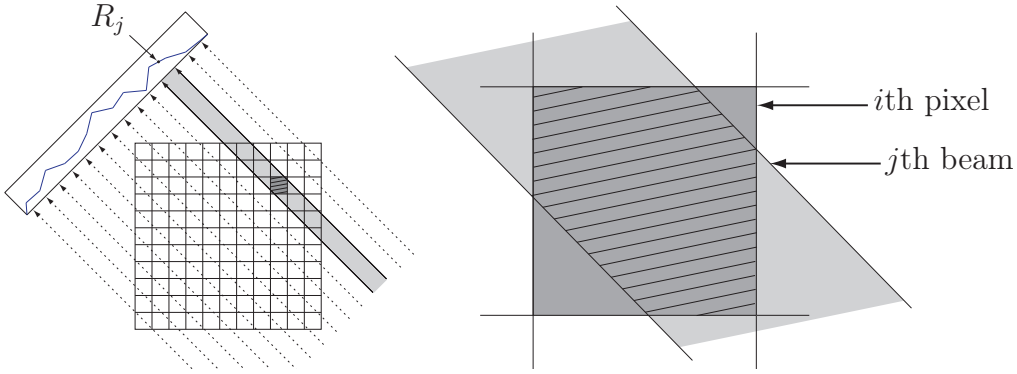


Figure 3.9: The DRT model of Eq. 3.4.1. In this case, α_{ij}, that is, the weight of the contribution of the ith pixel to the jth beam-sum is approximately 0.6 (indicated by the patterned region). This means that the jth beam overlaps 60% of the ith pixel.

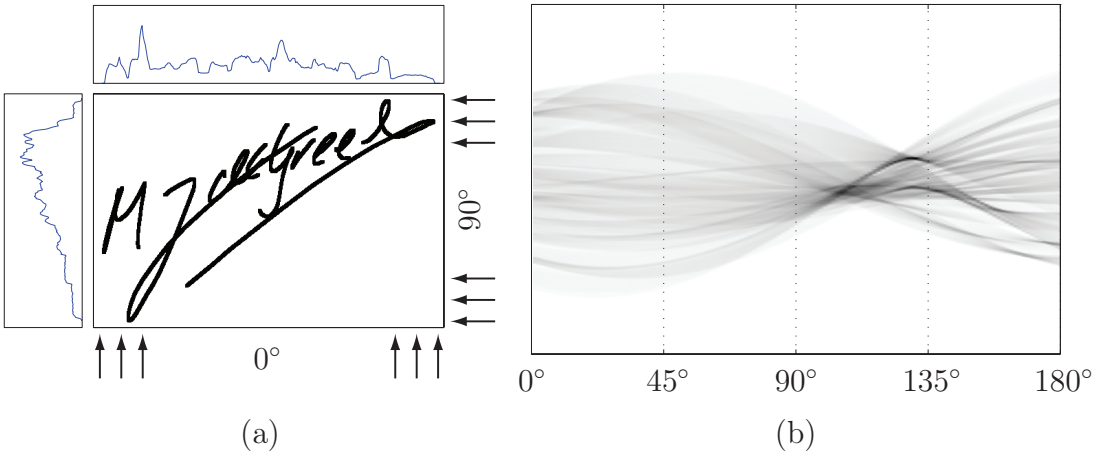


Figure 3.10: The DRT. (a) A typical signature and its projections at $0°$ and $90°$. (b) The sinogram of the signature in (a).

calculated. The parameter d determines the length of each column of the DRT, and therefore the *dimension* of each feature vector. The parameter N_θ, which is the number of angles used to calculate the DRT, determines the length of the preliminary observation sequence.

Each of the N_r observation sequences is derived directly from the DRT of the corresponding retina image. Several modifications are made to each DRT before the final observation sequence is obtained. The DRT can therefore be considered a *preliminary observation sequence*.

All zero-values are decimated from each column of the DRT, and each column is subsequently rescaled to have length d, using linear interpolation. This is done primarily to ensure scale and translation invariance. Invariance issues are discussed further in Chapter 5. This process is illustrated in Figure 3.11.

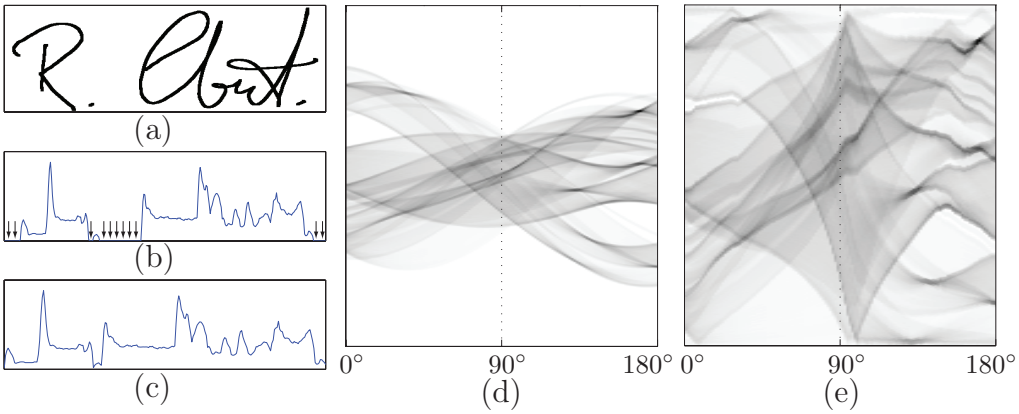


Figure 3.11: Observation sequence construction. (a) Signature image (global retina). (b) The projection calculated from an angle of $0°$. This projection constitutes the first column of the image in (d). The arrows indicate zero-values. (c) The projection in (b) after zero-value decimation and subsequent stretching. This vector constitutes the first column of the image in (e).

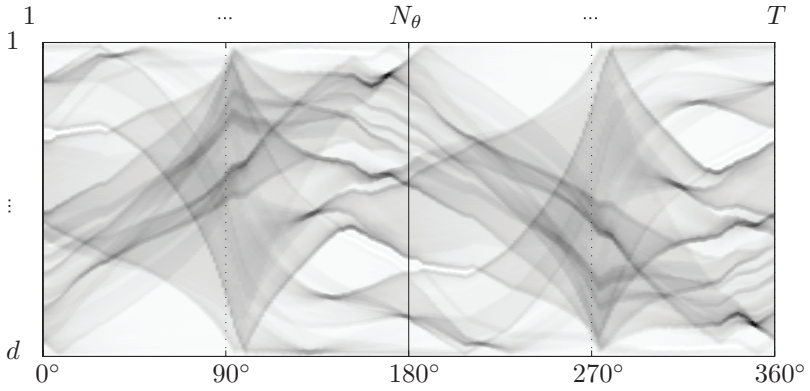


Figure 3.12: Final (global) observation sequence extracted from the entire signature image shown in Figure 3.11a. The complete observation sequence is obtained from the image depicted in Figure 3.11e by appending its horizontal reflection (this is equivalent to the projections obtained from the angle range $180° - 360°$).

Although the projections at angles ranging from $180°$ to $360°$ contain no additional information over the projections at angles ranging from $0°$ to $180°$, these additional angles are appended to the preliminary observation sequence, as shown in Figure 3.12. This is done to aid the construction of a rotation invariant system, the justification of which can be found in Chapter 5. The additional projections are simply the reflections of the projections which form the preliminary observation sequence; no additional calculations are necessary. The final observation sequence therefore has length $T = 2N_\theta$.

The final modification made to the observation sequence is the normalisation of each feature vector by the standard deviation of the intensity of the

entire set of T feature vectors.

3.6 Concluding remarks

Although all the examples in the previous section pertain to the extraction of features from the entire signatures image (global features), it is worth emphasising that the same procedure is applied to each of the $N_r - 1$ local retinas, as well as the global "retina". For each signature, N_r observation sequences $\mathbf{X}_w^r$ for $r \in \{1, ..., N_r\}$ of length T, where $\mathbf{X}_w^r$ denotes the observation sequence extracted from retina r, belonging to writer w, are extracted.

It is worth noting that the feature extraction techniques used in this dissertation are identical to the techniques used in Coetzer (2005), with the exception that in this dissertation feature extraction takes place on both the global and local level.

In this chapter we discussed feature extraction, that is the process of obtaining features from a raw signature image. In the next chapter, we look at how a model is created for each retina.

Chapter 4

Signature Modelling

4.1 Introduction

In this chapter we focus on the technique used in this dissertation to model each client's signature. As we employ both local and global features, each signature is modelled by multiple, distinct HMMs, that is one HMM for each of the N_r observation sequences extracted from each of the N_r retinas defined in Chapter 3. Each of these HMMs will constitute a candidate base classifier within the ensembles constructed in Chapter 8.

In Section 4.2 a brief general overview of HMMs is given. The notation pertaining to HMMs, which is used throughout this dissertation, is provided in Section 4.3. Section 4.4 introduces the HMM topology used in this dissertation, while Section 4.5 discusses the HMM training protocol.

4.2 HMM overview

An HMM is an example of a generative stochastic model used to model structured data, that is an observation sequence, as well as the relationship between the observations. HMMs therefore assume time-evolution, and for this reason, they are especially well suited to represent temporal data, such as human speech, handwriting and dynamic signature data. HMMs can, however, also be successfully utilised in off-line (static) signature verification systems where no dynamic variable is present. By extracting features like those based upon the DRT, time-evolution can be simulated. In this system, each feature vector, or observation, represents a projection of the signature (or retina) at a different angle. The angle can therefore be considered the dynamic variable.

A comprehensive tutorial on HMMs can be found in Rabiner (1989).

4.3 HMM notation

A continuous observation sequence of length T is denoted by

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}, \tag{4.3.1}$$

where $\mathbf{x}_i$, for $i = 1, 2, \ldots, T$ denotes the ith feature vector in the sequence. The following notation is used for a continuous first order HMM λ:

1. The N individual states are denoted by

$$\mathbf{S} = \{s_1, s_2, \ldots, s_N\}. \tag{4.3.2}$$

   The state at time t is denoted by q_t.

2. The symbol $\boldsymbol{\pi} = \{\pi_i\}$ denotes the initial state distribution, where

$$\pi_i = P(q_1 = s_i), i = 1, \ldots, N. \tag{4.3.3}$$

3. The state transition probability is denoted by $\mathbf{A} = \{a_{i,j}\}$, where

$$a_{i,j} = P(q_{t+1} = s_j | q_t = s_i), i, j = 1, \ldots, N. \tag{4.3.4}$$

4. The similarity between a feature vector $\mathbf{x}$ and the state s_j, which is represented by a multivariate PDF, is denoted by the likelihood

$$f(\mathbf{x}|s_j, \lambda), j = 1, \ldots, N. \tag{4.3.5}$$

5. The similarity between an observation sequence $\mathbf{X}$ and an HMM λ, is denoted by the likelihood

$$f(\mathbf{X}|\lambda). \tag{4.3.6}$$

6. The dissimilarity between an observation sequence $\mathbf{X}$ and an HMM λ is expressed as follows

$$D(\mathbf{X}|\lambda) = -\ln(f(\mathbf{X}|\lambda)). \tag{4.3.7}$$

7. The HMM which models retina r belonging to writer w is denoted by λ_w^r.

4.4 HMM topology

The two most commonly utilised HMM topologies are the ergodic and left-to-right models. An ergodic HMM represents the most general and flexible model, since every state is reachable from every other state, that is $a_{i,j} > 0$ for $i, j = 1, \ldots, N$, and the probability of entering each state is nonzero, that is $\pi_i > 0$ for $i = 1, \ldots, N$. No restrictions are therefore placed on the initial state and state transition probabilities. This topology has been used extensively to model various pattern structures, recent applications of which can be found in Einsele *et al.* (2008) and You *et al.* (2010).

Various other HMM topologies are made possible by placing restrictions on the state transition probabilities and initial state distribution. In a left-to-right HMM (see Figure 4.1a), for example, $a_{i,j} = 0$ for $i > j$, indicating that the states must be traversed from left to right. This topology is popular for modelling speech and dynamic and static handwriting (see Varga and Moore (1990)).

The system developed in this dissertation represents each observation sequence with a ring-structured HMM with N states. A ring-structured HMM is similar to a left-to-right HMM, with the addition that the first state (s_1) may also be reached from the last state (s_N), that is $a_{N,1} > 0$. An example of a ring-structured HMM is shown in Figure 4.1b.

Using periodic, DRT-based observation sequences, together with a ring-structured HMM ensures that each model is a rotation invariant representation of a signature through 360°.

It may, however, be argued that the specific rotational orientation of a writer's signature is an important characteristic of the signature, and should therefore be included in the model. By manipulating the initial probability distribution, this system can easily be adapted to achieve this. For example, by setting $\pi_1 = 1$, no variation in rotation is tolerated. If, for example, in the eight state HMM shown in Figure 4.1b, only a 45° tolerance in rotational orientation is permitted, than the initial state probabilities can be set so that $\pi_{1,2,8} = 1/3$. Issues concerning translation, rotation and scale invariance of signatures are addressed in detail in Chapter 5.

4.5 Training

A continuous observation HMM λ consists of three sets of hyper-parameters, namely, the initial state distribution $\boldsymbol{\pi}$, the state transition probability distribution $\mathbf{A}$, and a PDF representing each of the N states. For an HMM to accurately model a set of observation sequences extracted from a number of training signatures, it is required that said three parameter sets are appropriately trained to fit the training data. In order to achieve this, the parameter

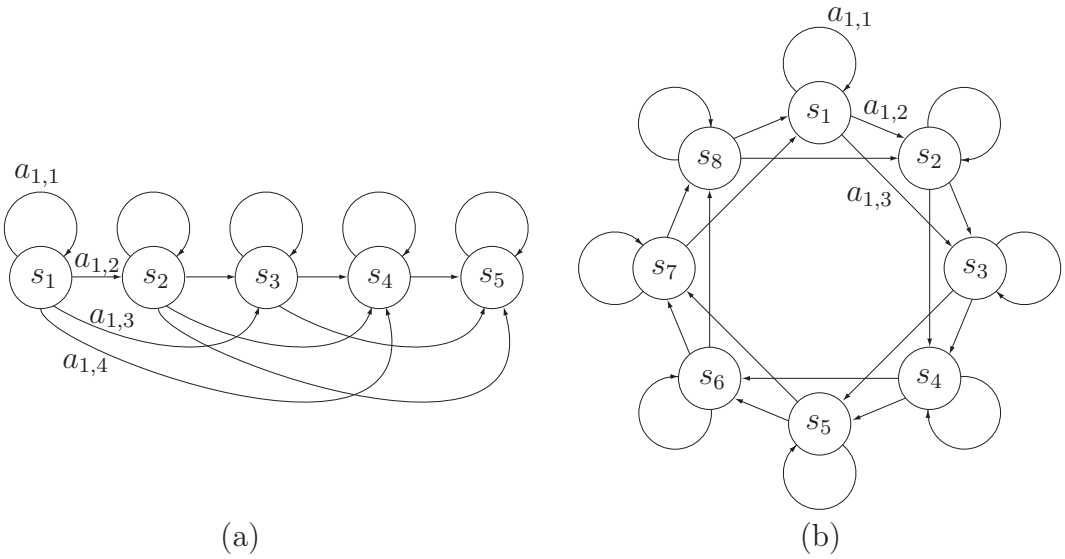


Figure 4.1: (a) A left-to-right HMM with five states. This HMM allows two state skips. (b) An eight state HMM with a ring topology. This HMM allows one state skip.

sets are initially estimated, and then optimised using an iterative re-estimation process, as discussed in the following sections.

4.5.1 Initial estimates

A *uniform* initial state distribution is used, that is, $\pi_i = 1/N$ for $i = 1, \ldots, N$. This implies that the HMM is equally likely to be entered at any given state, ensuring that the model is rotationally invariant through 360°.

The state transition probability distribution is initially specified as follows. The probability of staying in the same state, $a_{i,i}$ for $i = 1, \ldots, N$, is initially set to 0.8, while the probability of making a transition to the next state, $a_{i,i+1}$ for $i = 1, \ldots, N-1$ and $a_{N,1}$ is initially set to 0.2. This implies that a ring-structured HMM is used with no state skips.

An equal number of feature vectors (observations) are initially assigned to each state. Since high dimensional feature vectors are considered in this system, and signature training data is generally limited, the covariance matrix associated with each state's PDF cannot be reliably estimated[1]. We therefore estimate only the *mean* of the observations allocated to each state, while the covariance matrix is kept fixed. This implies that the dissimilarity value defined in Equation 4.3.7 is based on a Euclidean distance.

[1]This is often referred to as "the curse of dimensionality".

4.5.2 Parameter re-estimation

Each HMM is trained using the Viterbi re-estimation technique. The Viterbi re-estimation technique uses the Viterbi algorithm (Forney (1973)), which is a dynamic programming algorithm used to find the most likely sequence of hidden states that results in a specific observation sequence.

The Viterbi re-estimation technique, as applied to HMMs, is discussed in detail in Rabiner (1989) and Coetzer (2005).

4.6 Concluding remarks

In this chapter we explained how each writer's signature is modelled using N_r distinct, continuous observation ring-structured HMMs. Each of the N_r HMMs will constitute a base classifier. An optimal ensemble of base classifier, based on a specific operating criterion, is then selected, as described in Chapter 8.

In the next chapter we discuss how the system proposed in this dissertation achieves rotation, translation and scale invariance.

Chapter 5

Invariance and Normalisation Issues

5.1 Introduction

For a signature verification system to be used successfully in real world scenarios, it is essential that the system is able to tolerate a certain degree of variation in rotation, translation and scale of questioned signatures. There are several ways in which to make a signature verification system invariant to these three transformations. It is important to first make a distinction between what is meant by *invariance* and *normalisation*.

Figure 5.3 illustrates a basic system similar to the one developed in this dissertation. In Figure 5.3a we consider two identical signatures that differ only in their rotational orientation. In order for the signature verification system to be rotation invariant, $D(\mathbf{X}_1, \lambda)$ must be equal to $D(\mathbf{X}_2, \lambda)$. This implies that a system is considered to be rotation invariant if a questioned signature will always be classified in the same way, regardless of its rotational orientation.

The DRT-based observation sequences and HMM-based modelling techniques used in this dissertation constitute a rotation invariant system. It is important to note that, when considered separately, neither the observation sequence generation technique, nor the HMM-based signature modelling technique utilised in this dissertation is sufficient to ensure rotational invariance. It is the periodic nature of the observation sequences, *in conjunction* with the ring-structured HMMs, that guarantees a rotation invariant system. Note that in the system depicted in Figure 5.3a, no distinct step is necessary to correct the rotation of the signature; the system is intrinsically invariant with respect to changes in rotation.

In some cases, a system achieves rotation invariance through an explicit *normalisation* step. This normalisation step, which can apply to rotation, translation or scale, may occur at several different levels. For example, each signature image can be normalised with regard to its rotation before features

are extracted (see Figure 5.3b). Alternatively, a normalisation step may be applied directly to the features, so that $\mathbf{X}_1 = \mathbf{X}_2$, when the signatures from which the observation sequences are extracted differ only in translation, rotation or scale. If this is the case, the features themselves can be considered a rotation, translation or scale invariant representation of the signature.

In the following sections, we discuss how, and to what degree, the system developed in this dissertation achieves rotation, scale and translation invariance, either through the intrinsic design of the system, image processing techniques, or through specific normalisation steps. In Figures 5.1 and 5.2, the success of the scale, translation and rotation invariance of the proposed system is demonstrated.

5.2 Global features

Since the extraction of the global features is independent of the zoning process (see Section 3.3), it is worth considering the global observation sequence separately. Figure 5.4 illustrates the basic steps in obtaining a dissimilarity measure for the *global* observation sequence extracted from a questioned signature.

5.2.1 Rotation

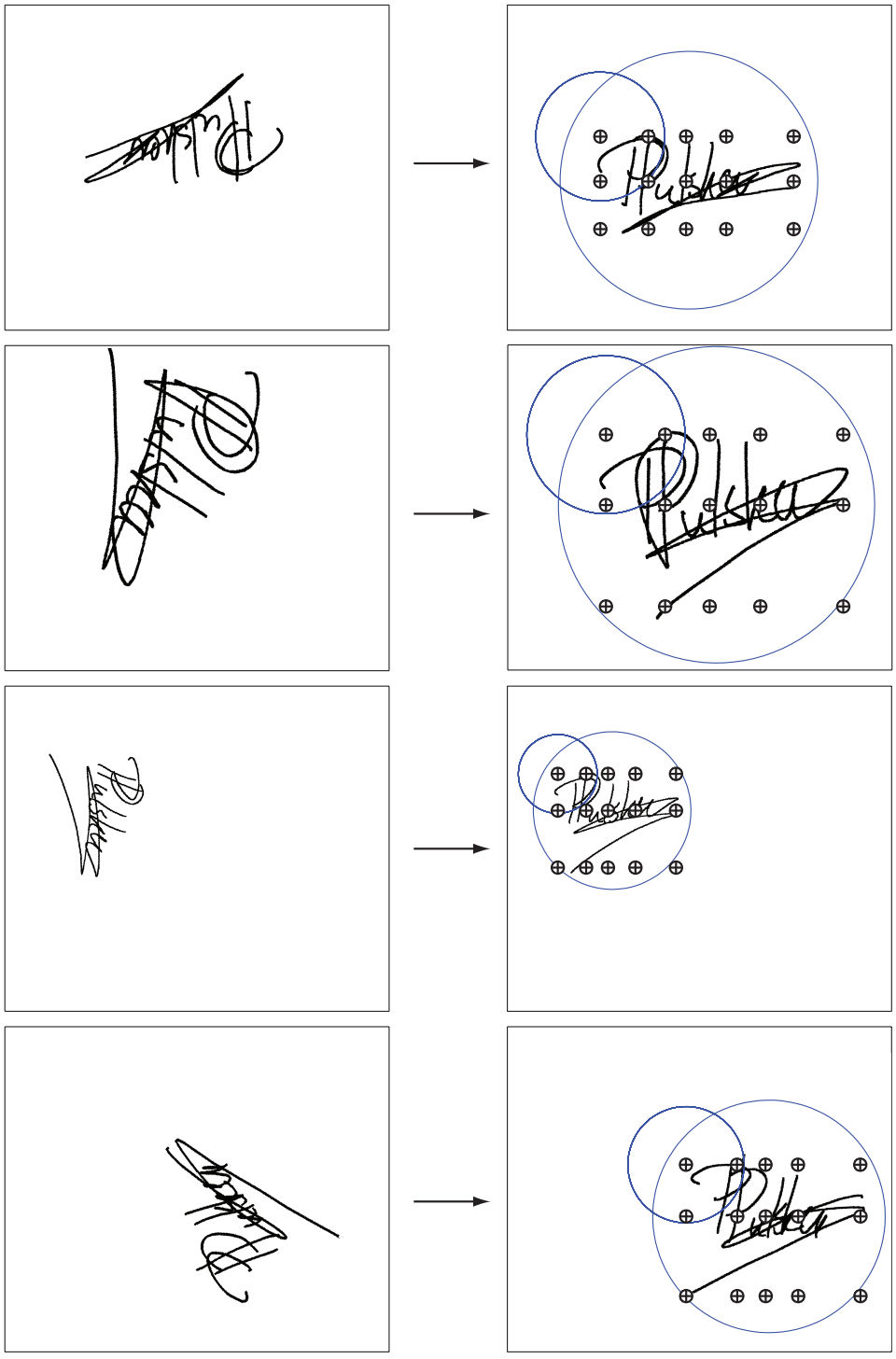
The inclusion of the projections at angles ranging from 180° to 360° (Figure 5.4c) make each observation sequence periodic, as explained in Chapter 3. The ring-structured, HMM-based modelling techniques (Figure 5.4d), as discussed in Chapter 4, in conjunction with the utilisation of periodic observation sequences, ensure that the system is rotation invariant with respect to the global features.

5.2.2 Translation

The DRT itself is not a shift invariant representation of an image, however the decimation of all zero-valued entries, and the subsequent rescaling of each column through linear interpolation (Figure 5.4c) ensures that the observation sequence extracted from the global retina is invariant with respect to translation.

5.2.3 Scale

The normalisation of the largest dimension of each signature (Figure 5.4a), together with the rescaling of each feature vector and the normalisation in variance of each observation sequence (Figure 5.4c), ensure that each global observation sequence is scale invariant.



(a) (b)

Figure 5.1: (a) Four different genuine samples belonging to the same writer. The signatures vary in rotation, translation and scale. (b) The signatures in (a) after rotation normalisation has been applied. Retina centroids are indicated with the $\oplus$ symbol. The first local retina, as well as the global retina are shown. Note that the retinas are scaled correctly.

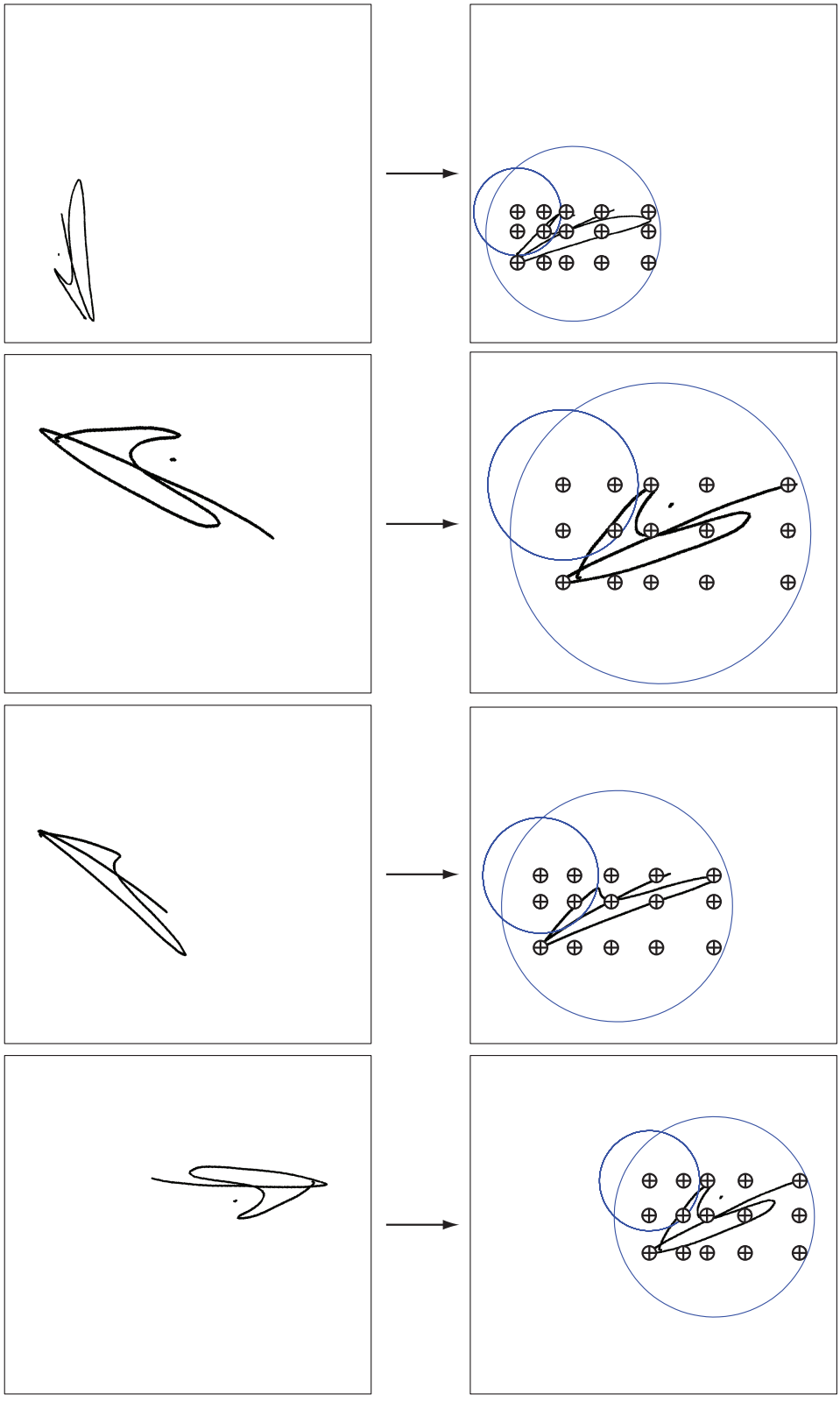


Figure 5.2: (a) Four different genuine samples belonging to the same writer. The signatures vary in rotation, translation and scale. (b) The signatures in (a) after rotation normalisation has been applied. Retina centroids are indicated with the $\oplus$ symbol. The first local retina, as well as the global retina are shown. Note that the retinas are scaled correctly.

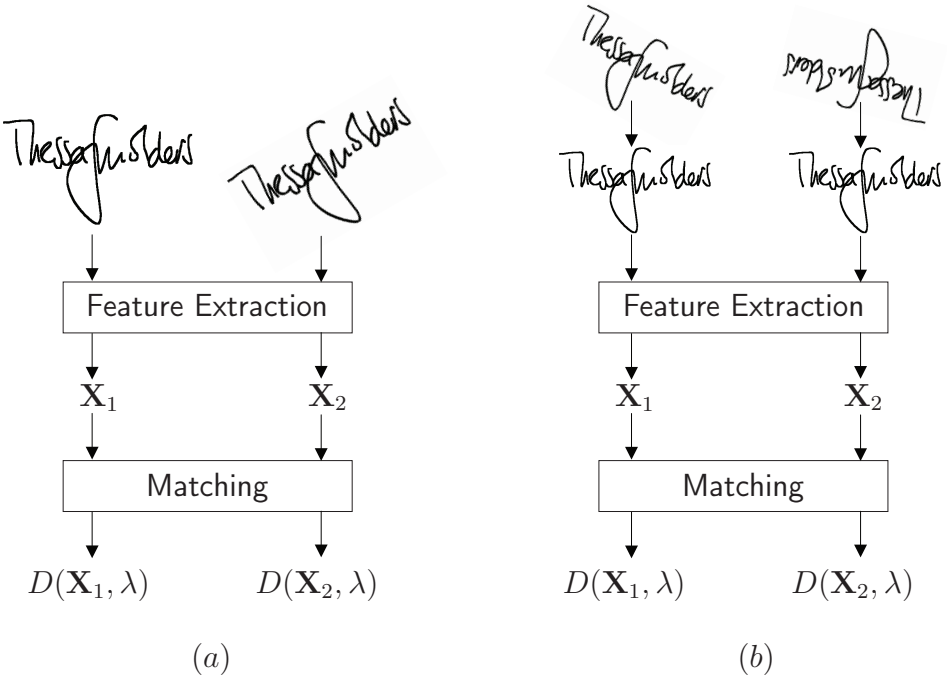


Figure 5.3: Examples of ways to achieve rotation invariance. In both (a) and (b) the system is only considered rotation invariant if $D(\mathbf{X}_1, \lambda) = D(\mathbf{X}_2, \lambda)$. If $\mathbf{X}_1 = \mathbf{X}_2$, the features can also be considered rotation invariant.

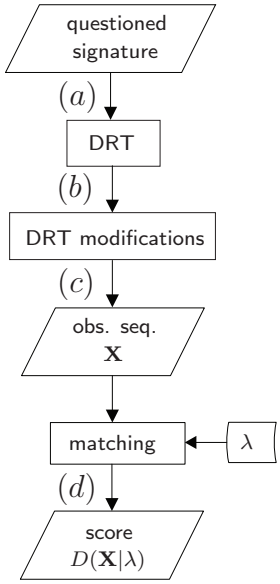


Figure 5.4: (a) A questioned signature. (b) DRT-based feature extraction, generating an initial observation sequence. (c) Modifications made to generate a final observation sequence $\mathbf{X}$. (d) Matching of the question signature, to produce a dissimilarity value $D(\mathbf{X}, \lambda)$.

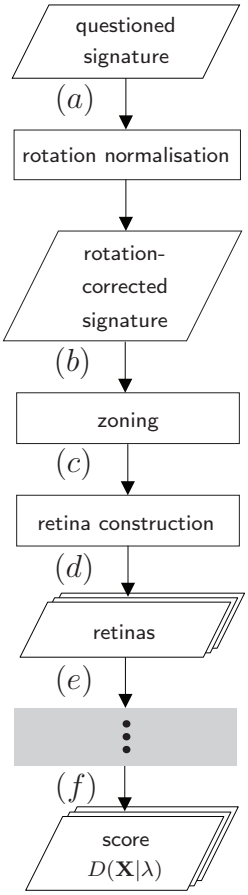


Figure 5.5: (a) A questioned signature. (b) Signature after rotation normalisation. (c) Zoning. (d) Retina construction. (e) Retinas. (f) The steps illustrated in Figure 5.4, which are applied to each of the N_r retinas to produce N_r dissimilarity values.

5.3 Local features

The local feature extraction and modelling techniques are identical to those used for the global features, and are therefore rotation, translation and scale invariant for the reasons set out in the previous section. The local features, however, also rely on the flexible grid-based zoning and retina construction techniques discussed in Section 3.3. It is therefore essential that the zoning and retina construction methods used are also rotation, translation and scale invariant in order for the system as a whole to possess these three properties.

Figure 5.5 illustrates the steps, in addition to those shown in Figure 5.4, that are necessary to obtain a dissimilarity value from a questioned signature.

5.3.1 Rotation

The zoning scheme (Figure 5.5c) is not rotation invariant, as it is implemented relative to a fixed axis. It is therefore necessary to explicitly normalise the

rotation of each signature before zoning (Figure 5.5b). The rotation of each questioned signature is normalised so that it has a similar rotational orientation to that of the enrolment signatures used for the claimed writer. This normalisation step is described in Section 5.4.2.

5.3.2 Translation

Since the zoning scheme (Figure 5.5c) utilises the gravity centre of the signature as a reference point (see Section 3.3.3), and all grid-lines are defined relative to this point using only information about the percentage of black pixels, the zoning scheme is inherently invariant to any translation of the signature.

5.3.3 Scale

The zoning scheme (Figure 5.5c) is scale invariant for the same reasons that it is translation invariant, however, it is also essential that the implemented retina construction method (Figure 5.5d) is scale invariant. By defining the radius of each retina as a function of the largest dimension of each signature, this criterion is met.

5.4 Rotation normalisation using HMMs

5.4.1 Background

It is assumed for now that all training signatures for a certain writer have consistent rotational orientation[1]. Normalizing the rotation of a questioned signature is therefore the process of rotating said signature to have a rotational orientation consistent with the training signatures of said writer.

While the periodic DRT-based observation sequence and the use of a ring-structured HMM to model each signature ensure that each global model is a rotation invariant representation of each signature, it is still necessary to normalise the rotation of each signature *before* zoning. This is done to ensure that corresponding retinas contain corresponding regions across all samples belonging to the same writer.

Fortunately, each trained HMM provides a convenient and robust way in which to normalise the rotation of a signature. Since each state in the trained HMM corresponds to observations obtained by calculating projections of the signature at a certain angle range, by determining the most likely state sequence, the most likely angle of rotation of a signature can be determined.

The algorithm for achieving this is outlined in Section 5.4.2 and clarified with a simple example in Section 5.4.3. Examples demonstrating the efficacy

[1]This assumption is not required, as illustrated in Section 5.4.5, but is made here for the sake of simplicity.

of the rotation normalisation algorithm are provided in Section 5.4.4, while the specific application of the algorithm to signatures in this dissertation are discussed in Section 5.4.5.

5.4.2 Algorithm

1. Define a reference state sequence $\mathbf{Q}^* = \{q_1^*, q_2^*, \ldots, q_T^*\}$, where

$$q_i^* = \left\lceil \frac{iN}{T} \right\rceil, i = 1, \ldots, T. \tag{5.4.1}$$

The reference state sequence $\mathbf{Q}^*$ depends only on T and N, where T is the length of the observation sequence (and state sequence), and N is the number of states in the trained HMM. Note that T and N are both system hyper-parameters; $\mathbf{Q}^*$ can therefore be considered a universal reference state sequence, as it is applicable to all writers.

2. Use Viterbi alignment to determine the most likely state sequence when matching the questioned signature's observation sequence with the relevant HMM,

$$\mathbf{Q} = \{q_1, q_2, \ldots, q_T\}.$$

3. The state sequence $\mathbf{Q}$ is then modified so that $q_i \geq q_{i-1}$ for $i = 2, \ldots, T$. This is achieved by substituting q_i with $q_i + N$ whenever this condition is not met. The modified state sequence is denoted by $\mathbf{Q}'$.

4. The mean difference between $\mathbf{Q}^*$ and $\mathbf{Q}'$, denoted by μ_Q, is then calculated as follows

$$\mu_Q = \frac{1}{T} \sum_{i=1}^{T} (q_i' - q_i^*). \tag{5.4.2}$$

5. The correction angle, denoted by Δ, is defined as

$$\Delta = \mu_Q \frac{360^\circ}{N}. \tag{5.4.3}$$

The correction angle Δ is the angle by which the questioned signature has to be rotated in order to align said signature with the reference orientation for the writer.

5.4.3 Example

The above algorithm is now demonstrated using a simple example. Figure 5.6a shows a typical training signature for a specific writer. The rotational orientation of this signature is also typical and can therefore be considered the reference orientation for this writer. Figure 5.6b shows a questioned (positive)

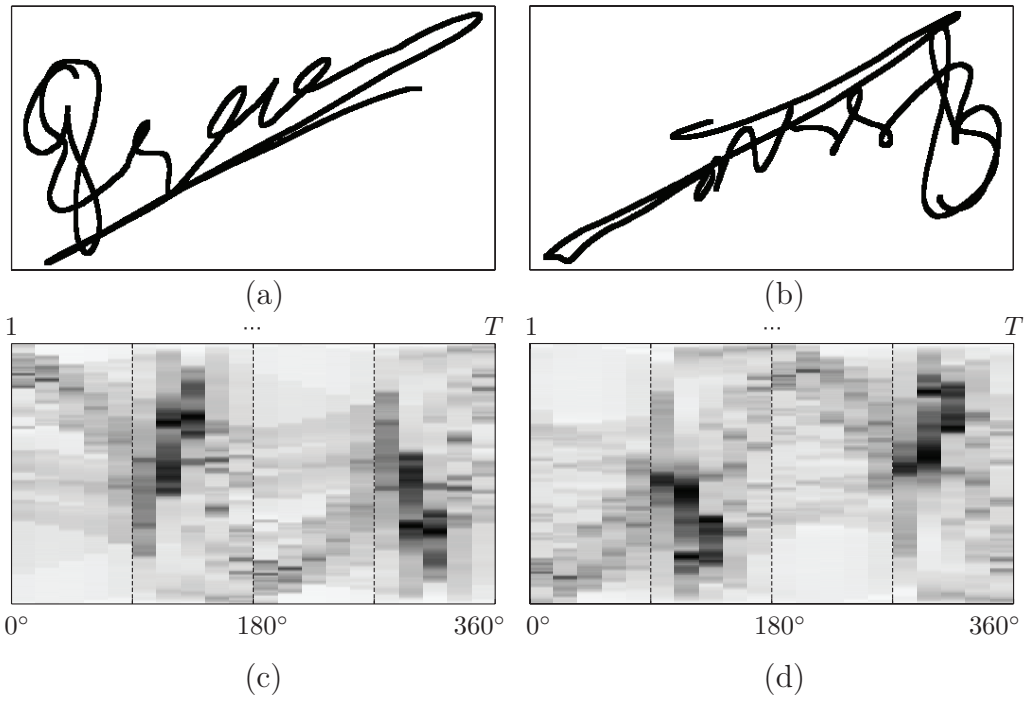


Figure 5.6: (a) A training signature for a specific writer. The rotational orientation of this signature is typical and can be considered as the reference orientation for this writer. (b) A questioned signature that has been rotated by approximately 180° relative to the reference orientation for this writer. (c) The observation sequence extracted from (a) with $T = 20$. (d) The observation sequence extracted from (b) with $T = 20$.

signature for this writer that has been rotated by approximately 180° relative to the reference orientation. Figures 5.6c and d show the observation sequences for the respective signatures. For this example, $N = 10$ and $T = 20$, indicating that the signature is modelled using an HMM with ten states, and an observation sequence of length twenty. These values are typically too small to be appropriate for real-world scenarios, but are chosen here for ease of illustration. The five steps of the algorithm are shown below.

1. For the case when $N = 10$ and $T = 20$, the reference state sequence is

$$\mathbf{Q}^* = \{1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10\}.$$

2. The most probable state sequence for the observation sequence of Figure 5.6d, when matched with the trained HMM for this writer, is

$$\mathbf{Q} = \{6, 6, 6, 7, 7, 8, 9, 9, 10, 1, 1, 1, 1, 2, 2, 3, 4, 4, 5, 6\}.$$

3. The state sequence is then modified to become

$$\mathbf{Q}' = \{6, 6, 6, 7, 7, 8, 9, 9, 10, 11, 11, 11, 11, 12, 12, 13, 14, 14, 15, 16\}.$$

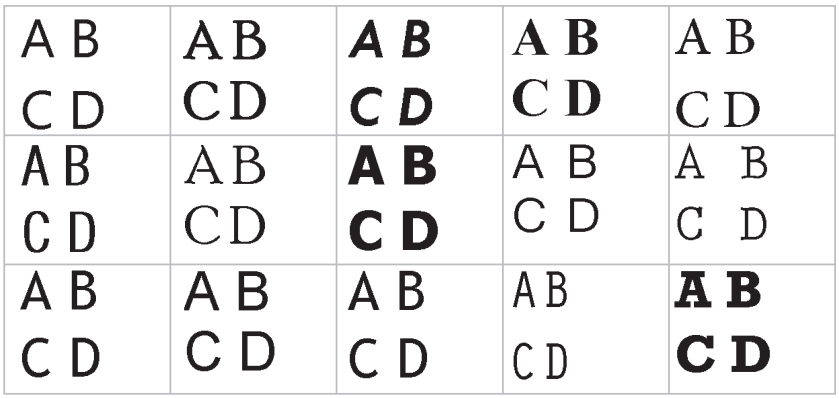


Figure 5.7: 15 images used to train an HMM (using DRT-based features). Each image contains the letters "A B C D" in a different font.

4. The difference between $\mathbf{Q}'$ and $\mathbf{Q}^*$ is then calculated:

$$\mathbf{Q}' - \mathbf{Q}^* = \{5, 5, 4, 5, 4, 5, 5, 5, 5, 6, 5, 5, 4, 5, 4, 5, 5, 5, 5, 6\}.$$

The mean difference is therefore

$$\mu_Q = 4.9.$$

5. Finally, the correction angle is calculated,

$$\Delta = (4.9)\frac{360^\circ}{10} = 176.4^\circ.$$

This correction angle of $\Delta = 176.4^\circ$ is consistent with the initial assumption that the signature had been rotated by approximately 180°. Using small values for T and N will not typically allow very precise rotation corrections, but higher values of T and N, which are used in practice, allow accurate correction angles to be determined.

5.4.4 Examples of efficacy

The efficacy of the rotation normalisation method is now demonstrated. Figure 5.7 shows fifteen samples which are used to train an HMM (features are extracted using the DRT-based feature extraction technique used throughout this dissertation). Each training image contains the letters "A B" written above the letters "C D". This configuration is chosen because it has an obvious correct rotational orientation, and because it contains no obvious principal component (direction of maximum variation).

Since the rotation normalisation algorithm is very robust, testing images were created by altering the training images significantly. Figure 5.8a shows

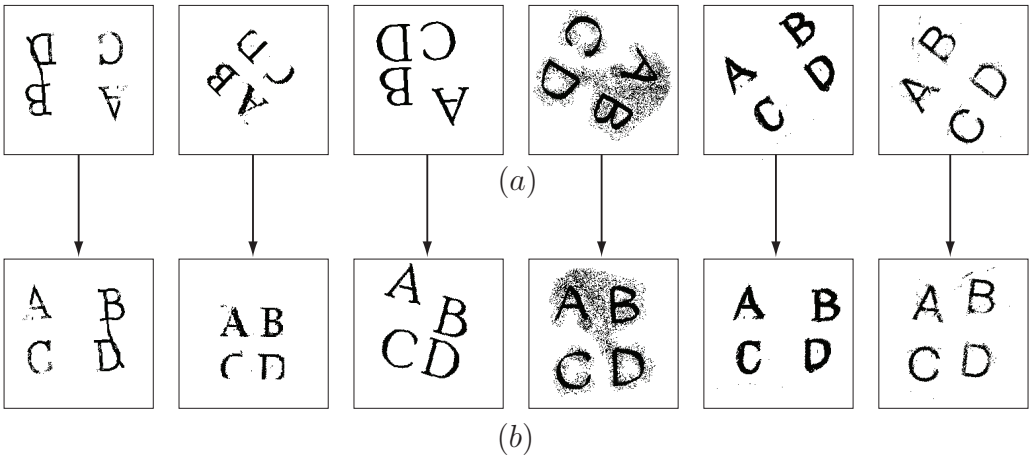


Figure 5.8: (a) Sample images used to test the rotation normalisation scheme. When compared to the training signatures of Figure 5.7, these images contain significant noise. (b) Sample images after rotation normalisation. Each image has been rotated correctly.

several examples of these testing images with various rotational orientations. Figure 5.8b shows the images in Figure 5.8a, after rotation normalisation has been applied. Note that each image has a rotational orientation consistent with the images used to train the HMM.

5.4.5 Rotation normalisation applied to signatures

The assumption was initially made that all training signatures belonging to a certain writer have a consistent rotational orientation. The reference rotation for a certain writer can therefore be determined by examining any one of said writer's training signatures. This assumption is in fact not necessary. When training an HMM using signatures that differ in rotational orientation, the Viterbi re-estimation algorithm will take longer to converge, but will typically model the writer's signature with a rotational orientation corresponding to the *average* rotational orientation of the training samples.

When a new writer is enrolled into the system, a global model is trained, that is then used to normalise the rotational orientation of said training signatures. It is therefore necessary to train the global HMM for each writer before each signature can be zoned and HMMs for the local retinas can be trained.

Figure 5.9 illustrates this process. Although fifteen signatures were used to train this model, only seven are shown. The rotational orientation of the signatures is normalised so that each signature now has an orientation consistent with the *average* orientation of the training signatures.

When a questioned signature is presented, the system attempts to align the rotational orientation of said signature with that of the claimed writer's reference rotational orientation. In the case of positive questioned signatures, this normalisation scheme works as expected, correcting the rotational orienta-

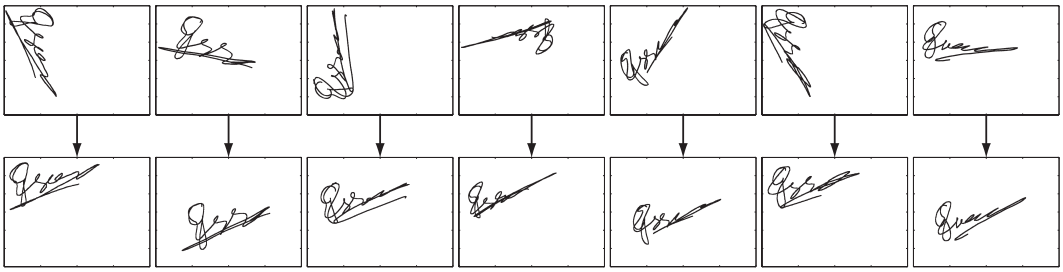


Figure 5.9: Rotation normalisation applied to training signatures. Seven typical training signatures for this writer are shown. The HMM trained using these signatures is used to normalise the rotation of each training signature.

tion of the questioned signature. If the questioned signature is a poor forgery, attempting to normalise said signature's rotation with respect to what is essentially a different signature is nonsensical. The correction angle in this case may be considered arbitrary, and irrelevant.

A significant correction angle for a questioned signature (greater than $30°$ or less than $-30°$), implies that the signature is either genuine (but written with a significantly different rotational orientation to that of the training signatures) or that the signature is a forgery. Since the system developed in this dissertation is designed to be completely rotation invariant, the correction angle does not directly influence the classification decision. However, if the rotational orientation of each questioned signature is considered to be a vital discriminative writer-specific attribute, the correction angle can be used to influence the decision of the classifier.

5.5 Conclusion

In this chapter we discussed how the proposed system is invariant with respect to the scale, translation and rotation of a signature. We introduced an algorithm that utilises a trained HMM to correct the rotational orientation of each questioned signature. The efficacy and robustness of the algorithm is clearly demonstrated. In the next chapter, we discuss verification and performance evaluation measures.

Chapter 6

Verification and Performance Evaluation Measures

6.1 Introduction

In this chapter, we discuss how a decision is made, as whether to classify a test pattern as positive or negative. Although we consider multiple classifiers for each writer (that is a classifier associated with each of the N_r retinas defined in Chapter 3), in this chapter, we only illustrate how a decision is made at the level of an individual classifier. A discussion on how a final decision is made, using classifier combination, is postponed to Chapter 8.

In order to illustrate the principles involved in classifying a single retina as fraudulent or genuine, in the remainder of this chapter, we consider only the global retina defined in Chapter 3. For this discussion only, we further assume that a signature is classified solely based on said global retina. Throughout the remainder of this chapter (as well as in Chapter 7) the term "classification of a signature" therefore implies classification based on the global retina only. The reader is reminded though, that the principles discussed here apply to each of the N_r classifiers defined in Section 3.3.4 (each one associated with a specific retina) and that the final decision is made by combining the decisions of a subset of these classifiers (as discussed in Chapter 8).

6.2 Thresholding

The decision as whether to classify a questioned signature as positive or negative is based on the dissimilarity value $D(\mathbf{X}|\lambda)$ defined in Chapter 4. A lower value for $D(\mathbf{X}|\lambda)$ corresponds to a higher likelihood that the questioned signature is positive. Figure 6.1a shows the dissimilarity value distribution (histogram) of positive and negative signatures for a *specific* writer w. This figure was generated by training an HMM for writer w (that is λ_w) using fifteen positive training signatures, and then matching positive evaluation signatures and

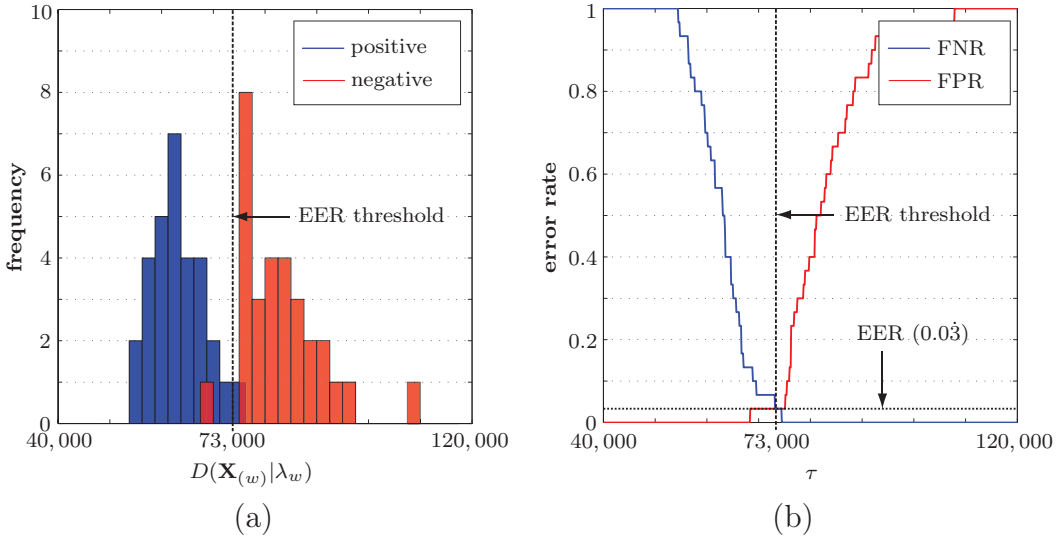


Figure 6.1: (a) Histogram of dissimilarity values for thirty positive (blue) and thirty negative (red) evaluation signatures belonging to the *same* writer. (b) FPR and FNR versus τ for the same signatures considered in (a). The EER occurs where $\tau \approx 73,000$.

thirty negative evaluation signatures to λ_w through Viterbi alignment in order to obtain sixty dissimilarity values, $D(\mathbf{X}_{(w)}|\lambda_w)$ (negative log-likelihoods). As expected, the positive evaluation signatures (shown in blue) are generally associated with lower dissimilarity values than negative evaluation signatures (shown in red). Classifying a questioned signature therefore involves stipulating a decision boundary, or threshold, denoted by τ, so that a questioned signature $\mathbf{X}_{(w)}$ is *accepted* (classified as positive) if

$$D(\mathbf{X}_{(w)}|\lambda_w) < \tau \tag{6.2.1}$$

and is otherwise rejected (classified as negative).

6.3 Performance evaluation measures

It is clear from Figure 6.1a that (for the specific writer considered here) no threshold exists that creates a perfect separation between the positive and negative signatures. It is therefore impossible to classify each of these questioned signatures correctly. In practice, a compromise therefore has to be made between incurring false positives (FPs) and false negatives (FNs). A FP is the classification of a negative (fraudulent) signature as positive (genuine), while a FN is the classification of a positive signature as negative.[1] A true positive (TP) and a true negative (TN) refer to the correct classification of positive and negative signatures, respectively.

[1]In statistics, FPs and FNs are known as Type 1 and Type 2 errors, respectively.

The *false negative rate* (FNR), defined as,

$$\text{FNR} = \frac{\text{FN}}{\text{FN} + \text{TP}} \tag{6.3.1}$$

is the ratio of the number of rejected positive signatures to the total number of positive signatures considered. The *false positive rate* (FPR), defined as,

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{6.3.2}$$

is the ratio of the number of accepted negative signatures to the total number of negative signatures considered.

The FNR and FPR form the basis of all the performance evaluation measures used in this dissertation. The *true positive rate* (TPR) and *true negative rate* (TNR) are defined as $1-\text{FNR}$ and $1-\text{FPR}$, respectively. Another important performance evaluation measure, the *equal error rate* (EER), is defined as the error rate at which FNR = FPR (or TPR = TNR). Figure 6.1b shows the FPR and FNR versus τ for a *specific* writer.

The threshold value ($\tau \approx 73,000$) that results in the ERR is indicated in both Figures 6.1a and b. It is clear from Figure 6.1a that there are two misclassifications at the EER; one positive signature is classified as negative, and one negative signature is classified as positive. Since thirty positive and thirty negative signatures were used to evaluate this classifier, one misclassification per class implies that the EER is equal to $3.\dot{3}\%$ (see Figure 6.1b).

A further performance evaluation measure used throughout this dissertation is the so-called *area under the curve* (AUC), which refers to the area under a classifier's ROC curve. An AUC of 1 indicates perfect performance, while an AUC of 0.5 indicates that the classifiers performance is equivalent to random chance. ROC curves are discussed in the next section.

6.4 ROC curves

Receiver operating characteristic (ROC) space is a two-dimensional space with the FPR and TPR on the horizontal and vertical axes, respectively. Figure 6.2 illustrates several points of interest in ROC space. The points $A(0,0)$ and $B(1,1)$ indicate the performance of classifiers which reject and accept *all* signatures, respectively. The point $C(0.7,0.7)$ depicts the performance of a classifier that accepts 70% of all positive questioned signatures, and 70% of all negative questioned signatures. The classifiers of which the performance is represented by the points A,B and C, and all classifers of which the performance is represented by points which lie on the diagonal, FPR = TPR, can therefore be considered trivial. The point $D(0,1)$ represents the performance of a classifier which classifies every signature correctly. Generally, a classifier's performance increases as it approaches the operating point $D(0,1)$.

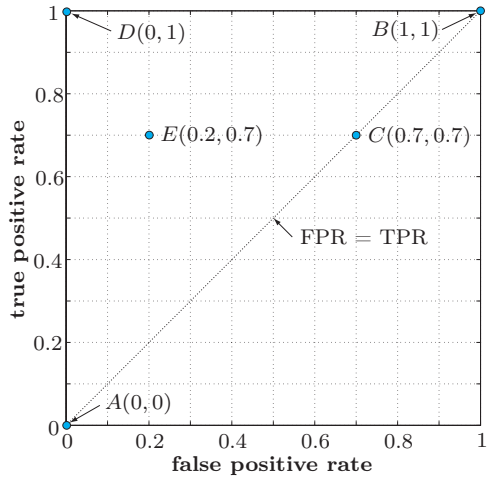


Figure 6.2: ROC space. The points A,B and C all lie on the diagonal FPR = TPR, and are therefore considered trivial. The point D depicts the performance of a classifier which makes perfect decisions. A classifier with an FPR of 0.2 and a TPR of 0.7 is depicted by the point E.

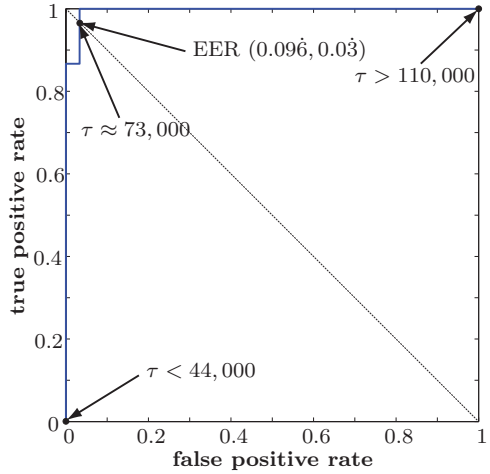


Figure 6.3: ROC curve of the classifier considered in Figure 6.1. The arrow indicates the EER of $3.\dot{3}\%$, which occurs at $\tau \approx 73,000$.

The performance of a *continuous* classifier, like one of the HMMs used in this dissertation, can therefore be depicted by a continuous parametric curve in ROC space, where the parameter is the decision threshold τ. This curve is referred to as an ROC curve. A *discrete* classifier (of which the performance is depicted by a single point in ROC space) is therefore associated with each selected threshold. In the context of continuous classifiers, a point in ROC space is therefore also referred to as an operation point. The ROC curve for the classifier considered in Figure 6.1b is shown in Figure 6.3.

6.5 Conclusion

In this chapter, we showed how a threshold is used to classify a signature as either positive or negative. We also introduced several performance evaluation measures, and showed that different thresholds are associated with different operating points in ROC space. In the next chapter, we discuss how the performance of a system, that is the combined performance for multiple writers, is evaluated in ROC space.

Chapter 7

Score Normalisation

7.1 Introduction

Throughout this chapter we use the word *score* as a synonym for *dissimilarity value* (see Equation 6.2.1). Although this is technically incorrect, since a smaller dissimilarity value is typically associated with a higher score, the relevant literature refers to the normalisation of the numerical output (matching scores or dissimilarity values) of a classifier as "score normalisation" (see Jain *et al.* (2005)). We continue with this trend for the sake of consistency and clarity.

In Section 7.2, we define and justify the need for score normalisation. We present a brief overview of current score normalisation strategies, using generic classifiers, and introduce several novel strategies in Section 7.3. Practical considerations, which pertain to the limitations on the available data, are also discussed in Section 7.3. Section 7.4 deals with the important operational considerations of implementing the score normalisation strategies introduced in Section 7.3. In Section 7.5 we discuss the specific application of score normalisation in this dissertation. Finally, in Section 7.6.3 we propose some further transformations of the normalised decision threshold, in order to enable an operator (for example, a bank manager) to impose more intuitively understandable thresholds, as well as to simplify the ensemble selection and fusion strategies discussed in Chapter 8.

7.2 Background

The ROC curve shown in Figure 6.3 (see previous chapter) depicts the performance of a single continuous classifier trained for a specific writer. In practice, however, it is essential to consider the global performance of a system, that is for *all* the enrolled writers.

Since the threshold τ is the only independent variable, and therefore the only variable under the direct control of the operator (a bank manager, for

example), it is only feasible to generate a combined ROC curve by summing the FPs and TPs that correspond to the same threshold across all writers. If the same number of signatures are used to evaluate each classifier's error rates, we can consider the combined ROC curve as an *average* ROC curve, where each new point on the averaged curve is calculated by averaging points on the individual ROC curves that correspond to the same threshold τ[1]. This is known as *threshold averaging* (see Fawcett (2006)).

Figure 7.1a shows the error rates plotted against τ for four different classifiers corresponding to four different writers. It is clear that the numerical ranges of the scores for these different writers are incommensurable. For example, a decision threshold of $\tau \approx 5,800$ (indicated by the vertical dotted line) results in the fourth classifier (that is, the 4th writer, $j = 4$) operating with an EER, yet the same threshold will result in *all* signatures being rejected for $j = 1$. Similarly, the same threshold applied to $j = 2$ and $j = 3$ results in all positive signatures and the majority of the negative signatures being accepted.

The operating points corresponding to $\tau \approx 5,800$ (now indicated with x's), as well as the combined ROC curve, are shown in Figure 7.1b. The error rates plotted against τ for the system are also shown in Figure 7.1a and labelled as "ave". The x's on each ROC curve in Figure 7.1b indicate the operating points (or discrete classifiers) which were averaged to generate the new point, indicated by $\circ$ on the averaged ROC curve. The poor performance of the combined ROC curve (relative to the performance of the individual ROC curves), as well as the diverse range of operating points (amongst individual classifiers/writers) associated with the same threshold, make it necessary to *normalise* each writer's scores before generating an average ROC curve.

7.3 Normalisation strategies

In practice, the scores of each writer must be normalised (transformed to a common domain), so that threshold averaging can be used to generate a ROC curve depicting the performance of the system across all writers. The symbol τ is used to denote a decision threshold in the domain of un-normalised scores (see Equation 6.2.1). We now define a new decision threshold ϕ, which is used in the domain of normalised, or transformed, scores. We use $D_*(\mathbf{X}_{(w)}|\lambda_w)$ to denote a transformed score (or dissimilarity value), so that if

$$D_*(\mathbf{X}_{(w)}|\lambda_w) < \phi \tag{7.3.1}$$

the questioned signature $X_{(w)}$ is classified as a positive signature belonging to writer w, where $*$ denotes the normalisation strategy used (see Section 7.3.2 - 7.3.4).

[1]If each classifier is evaluated using a different number of signatures, the combined ROC curve can be generated by considering *weighted*-averaging, where the weights are proportional to the number of signatures used to evaluate each individual writer's performance.

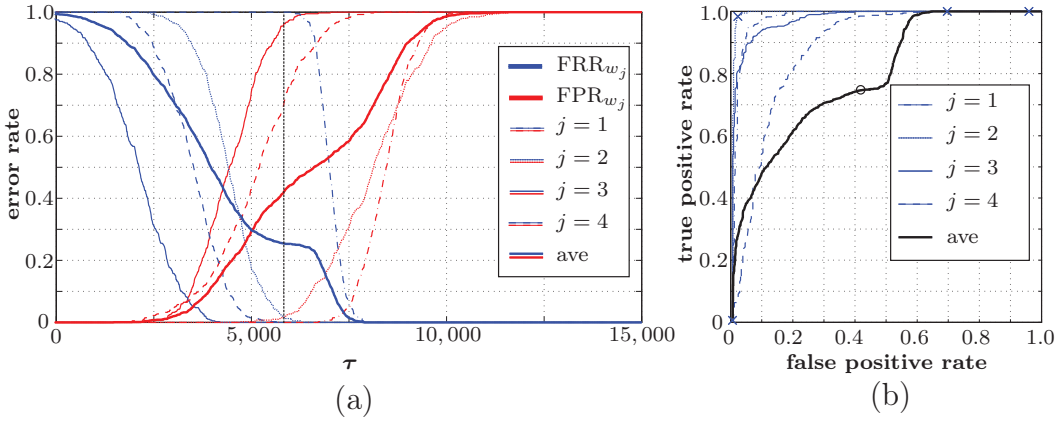


Figure 7.1: (a) Error rates shown for four different writers, without score normalisation. (b) ROC curves for the four writers considered in (a). The average ROC curve is also shown. The x's indicate the points corresponding to a threshold of $\tau \approx 5,800$ for each writer. The $\circ$ indicates the point on the average ROC curve which was calculated by averaging the x's.

In order to illustrate several different approaches to score normalisation, we now consider four *generic* classifiers, which each have predefined *Gaussian*[2] positive and negative score distributions, that is $N^w_{\oplus}(\mu^w_{\oplus}, \sigma^w_{\oplus})$ and $N^w_{\ominus}(\mu^w_{\ominus}, \sigma^w_{\ominus})$, respectively, where $N^w_{\ominus}(\mu^w_{\ominus}, \sigma^w_{\ominus})$ denotes the Gaussian distribution with a mean $\mu^w_{\ominus}$ and standard deviation $\sigma^w_{\ominus}$ associated with negative samples of user w. Similarly, $N^w_{\oplus}(\mu^w_{\oplus}, \sigma^w_{\oplus})$ denotes the Gaussian distribution associated with the positive samples of user w. These predefined distributions are shown in Figure 7.2. Each classifier is associated with a different *user*[3]. Note that these are the same "classifiers" that were considered in Figure 7.1, where no score normalisation was used. We use $E = 400$ positive and $E = 400$ negative samples per user to evaluate each score normalisation scheme. The scores are randomly sampled from the respective distributions, that is $D(\mathbf{X}_{(w)}|\lambda_w) \sim N^w_{\oplus}(\mu^w_{\oplus}, \sigma^w_{\oplus})$ for positive samples, and $D(\mathbf{X}_{(w)}|\lambda_w) \sim N^w_{\ominus}(\mu^w_{\ominus}, \sigma^w_{\ominus})$ for negative samples. In the following section, we discuss the practical constraints on the available data, before we consider score normalisation strategies based on the estimation of *only* positive, *only* negative and *both* positive and negative score distributions. Note that this is not intended to be a comprehensive overview of all possible score normalisation strategies, but only those which are relevant to this discussion. A discussion on several other score normalisation methods, such as the so-called Tanh, Sigmoid, Min-Max and Median methods, can be found in Jain *et al.* (2005), where the emphasis is on score normalisation in multi-modal biometric systems.

[2]The limitations of assuming Gaussian score distributions are elaborated on in the following sections.

[3]Since we are first considering the general application of score normalisation, we use the term "user" as a generic term for "writer", and "sample" as a generic term for "signature".

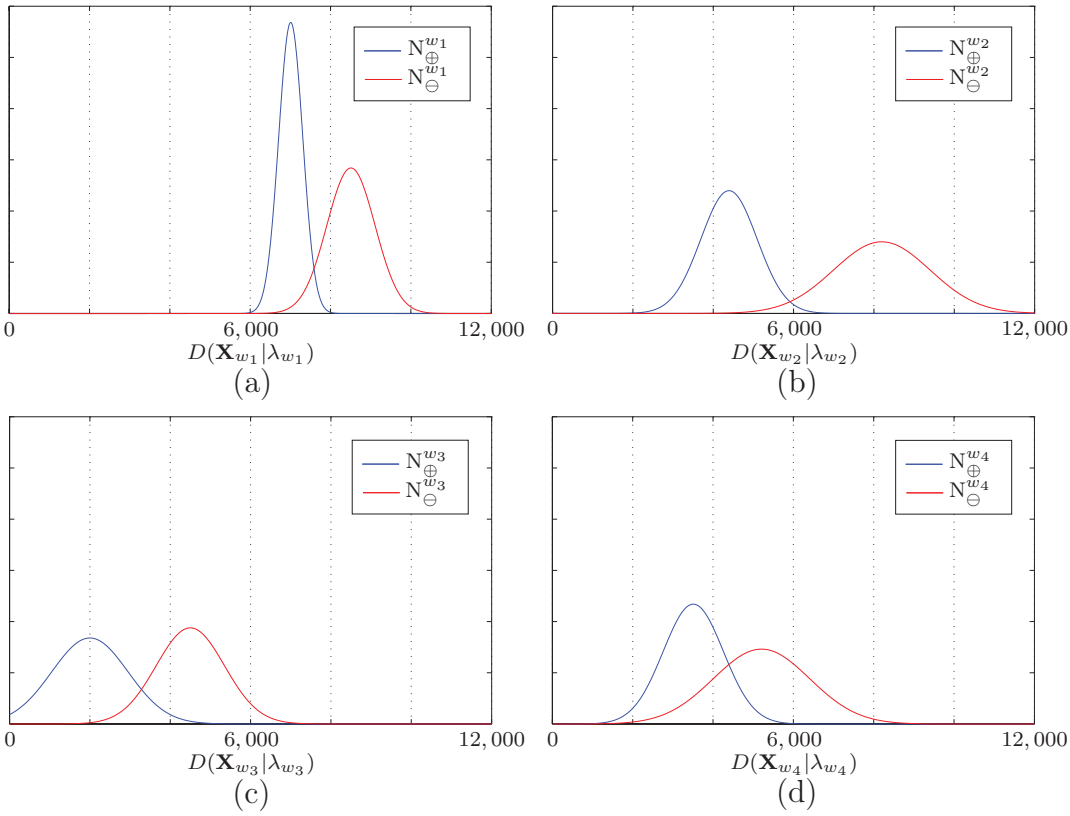


Figure 7.2: (a)-(d) The Gaussian score distributions associated with the four generic classifiers used in this section.

7.3.1 Practical considerations

Since all normalisation schemes essentially entail the estimation of score distributions for each writer using the *available* data, it is important to consider exactly what data is available at the time of deployment. We focus specifically on score normalisation as it applies to biometric verification systems.

Since positive samples are necessary for the initial training of a generative classifier (like an HMM), it is always assumed that a number of positive samples are available for each user. These available positive samples can, in addition to training the classifier (model), also be used to estimate the distribution of positive scores for the user[4], by, for example, matching the same samples to the trained model. In most biometric systems (including signature verification systems aimed at detecting only random forgeries) it is possible to also accurately estimate the score distribution of negative samples for each user, as negative samples are not user-specific in these scenarios. For example,

[4]Ideally, a separate set of positive samples should be used to estimate the positive score distribution, as using the same samples that were used to train the classifier may result in an optimistically biased positive score distribution. A limited number of available positive samples often makes this strategy unfeasible in practice.

in a fingerprint verification system, any fingerprint may be considered a negative example of another user's fingerprint, and may be used to estimate the negative score distribution for this user.

In behavioural biometrics, of which signature verification aimed at detecting skilled forgeries is an example, negative samples are user-specific. This is the case since a negative sample is produced when a person makes a concious effort to imitate another user's signature. It is therefore not reasonable to expect user-specific negative examples to be available at the time of enrolment (that is, before system deployment), in these scenarios.

In the following sections, we use V to denote the number of samples, per user, which are available to estimate either the positive or negative (or both) score distributions. A score associated with each of these samples is obtained by randomly sampling from the relevant distribution. We also consider the *ideal* case, where both the size of the evaluation set (E) and the size of the set of available samples (V) approach infinity. We approximate the ideal case by using $V = 10,000$ and $E = 10,000$. The ideal case can therefore be considered the upper limit of each score normalisation scheme, in that it illustrates what can be achieved if each writer's score distribution(s) are estimated with negligible error.

7.3.2 Positive samples

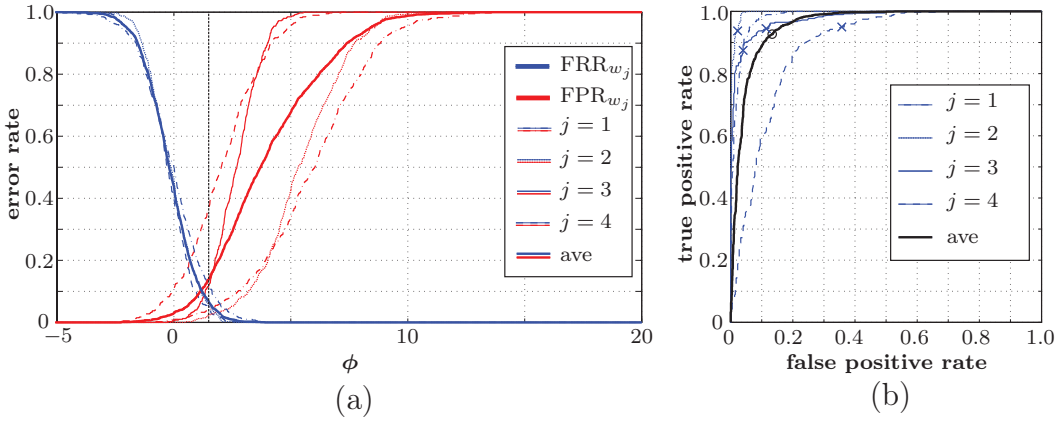
We consider two different approaches to estimating the distribution of scores associated with positive samples.

Z_P-score normalisation. The first approach, referred to as Z_P-score normalisation, assumes a Gaussian[5] distribution of scores associated with the positive samples belonging to each user. The V *available* positive samples, which are generated randomly based on each user's defined distributions, are used to estimate the mean μ_w and standard deviation σ_w for *each* classifier/user w. The transformed score, denoted by $D_{Z_P}(\mathbf{X}_{(w)}|\lambda_w)$, is defined as (Jain *et al.* (2005))

$$D_{Z_P}(\mathbf{X}_{(w)}|\lambda_w) = \frac{D(\mathbf{X}_{(w)}|\lambda_w) - \mu_w}{\sigma_w}. \tag{7.3.2}$$

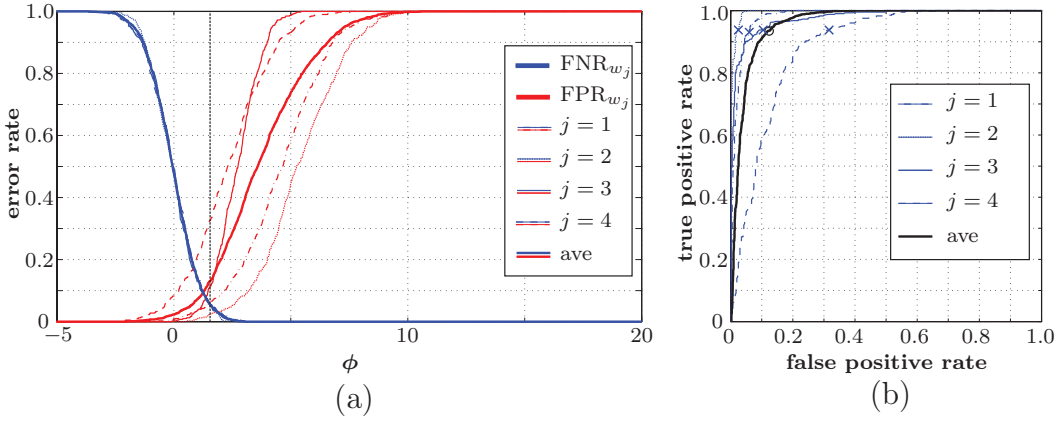
If the assumption that each distribution is Gaussian is valid, and each mean and standard deviation is estimated accurately, then the distribution of normalised scores for positive samples should be Gaussian with a mean of 0 and a standard deviation of 1. Figures 7.3 and 7.4 show the normalised error rates (plotted against ϕ) and ROC curves for $V = 10$ and $V = 100$, respectively. Note that each FNR can be considered an approximation of a Gaussian cumulative distribution function. Figure 7.5 depicts the "ideal" case ($V, E = 10,000$), where the score distribution of positive samples for each

[5]Although not typically found in practice, Z_P-score normalisation will work equally well when the positive scores have a uniform distribution.



(a) (b)

Figure 7.3: Z_P-score normalisation with V = 10. (a) Error rates, plotted against ϕ, shown for four different users after normalisation. (b) ROC curves for the four writers considered in (a). The combined ROC curve (ave) is also shown. The ○ indicates the point on the average ROC curve which was calculated by averaging the x's on each individual ROC curve. The vertical line in (a) indicates the threshold that corresponds to said x's.

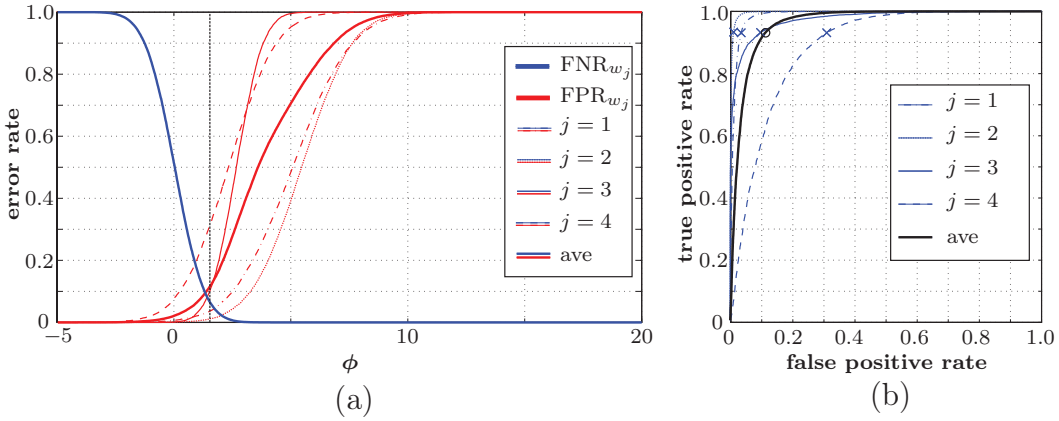


(a) (b)

Figure 7.4: Z_P-score normalisation with V = 100. (a) Error rates, plotted against ϕ, shown for four different users after normalisation. (b) ROC curves for the four writers considered in (a). The combined ROC curve (ave) is also shown. The ○ indicates the point on the average ROC curve which was calculated by averaging the x's on each individual ROC curve. The vertical line in (a) indicates the threshold that corresponds to said x's.

user is estimated with negligible error. Note that, as the estimations improve, Z_P-score normalisation approaches horizontal averaging in ROC space. That is, each point on the combined ROC curve is calculated by averaging points on the individual ROC curves which have the same TPR (or lie on the same horizontal line).

TPR-score normalisation. A second approach to score normalisation based on estimating the distribution of positive samples, is so-called TPR-



(a) (b)

Figure 7.5: Z_P-score normalisation in the "ideal" case. (a) Error rates, plotted against ϕ, shown for four different users after normalisation. (b) ROC curves for the four writers considered in (a). The combined ROC curve (ave) is also shown. The ∘ indicates the point on the average ROC curve which was calculated by averaging the x's on each individual ROC curve. The vertical line in (a) indicates the threshold that corresponds to said x's.

score normalisation. TPR-score normalisation (also referred to as FNR-score normalisation) has been proposed in Wang *et al.* (2008), although no experiments were conducted in this paper. In TPR-score normalisation, a discrete distribution for each user's positive scores is estimated. The scores are then normalised so that for a specific ϕ, each writer operates with the same TPR. An algorithm for calculating the discrete normalisation function can be found in Wang *et al.* (2008). Note that this discrete transformation function is non-linear. It is convenient in this case to choose ϕ to correspond to the FNR, so that, for example, when $\phi = 0.1$, each classifier operates with a FNR of 0.1. Figures 7.6 and 7.7 depict TPR-score normalisation for $V = 10$ and $V = 100$, respectively. The "ideal" case ($V, E = 10,000$) is shown in Figure 7.8.

It is clear from these artificial examples, that Z_P-score normalisation and TPR-score normalisation are similar, in that they both approach horizontal averaging when $V, E \to \infty$. In the ideal case, where the distributions are Gaussian (as they were defined in the above experiments) and estimated accurately, Z_P-score normalisation and TPR-score normalisation are equivalent. Figures 7.5b and 7.8b verify this.

TPR-score normalisation is preferable when the score distributions of each user are not known, or cannot be estimated using a continuous distribution. Estimating a discrete distribution accurately, however, requires a significant number of samples, and is therefore not always possible in practice. If the number of samples is limited, and the distributions of scores for positive samples are known to be approximately Gaussian, then superior results should be possible when Z_P-score normalisation is used.

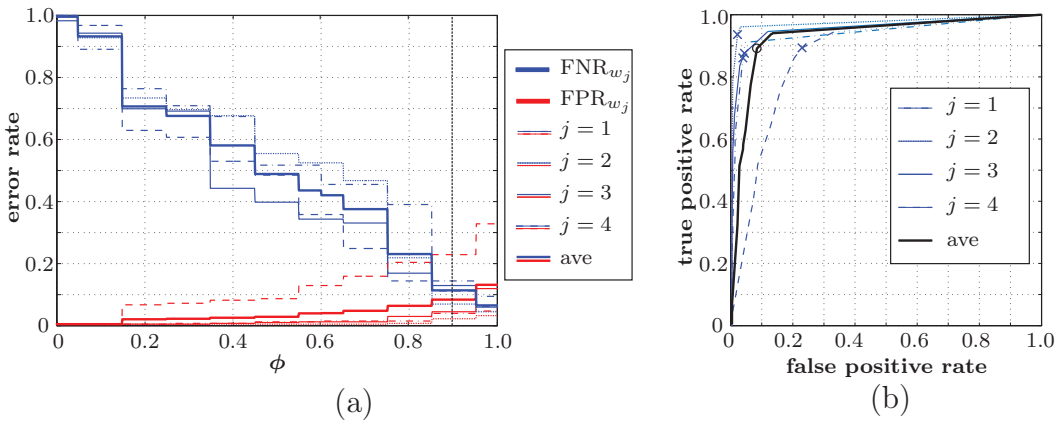


Figure 7.6: TPR-score normalisation with V = 10. (a) Error rates, plotted against ϕ, shown for four different users after normalisation. (b) ROC curves for the four writers considered in (a). The combined ROC curve (ave) is also shown. The ∘ indicates the point on the average ROC curve which was calculated by averaging the x's on each individual ROC curve. The vertical line in (a) indicates the threshold that corresponds to said x's.

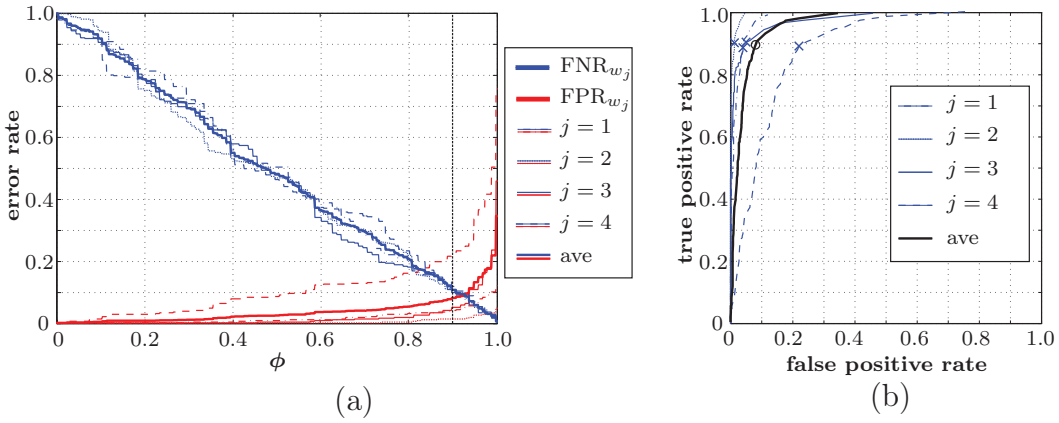


Figure 7.7: TPR-score normalisation with V = 100. (a) Error rates, plotted against ϕ, shown for four different users after normalisation. (b) ROC curves for the four writers considered in (a). The combined ROC curve (ave) is also shown. The ∘ indicates the point on the average ROC curve which was calculated by averaging the x's on each individual ROC curve. The vertical line in (a) indicates the threshold that corresponds to said x's.

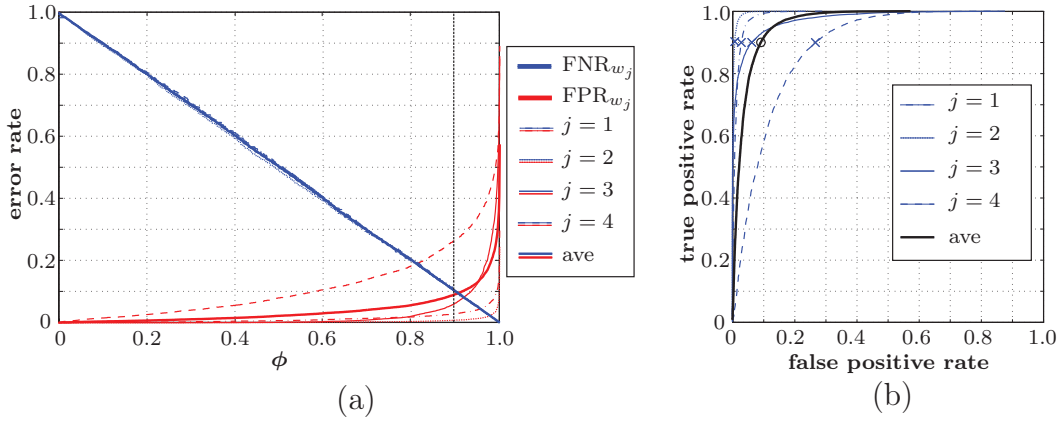


Figure 7.8: TPR-score normalisation in the "ideal" case. (a) Error rates, plotted against ϕ, shown for four different users after normalisation. (b) ROC curves for the four writers considered in (a). The combined ROC curve (ave) is also shown. The ∘ indicates the point on the average ROC curve which was calculated by averaging the x's on each individual ROC curve. The vertical line in (a) indicates the threshold that corresponds to said x's.

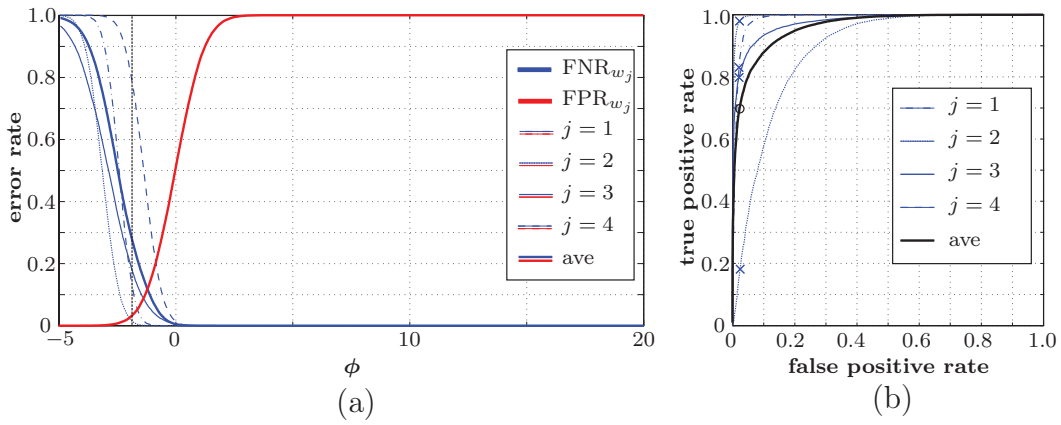
7.3.3 Negative samples

The score normalisation strategies described in the previous section, which were based on estimating the distribution of positive samples, can also be applied to the scenario when the distribution of negative samples is estimated.

Z_N-score normalisation. Z_N-score normalisation is the same as Z_P-score normalisation (as it is defined in Equation 7.3.2), except that the parameters μ_w and σ_w are estimated using *negative* samples. As this is similar to Z_P-score normalisation, we only show the "ideal" case ($V, E = 10,000$) for Z_N-score normalisation in Figure 7.9. Note that, after normalisation, the distribution of scores associated with *negative* samples has a mean of 0, and a standard deviation of 1. In ROC space (see Figure 7.9b), Z_N-score normalisation is equivalent to vertical averaging when $V, E \to \infty$. Vertical averaging, as it relates to ROC curves, is discussed in more detail in Fawcett (2006).

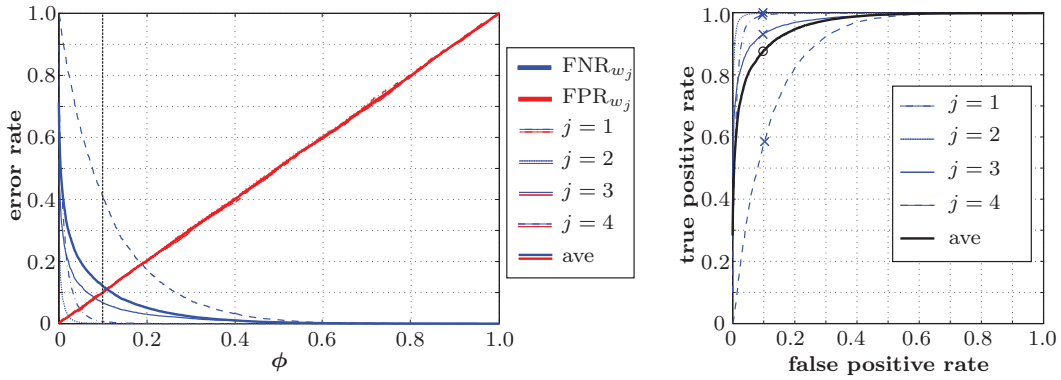
FPR-score normalisation. FPR-score normalisation can be considered instead of TPR-score normalisation, when only negative samples are available. FPR-score normalisation is similar to TPR-score normalisation, except that the distribution of negative samples is estimated. FPR-score normalisation has been proposed by Ross (2003) and successfully demonstrated in a fingerprint verification system. FPR-score normalisation has therefore been referred to as "Ross's Method" in the literature. Wang *et al.* (2008) has also, independently of Ross, suggested FPR-score normalisation (which they referred to as FAR-score normalisation). Figure 7.10 depicts the "ideal" case ($V, E = 10,000$) for FPR-score normalisation.

As in the case of score normalisation based on positive samples, Z_N-score normalisation and FPR-score normalisation (Ross's Method) are identical un-



(a) (b)

Figure 7.9: Z_N-score normalisation in the "ideal" case. (a) Error rates, plotted against ϕ, shown for four different users after normalisation. (b) ROC curves for the four writers considered in (a). The combined ROC curve (ave) is also shown. The ∘ indicates the point on the average ROC curve which was calculated by averaging the x's on each individual ROC curve. The vertical line in (a) indicates the threshold that corresponds to said x's.



(a) (b)

Figure 7.10: FPR-score normalisation in the "ideal" case. (a) Error rates, plotted against ϕ, shown for four different users after normalisation. (b) ROC curves for the four writers considered in (a). The combined ROC curve (ave) is also shown. The ∘ indicates the point on the average ROC curve which was calculated by averaging the x's on each individual ROC curve. The vertical line in (a) indicates the threshold that corresponds to said x's.

der ideal conditions ($V, E \to \infty$). The choice of score normalisation strategy should be based on the same criteria outlined towards the end of the previous section.

7.3.4 Positive and negative samples

We now briefly consider score normalisation strategies that are feasible when both positive and negative score distributions can be estimated for each user. Although both the following proposed schemes are possible by estimating continuous distributions (as in Z_P-score normalisation, for example), we only consider estimating discrete distributions in this section. Comparable results should be obtained when the positive and negative score distributions are Gaussian, and the parameters can be estimated accurately.

While the availability of either positive or negative samples allows us to estimate the relationship between ϕ and either the TPR or FPR respectively, estimating *both* positive and negative score distributions enables us to determine the relationship between ϕ and some property of *both* the FPR and TPR.

R-score normalisation. The first score normalisation strategy we introduce in this dissertation which assumes some knowledge of both the positive and negative score distributions, we shall call R-score normalisation (or Ratio-score normalisation). R-score normalisation is novel, as far as we know. Once the discrete score distributions of positive and negative samples have been estimated for each writer, a discrete function is defined for each user which maps the user's scores onto a domain ϕ where $\tan(\phi) = \frac{\text{TPR}(\tau)}{1-\text{FPR}(\tau)}$. This is equivalent to averaging points which have the same TNR:FPR ratio[6]. Figure 7.11 shows R-score normalisation with $V = 10$. In Figure 7.11b, the x's indicate the points which correspond to $\phi = 45°$. As the estimates improve, points corresponding to $\phi = 45°$ should approximate an EER, as $\tan(45°) = \frac{\text{TPR}}{1-\text{FPR}} = 1$. Figures 7.12 and 7.13 illustrate R-score normalisation for $V = 100$ and the "ideal" case ($V, E = 10,000$), respectively.

CH-score normalisation. CH-score (Convex hull-score) normalisation is best described with the aid of an illustration. Consider a radial line that emanates from the point (0,1) in ROC space, and forms an angle ϕ with the horizontal. Figure 7.14a and b show this line for the case when $\phi = 45°$ and $\phi = 15°$, respectively. A discrete, non-linear, transformation function is then computed which transforms each user's score distributions onto a domain $\phi \in [0°, 90°)$, so that the discrete classifiers associated with a specific value of ϕ (where $\phi \in [0°, 90°)$ will be the discrete classifier with the shortest distance to the radial line forming an angle ϕ with the horizontal. In Figure 7.14a and b, τ_1 and τ_2, indicate the points on the two ROC curves (each associated with a user) that will be associated with a threshold $\phi = 45°$ and $\phi = 15°$, respectively, after normalisation. The operating points labelled τ_1 and τ_2 will therefore

[6]Other ratios are also possible.

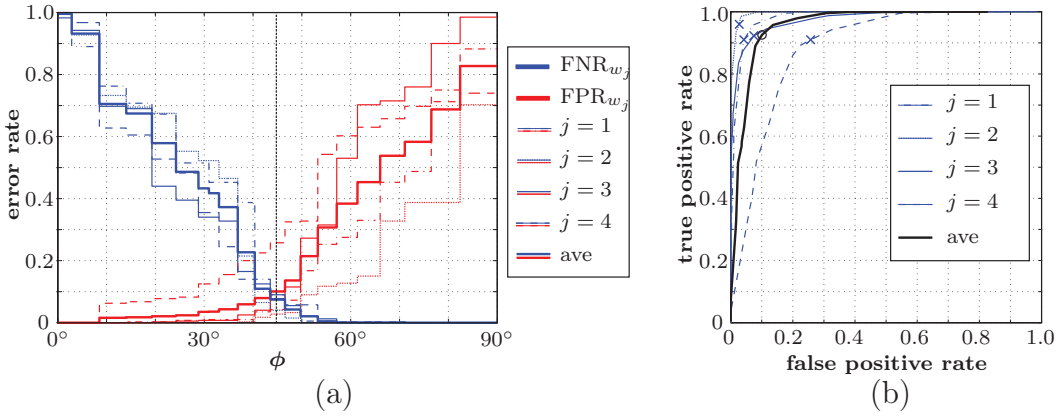


Figure 7.11: R-score normalisation with V = 10. (a) Error rates, plotted against ϕ, shown for four different users after normalisation. (b) ROC curves for the four writers considered in (a). The combined ROC curve (ave) is also shown. The ∘ indicates the point on the average ROC curve which was calculated by averaging the x's on each individual ROC curve. The vertical line in (a) indicates the threshold that corresponds to said x's.

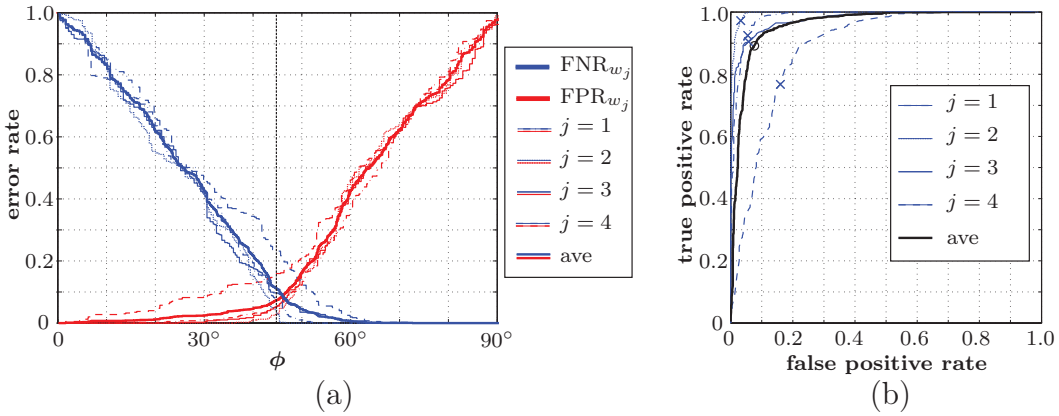


Figure 7.12: R-score normalisation with V = 100. (a) Error rates, plotted against ϕ, shown for four different users after normalisation. (b) ROC curves for the four writers considered in (a). The combined ROC curve (ave) is also shown. The ∘ indicates the point on the average ROC curve which was calculated by averaging the x's on each individual ROC curve. The vertical line in (a) indicates the threshold that corresponds to said x's.

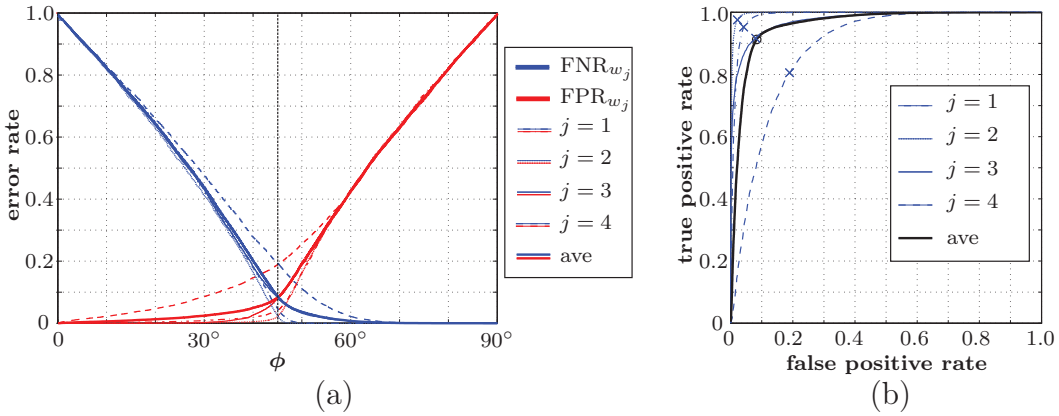


Figure 7.13: R-score normalisation in the "ideal" case. (a) Error rates, plotted against ϕ, shown for four different users after normalisation. (b) ROC curves for the four writers considered in (a). The combined ROC curve (ave) is also shown. The ∘ indicates the point on the average ROC curve which was calculated by averaging the x's on each individual ROC curve. The vertical line in (a) indicates the threshold that corresponds to said x's.

be averaged to form a point on the averaged ROC curve, when CH-score normalisation is applied. Note that CH-score normalisation only considers operating points which lie on the convex hull of each classifier's ROC curve.

CH-score normalisation, in the "ideal" case, will yield the best possible combined ROC curve. The ROC curve will have the maximum attainable AUC, and will also dominate ROC curves generated from any other possible score normalisation method, for all regions in ROC space.

7.3.5 Performance comparison

In the above experiments, only four classifiers (or users) were considered– this enabled us to clearly illustrate how each averaged ROC curve was constructed. However, four classifiers is insufficient to allow a fair comparison of the effect on combined performance for the different score normalisation strategies. Figure 7.18 shows the combined ROC curves which were generated using different score normalisation strategies, for fifty different classifiers, in the *ideal* case. It is clear that CH-norm produces the best performing ROC curve (largest AUC), and this will always be the case, given that sufficient samples are available to estimate *both* the positive and negative score distributions accurately. It is not reasonable to make any other conclusions about the relative performances from these experiments, as the performances will depend entirely on the type of distributions encountered in practice, as well as the number of samples available to estimate the distributions. Z_P-score normalisation and Z_N-score normalisation, perform as well as TPR-score and FPR-score normalisation respectively, but this is to be expected, since each generic classifier in

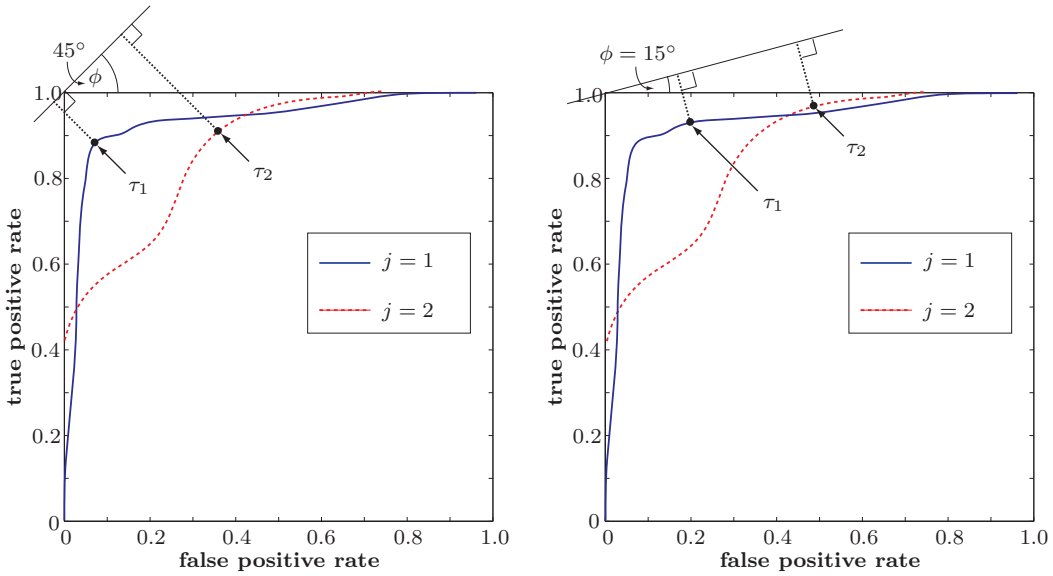


Figure 7.14: CH-score normalisation. (a-b) The discrete classifiers, indicated by τ_1 and τ_2 associated with $\phi = 45°$ and $\phi = 15°$, respectively.

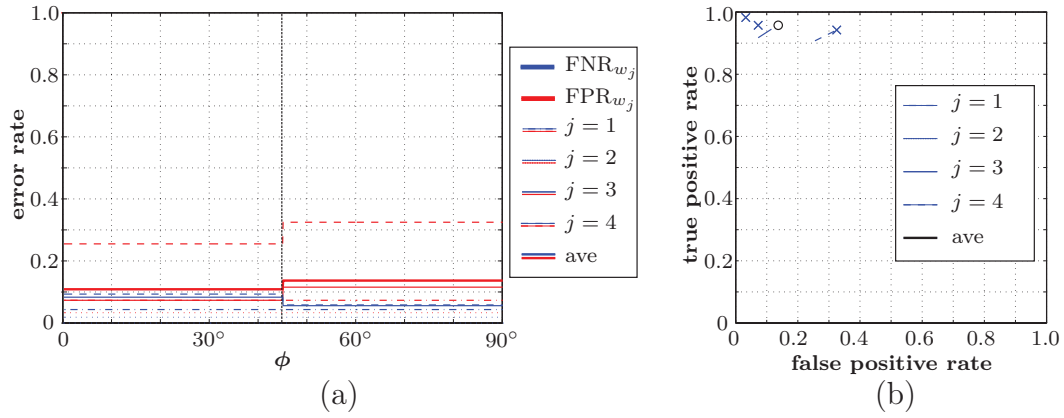


Figure 7.15: CH-norm with V = 10. (a) Error rates, plotted against ϕ, shown for four different users after normalisation. (b) ROC curves for the four writers considered in (a). The combined ROC curve (ave) is also shown. The ∘ indicates the point on the average ROC curve which was calculated by averaging the x's on each individual ROC curve. The vertical line in (a) indicates the threshold that corresponds to said x's.

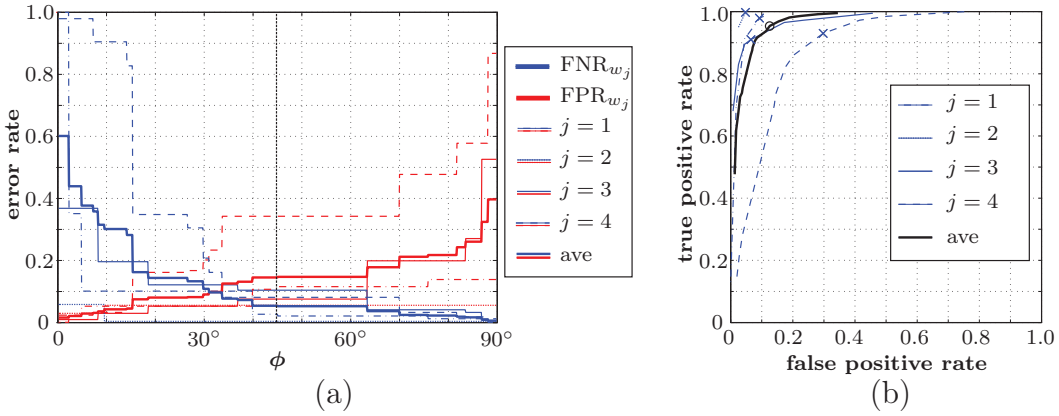


Figure 7.16: CH-norm with V = 100. (a) Error rates, plotted against ϕ, are shown for four different users after normalisation. (b) ROC curves for the four writers considered in (a). The combined ROC curve is also shown. The ○ indicates a point on the average ROC curve which was calculated by averaging the x's on each ROC curve.

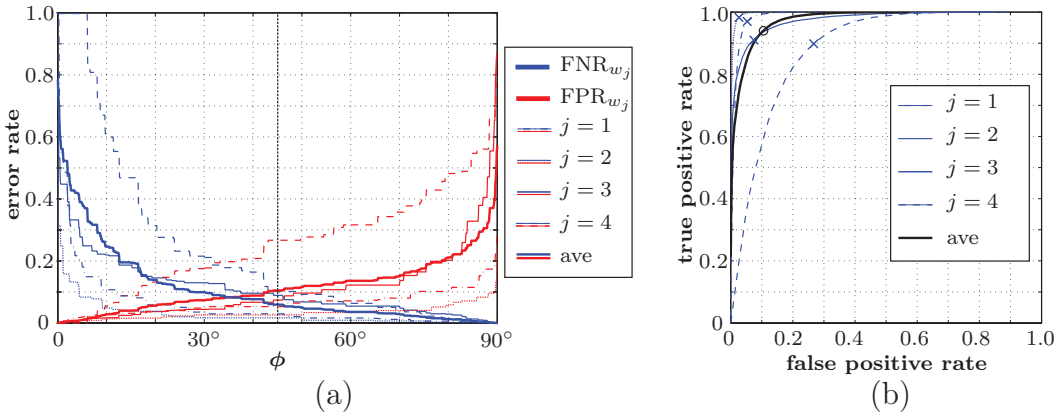


Figure 7.17: CH-norm in the "ideal" case. (a) Error rates, plotted against ϕ, shown for four different users after normalisation. (b) ROC curves for the four users considered in (a). The combined ROC curve (ave) is also shown. The ○ indicates the point on the average ROC curve which was calculated by averaging the x's on each individual ROC curve. The vertical line in (a) indicates the threshold that corresponds to said x's.

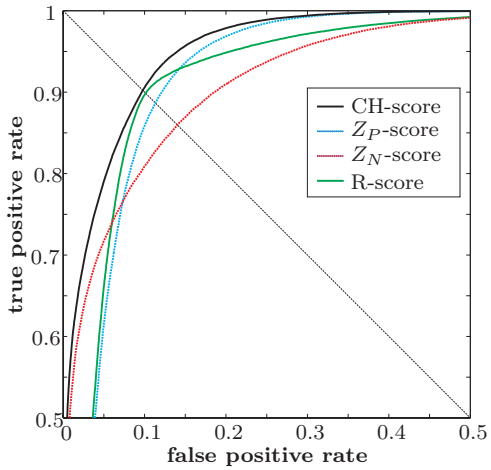


Figure 7.18: ROC curves generated using different score normalisation strategies (in the ideal case) on the same fifty generic classifiers. CH-score normalisation clearly has the best performance. Note that the average ROC curve generated when utilising FPR-score normalisation and TPR-score normalisation will be identical to the average ROC curves resulting from Z_N-score and Z_P-score normalisation, respectively, and are therefore omitted from the plot.

these experiments was predefined to have Gaussian score distributions.

7.4 Operational considerations

While it is clear from the above examples that CH-score normalisation will produce the best overall system performance when *both* positive and negative distributions can be estimated for each writer, there are operational considerations which may influence the decision as to which score normalisation strategy should be used. These operational constraints should be considered in addition to the practical constraints outlined in Section 7.3.1.

Consider, for example, a scenario in which a bank manager imposes a maximum allowable FPR of 0.1 for a signature verification system. A criterion in which a maximum FPR is imposed is equivalent to the so-called Neyman-Pearson criterion (see Fawcett (2006)), and is used widely in hypothesis testing, as well as in verification systems. If it is desirable that this criterion is imposed on *each* writer, it is essential that a score normalisation strategy based on *negative* score distributions is utilised. That is, either Ross's Method or Z_N-score normalisation should be employed. While this may not necessarily maximise the combined system performance, it does ensure that each classifier (or writer) operates with the same imposed criterion, as far as possible. Consider also a scenario in which it is preferred that each user experiences an EER. In this case, R-score normalisation should be used.

Alternatively, if it is required that the system *as a whole* operates with

a certain imposed criterion, and no consideration needs to be made for the operating points of individual classifiers, then the choice of score normalisation strategy should be based solely on maximising combined performance.

7.5 Score normalisation in this dissertation

Since the objective of this dissertation is to detect skilled forgeries, it is not possible to estimate the score distribution of negative signatures for each enrolled client (writer). We are therefore limited to score normalisation strategies which rely on only an estimation of the score distribution of positive samples. Due to the limited training data available in the data set used in this dissertation, and prior knowledge that the score distributions are approximately Gaussian, we implement Z_P-score normalisation, as follows.

The mean dissimilarity value of the training samples for retina r, associated with writer w, is denoted by μ_w^r and calculated as follows,

$$\mu_w^r = \frac{1}{N_T} \sum_{i=1}^{N_T} D(\mathbf{X}_{w,i}^r, \lambda_w^r). \tag{7.5.1}$$

The standard deviation of the dissimilarity values of the training samples for retina r, associated with writer w, is denoted by σ_w^r and calculated as follows,

$$\sigma_w^r = \sqrt{\frac{1}{N_T - 1} \sum_{i=1}^{N_T} (D(\mathbf{X}_{w,i}^r, \lambda_w^r) - \mu_w^r)^2}. \tag{7.5.2}$$

The above statistics are calculated using the N_T (positive) training signatures available for each writer (that is, set $\boldsymbol{T}_O^+$ and $\boldsymbol{T}_E^+$). Combining Equation 7.3.1 and 7.3.2 produces the acceptance inequality

$$\frac{D(\mathbf{X}_{(w)}^r | \lambda_w^r) - \mu_w^r}{\sigma_w^r} < \phi \tag{7.5.3}$$

or

$$D(\mathbf{X}_{(w)}^r | \lambda_w^r) < \phi \cdot \sigma_w^r + \mu_w^r, \tag{7.5.4}$$

that is, retina r of writer w is classified as positive (accepted) if the above inequality is true, otherwise it is classified as negative (rejected).

7.6 Threshold parameter calibration

7.6.1 Notation

In Chapter 4 we defined λ_w^r as the HMM which models retina r of writer w. We now introduce a new notation, $C_w^r\{\sim\}$, which denotes the *continuous* classifier

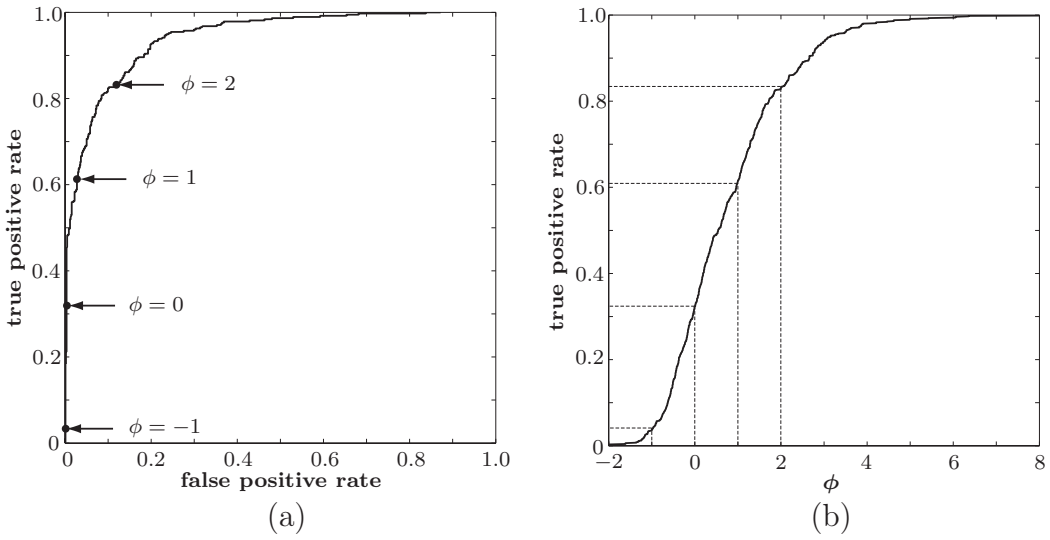


Figure 7.19: (a) Average ROC curve for all writers in an evaluation set, using Z_P-score normalisation. The threshold parameter ϕ associated with several discrete classifiers is indicated. (b) The relationship between ϕ and the TPR for the ROC curve in (a).

associated with retina r of writer w. Note that $C_w^r\{\sim\}$ is equivalent to λ_w^r, as an HMM is also considered a generative continuous classifier. We use the notation $C_w^r\{\sim\}$ when referring to the classifier, and λ_w^r, when referring to the model. The symbol $C_w^r\{\phi\}$, is used to denote the *discrete* classifier obtained from the continuous classifier C_w^r by imposing the threshold ϕ (see Equation 7.5.4). When the writer is not specified, as in $C^r\{\phi\}$, the classifier used to evaluate retina r for *all* writers (using Z_P-score normalisation) is implied. Finally, the single classifier (which results from the combination of base classifiers, see Chapter 8) used to evaluate all writers is denoted by $C\{\sim\}$.

7.6.2 Interpretation of ϕ

A convenient spin-off of each normalisation strategy is the calibration of the parameter ϕ. For example, when Ross's method of score normalisation is applied, ϕ is calibrated with the FPR, that is, the discrete classifier $C_w^r\{0.1\}$, for example, should operate with a FPR of approximately 0.1. Similarly, when R-score normalisation is used, the discrete classifier $C_w^r\{45°\}$ should operate with an EER.

Figure 7.19a shows the average ROC curve, using Z_P-score normalisation, for all the writers in an evaluation set. When Z_P-score normalisation is used, the parameter ϕ should correlate well with the TPR (the relationship between ϕ and the TPR is shown in Figure 7.19b). For example, the classifier $C_w^r\{0\}$ should operate with a TPR of approximately 0.5, as approximately half of all positive signatures will be accepted. Furthermore, the classifier $C_w^r\{1\}$ should accept $50\% + 34.1\%$ (1 standard deviation) of all positive signatures (that is, a

TPR of 0.841). The same should be true for the classifiers $C^r\{0\}$ and $C^r\{1\}$, respectively.

From the ROC curve and the labelled discrete classifiers depicted in Figure 7.19, it is clear that this is not quite the case. This can be explained in a number of ways: (1) The assumption that the positive score distribution for each writer is Gaussian is not valid. (2) The small amount of positive signatures available is insufficient to estimate the Gaussian distributions accurately, and perhaps most significantly, (3) the signatures used to estimate the score distributions were also used to train the classifier, which results in an "overfitting" or an optimistic bias in the estimated score distributions. Explanation (3) is given some credibility by noting that $\phi = 0$ results in a TPR of approximately 0.32, instead of the expected TPR of 0.5.

7.6.3 Threshold parameter transformation: $\phi \mapsto \rho$

In addition to the un-normalised threshold parameter τ and the normalised threshold parameter ϕ, we now define a new *calibrated* threshold parameter ρ. While it is not possible to calibrate the parameter of an individual classifier $C^r_w\{\sim\}$ accurately (other than what follows from the normalisation strategy used), it is possible to calibrate the threshold parameter of the classifier $C^r\{\sim\}$ with a high degree of accuracy using the optimisation set.

This is useful in many scenarios. In the signature verification system developed in this dissertation Z_P-score normalisation is used, which implies that the parameter ϕ is related to a classifier's TPR, as shown in Figure 7.19. However, it may also be desirable to have a parameter which predicts either the FPR, or a FPR:FNR ratio (like the EER). This enables an operator (for example, a bank manager) to intuitively select a threshold parameter that will result in the desired operating criterion. For example, if an operator wishes a signature verification system to operate with an FPR of 0.1, he/she can simply select a threshold of $\rho = 0.1$.

We now illustrate the process of threshold parameter transformation using an example. Figure 7.20a shows a discrete function $G_{\text{FPR}} : \phi \mapsto \rho$. In this case, the function G_{FPR} is such that ρ is calibrated with the FPR. This function is calculated by determining the relationship between ϕ and the FPR in the optimisation set. Figure 7.20b shows the ROC curve for the *optimisation* set (which is the same ROC curve shown in Figure 7.19a) with several discrete classifiers indicated. Note that the value of ρ is now equivalent to the FPR.

We can now modify Equation 7.5.4 to become

$$D(\mathbf{X}^r_{(w)}|\lambda^r_w) < G^{-1}_{\text{FPR}}(\rho)(\sigma^r_w) + \mu^r_w, \tag{7.6.1}$$

When this modified Z_P-score normalisation equation is used, we now have a parameter ρ which is related to the FPR. Figure 7.20c shows the ROC curve for the evaluation set (that is, a different set of writers than those used to

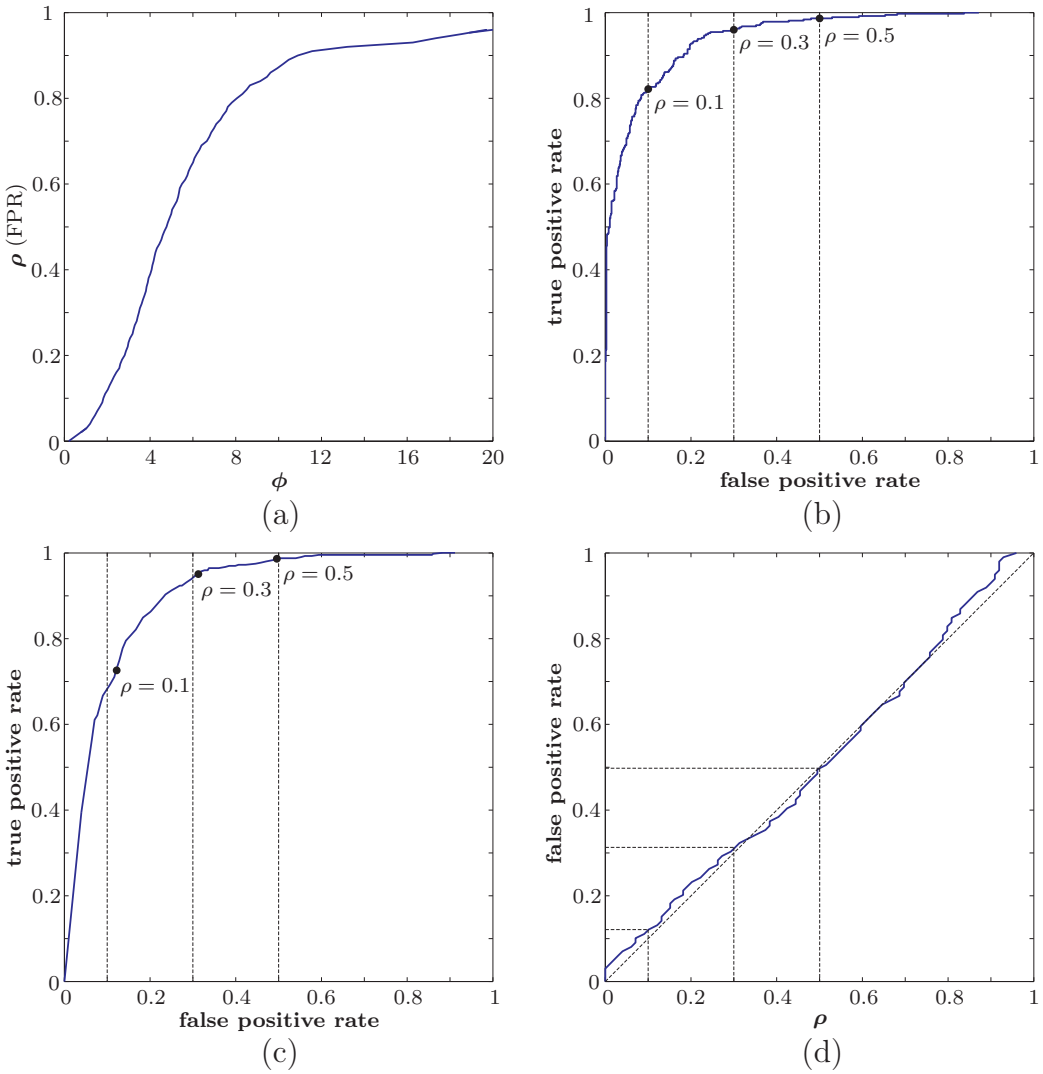


Figure 7.20: Threshold parameter calibration with the FPR. (a) The discrete function G_{FPR}, determined using an optimisation set, which maps ϕ onto ρ, where ρ is equivalent to the FPR. (b) Several discrete classifiers and their associated threshold parameter ρ, on the *optimisation* set. Since the function G_{FPR} was calculated using the *optimisation* set (ie., the same set of writers), ρ is calibrated perfectly with the the FPR. (c) The function G_{FPR} (shown in (a)) applied to an *evaluation* set (ie., a different set of writers). The parameter ρ is now an accurate predictor of the FPR. (d) The error between ρ and the actual FPR for the *evaluation* set used to generate the ROC curve in (c). Note that the straight line $\mathrm{FPR} = \rho$ depicts perfect mapping obtained when using the optimisation set (as shown in (b)).

calculate the function G_{FPR}). Note that the threshold parameter ρ is now an accurate predictor of the FPR, even though Z_P-score normalisation is used[7]. Figure 7.20d depicts the error between ρ and the actual FPR achieved. The discrepancy between ρ and the FPR will become negligible as the size of both the evaluation and optimisation sets increase.

Figure 7.21 illustrates the same process shown in Figure 7.20, except that a function G_R is calculated, which relates ρ to the ratio $\tan\left(\frac{\text{TPR}}{1-\text{FPR}}\right)$.

Further advantages of using a calibrated threshold parameter, ρ, will become clear in Chapter 8, where classifier combination is considered.

7.7 Conclusion

In Section 7.3.4 we showed that the optimal normalisation scheme, in terms of combined system performance (AUC of the combined ROC curve), is CH-norm, however, we also stated there are typically not enough positive *and* negative samples available at the time of enrolment. It may be possible, in practice, to initialise a system using a certain normalisation strategy, and then adapt and re-estimate the distributions over time, as more questioned signatures become available[8]. CH-norm may therefore have value in real world scenarios, although it is not possible to implement and test on the small signature data set available.

It is worth emphasising that, in our system, there are N_r classifiers associated with each writer. Z_P-score normalisation is therefore applied to each of the classifiers associated with each retina. In the next chapter we discuss how these classifiers are selected and combined.

[7]Note that, if Ross's Method were used, the parameter ϕ would be a predictor of the FPR by default.

[8]It would be sensible in this scenario to only label each new signature after sufficient time has passed to detect FPs and FNs.

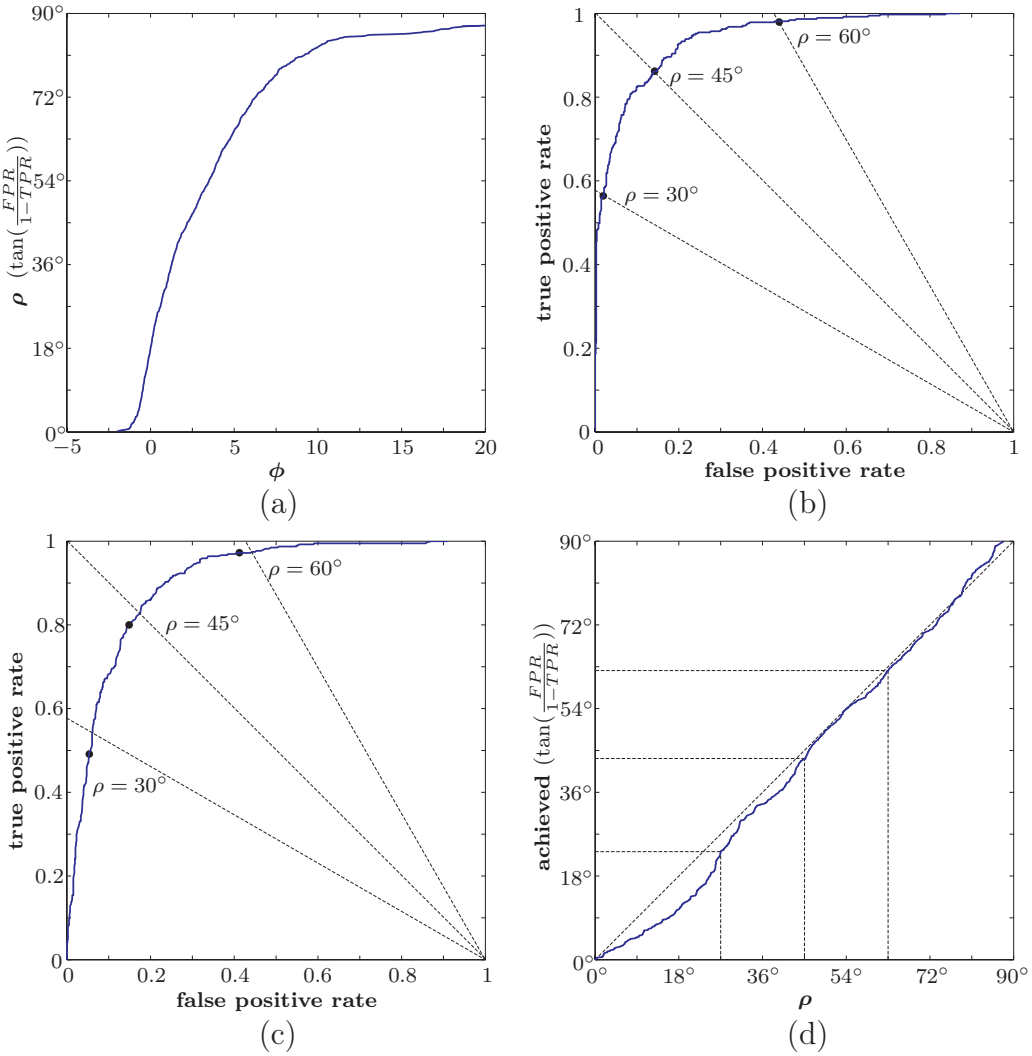


Figure 7.21: Threshold parameter calibration with the TPR:TNR ratio. (a) The discrete function G_{R}, determined using an optimisation set, which maps ϕ onto ρ. (b) Several discrete classifiers and their associated threshold parameter ρ, on the *optimisation* set. (c) The function G_{R}, shown in (a), applied to an evaluation set. The parameter ρ is now an accurate predictor of the TPR:TNR ratio. (d) The error between ρ and the actual TPR:TNR ratio for the evaluation set shown in (c).

Chapter 8

Ensemble Selection and Combination

8.1 Background and key concepts

In Chapter 6, we introduced the concept of a *classifier*. The reader is reminded of the distinction between *continuous* and *discrete* classifiers: a continuous classifier is constructed from a generative model using a sliding threshold, while a discrete classifier is obtained from a continuous classifier by imposing a specific threshold. The performance of a continuous classifier is depicted by a curve in ROC space, while the performance of a discrete classifier is depicted by a single point (that is, an operating point) in ROC space (see Section 6.4).

Classifier *fusion* is the process of combining individual classifiers, in order to construct a single classifier which is more accurate, albeit more computationally complex, than its constituent parts. A *combined* classifier therefore consists of an *ensemble* of classifiers that are combined using a specific *fusion* strategy. A broad overview of fusion strategies is provided in Section 8.2. The above-mentioned individual classifiers that constitute a classifier ensemble are referred to as *base* classifiers, and can be either continuous or discreet. A combined classifier is therefore defined by an ensemble of base classifiers *and* a specific fusion strategy.

In this dissertation we employ *ensemble generation* techniques in order to produce a pool of *candidate* ensembles. The performance of each combined classifier (that is, a classifier constructed from each candidate ensemble by combining the constituent base classifiers using a specific fusion strategy) is evaluated using the optimisation set $\boldsymbol{O}$, after which the most proficient combined classifier is selected.

All the ROC curves in this chapter (which are used for illustrational purposes) depict the performance of classifiers using writers in the optimisation set $\boldsymbol{O}$.

Ensemble selection therefore refers to the process of *selecting* a candidate

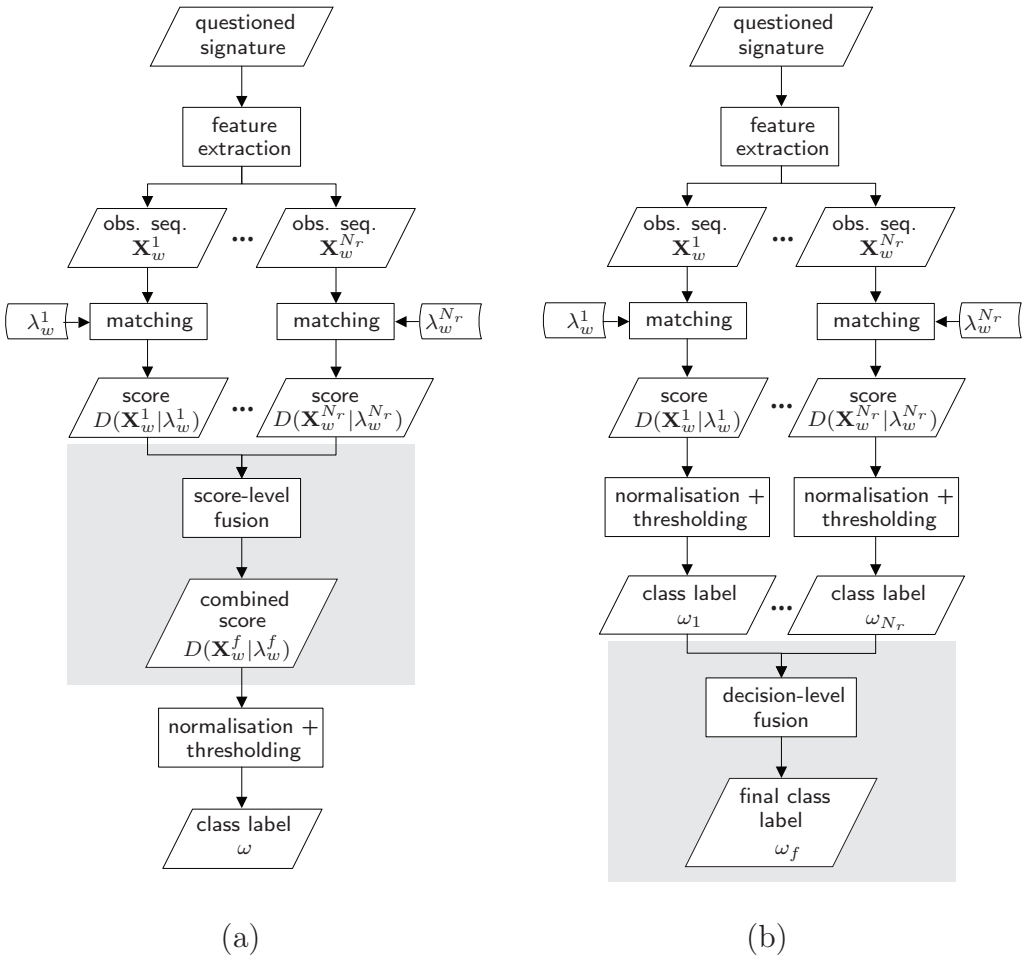


Figure 8.1: Classifier fusion. (a) Score-level fusion, and (b) decision-level fusion as they apply to the signature verification system developed in this dissertation.

ensembles that are optimal for certain operating criteria.

8.2 Fusion strategies

Classifier fusion can be employed at two fundamentally different levels, namely at the score level (score-level fusion) or at the decision level (decision-level fusion). A general, brief discussion of each fusion strategy is provided, as well as an outline of how each strategy could be implemented in a signature verification system.

Note that only decision-level fusion (fusion of label outputs) is implemented in this dissertation, but a brief discussion on score-level fusion is also provided here to create some perspective.

8.2.1 Score-level fusion

Figure 8.1a depicts score-level fusion, as it could be applied to the system developed in this dissertation. Multiple observation sequences are extracted from a questioned signature during feature extraction (see Chapter 3). Each observation sequence is then compared with the relevant trained HMM during the matching step, to produce a score/dissimilarity value $D_w^r = D(\mathbf{X}_w^r|\lambda_w^r)$ (see Equation 4.3.7). These scores are then combined to form a combined "score" D_w^f, where

$$D_w^f = \Upsilon(D_w^1, D_w^2, \ldots, D_w^{N_r}) \tag{8.2.1}$$

which is obtained by combining the dissimilarity values D_w^r for $r = 1, 2, \ldots, N_r$. The symbol $\Upsilon(.)$ denotes a fusion function. A threshold is then imposed on this combined score/dissimilarity value, after normalisation, in order to label a questioned signature as genuine or fraudulent.

Examples of score-level fusion strategies include the so-called "simple mean combination", in which the average of all the base scores is calculated, as well as "weighted mean", "trimmed mean" and "product", amongst others (see Kuncheva (2004)).

8.2.2 Decision-level fusion

Decision-level fusion differs from score-level fusion in that thresholding is applied to each score/dissimilarity value D_w^r *before* fusion occurs. A final decision as whether to accept or reject a questioned signature is therefore made by considering the *decisions* made by the individual base classifiers. A final class label, denoted by ω_f, is defined as

$$\omega_f = \Upsilon(\omega_1, \omega_2, \ldots, \omega_{N_r}) \tag{8.2.2}$$

where $\Upsilon(.)$ again denotes a fusion function.

Decision-level fusion, as it applies to this dissertation, is illustrated in Figure 8.1b.

Although score-level fusion is generally considered superior to decision-level fusion−once a threshold has been imposed, no information about the confidence of the label is retained−we only implement a decision-level fusion strategy.

Examples of decision-level fusion strategies include "majority voting", "weighted majority voting" (Kuncheva (2004)), "Haker's Algorithm" (Haker *et al.* (2005)), and "Iterative Boolean Combination" (Khreich *et al.* (2010)). We now consider the decision-level future strategy utilised in this dissertation, that is *majority voting*, in more detail.

Majority voting We use the convention of 1 to denote a positive label (that is an acceptance) and 0 to denote a negative label (that is a rejection). Given

a set of N class labels ω_i, $i = 1, \ldots, N$, the final decision, denoted by ω_f, based on majority voting is obtained as follows,

$$\omega_f = \begin{cases} 1, & \text{if } \sum_{i=1}^{N} \omega_i \geq \lceil \frac{N+1}{2} \rceil \\ 0, & \text{otherwise} \end{cases} . \tag{8.2.3}$$

In other words, the final label, ω_f, is positive if at least half (that is, the *majority*) of the individual decisions are positive.

8.3 Ensemble generation

In this dissertation we utilise a decision level fusion strategy (majority voting), which is only applicable to the fusion of *discrete* classifiers. We therefore consider each continuous classifier to be a finite set of discrete classifiers, where each discrete classifier corresponds to a specific imposed threshold. Formally, we denote the number of discrete classifiers associated with each continuous classifier by X, where X is equal to the number of discrete values selected for the threshold parameter. We therefore have a pool of $N_r \cdot X$ discrete classifiers, obtained from a set of N_r continuous classifiers (one associated with each retina). In Figure 8.2a, the performance of $N_r = 5$ continuous classifiers are shown. Figure 8.2b depicts the discrete classifiers obtained from said continuous classifiers by imposing $X = 50$ discrete threshold values. In this example, we therefore have a pool of 250 discrete classifiers from which to construct ensembles.

The success of a combined classifier is primarily determined by the independence of the base classifiers in the ensemble (that is, the base classifiers should make conditionally independent errors). Since discrete classifiers associated with the same continuous classifier invariably make *dependent* errors, it does not make sense to construct any ensembles which contain more than one discrete classifier associated with the same retina. The total number of possible ensembles of size N_S, $N_S \in \{1, \ldots, N_r\}$, denoted by T_{N_S} is therefore given by

$$T_{N_S} = \binom{N_r}{N_S} \cdot X^{N_S}, \tag{8.3.1}$$

where $\binom{N_r}{N_S}$ denotes the binomial coefficient[1] which is defined as follows,

$$\binom{N_r}{N_S} = \frac{N_r!}{N_S!(N_r - N_S)!}. \tag{8.3.2}$$

[1]The binomial coefficient $\binom{x}{y}$ gives the number of x-combinations of a y-element set.

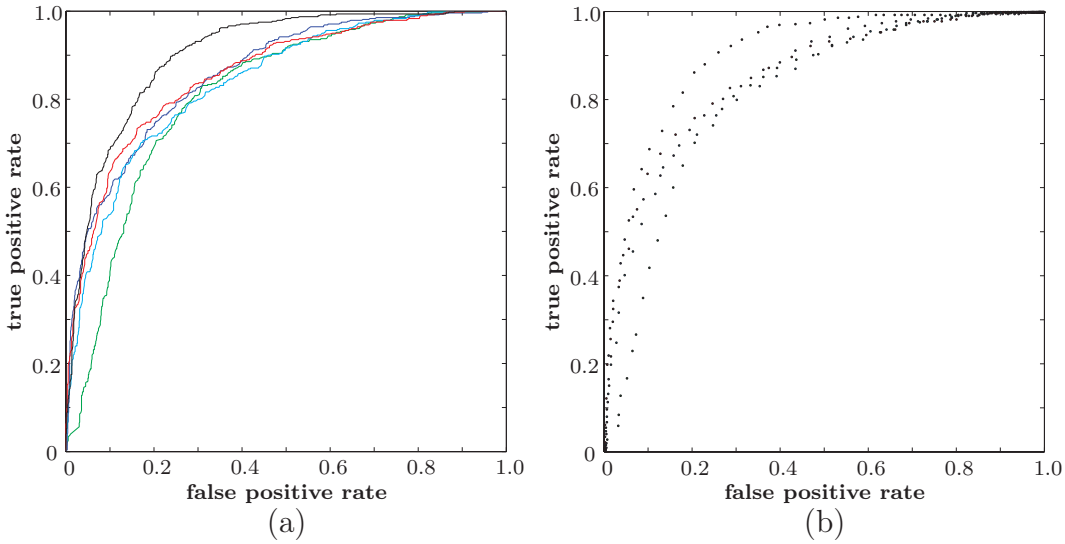


Figure 8.2: (a) The performance of $N_r = 5$ continuous classifiers in ROC space. (b) A pool of $N_r \cdot X$, in this case $5 \cdot 50 = 250$, discrete classifiers, obtained from the continuous classifiers in (a) by imposing $X = 50$ threshold values on each continuous classifier.

8.3.1 Exhaustive ensemble generation

It is easy to see that, for even modest values of N_r, N_S and X, the number of possible candidate ensembles, denoted by Ω_{ex}, is prohibitively large,

$$\Omega_{\text{ex}} = T_{N_S} = \binom{N_r}{N_S} \cdot X^{N_S}. \tag{8.3.3}$$

Since the performance of the combined classifier associated with each candidate ensemble needs to be evaluated, an exhaustive approach is not feasible. For example, assuming values of $N_r = 10$ (10 retinas), $N_S = 5$ (ensemble size of 5), and $X = 100$ (100 discrete classifiers per continuous classifier), the total number of candidate ensembles to evaluate is 2.52×10^{22}. We therefore need to constrain the number of ensembles that are generated and evaluated.

8.3.2 Performance-cautious ensemble generation

In order to limit the total number of ensembles considered, we only combine discrete classifiers associated with the same threshold parameter value. Since we use Z_P-score normalisation (see Chapter 7), discrete classifiers associated with the same threshold parameter ϕ should approximately lie on the same horizontal line in ROC space (that is, have the same TPR). Figure 8.3a depicts discrete classifiers obtained from five different continuous classifiers. Discrete classifiers associated with the same threshold value ϕ are indicated, for several selected values of ϕ. While it is clear from Figure 8.3a that discrete classi-

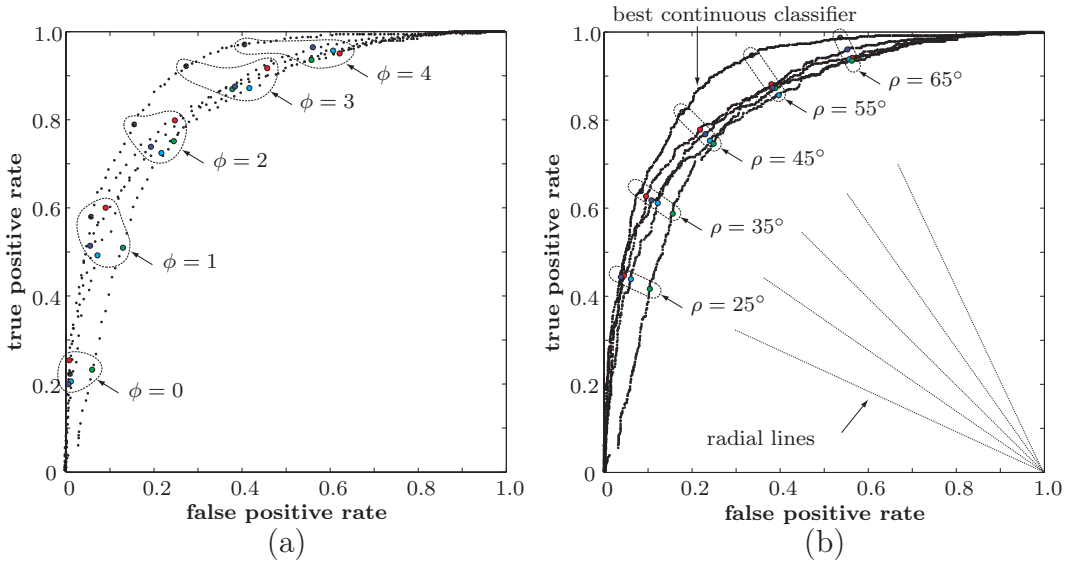


Figure 8.3: Threshold parameter transformations applied to classifier combination. (a) Discrete classifiers associated with 5 continuous classifiers, with $X = 100$. The discrete classifiers associated with several selected values of τ are indicated. (b) Discrete classifiers associated with the same continuous classifiers in (a). The threshold parameter has been transformed ($\phi \mapsto \rho$) so that discrete classifiers with the same TPR:1-FPR ratio are associated with the same threshold value ρ. The discrete classifiers associated with several selected values of ρ are indicated.

fiers associated with the same value of ϕ have approximately the same TPR[2], combining discrete classifiers associated with the same TPR is not necessarily optimal.

By employing threshold parameter transformations ($\phi \mapsto \rho$), as discussed in Section 7.6, we are afforded greater control over which discrete classifiers are combined (by manipulating which discrete classifiers are associated with a certain threshold value). A further benefit of threshold parameter transformations is the ability to control the distribution of discrete classifiers in ROC space. Note that in Figure 8.3a, the density of discrete classifiers increases towards the upper-righthand region of ROC space, despite the fact that the selected values of ϕ are evenly distributed. This is a direct consequence of using Z_P-score normalisation. This is undesirable though, since classifiers with very high FPRs are seldom employed in practice. By using a threshold parameter ρ, and a suitable transformation function G_R^{-1}, we are able to control the distribution of discrete classifiers in ROC space. Note that in Figure 8.3b, discrete classifiers are distributed at uniform angles in ROC space.

We are therefore able to combine discrete classifiers associated with the same transformed threshold parameter ρ, where $\rho \in (0°, 90°)$ at X evenly distributed intervals.

[2]Several reasons as to why the TPRs are not exactly the same are given in Section 7.6.

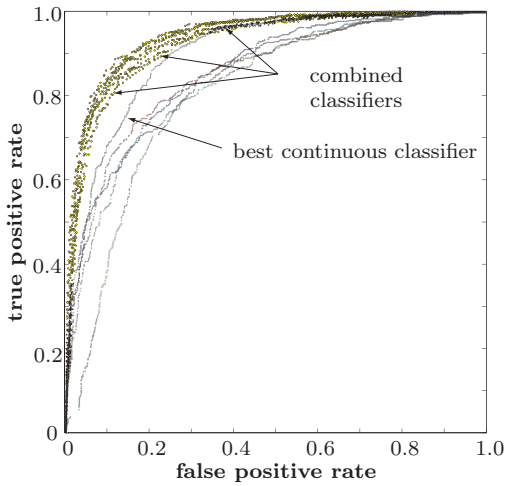


Figure 8.4: 9,000 candidate classifiers generated using the performance-cautious approach. The discrete base classifiers are also shown for comparison.

We now discuss and demonstrate an ensemble generation technique that is constrained in this way. By considering only ensembles that contain discrete classifiers associated with the same threshold value, the total number of possible candidate ensembles, denoted by Ω_{pc}, is reduced to

$$\Omega_{\text{pc}} = \binom{N_r}{N_S} \cdot X. \tag{8.3.4}$$

The above equation is based on the fact that, for each threshold value, we have a set of N_r discrete classifiers from which all possible combinations of N_S elements are considered. For the case when $N_r = 10$, $N_S = 5$ and $X = 100$, $\Omega_{\text{pc}} = 2.52 \times 10^4$, which is many orders of magnitude smaller than $\Omega_{\text{ex}} = 2.52 \times 10^{22}$.

Example. Performance-cautious ensemble generation is now demonstrated for the example classifiers, of which the performance is depicted in Figure 8.3. Since $N_r = 5$, $N_S = 3$ and $X = 900$ we therefore consider the performance of $\binom{5}{3} \cdot 900 = 9,000$ different combined classifiers, each formed by the fusion of the decisions of a candidate ensemble's classifiers using majority voting. The performances of these combined classifiers are shown in Figure 8.4. Note that the majority of the combined classifiers perform significantly better than the single best continuous classifier.

Although the ensemble generation technique introduced in this section is significantly more computationally efficient than an exhaustive search, large values of N_r will still yield a very high number of ensembles. In these scenarios, evaluating the performance of each of the Ω_{pc} combined classifiers may still not be computationally feasible.

In the next section, we introduce another ensemble generation method that is even more constrained, and therefore significantly more computationally

efficient than the method introduced in this section. We therefore refer to said method as efficiency-cautious constrained ensemble generation.

8.3.3 Efficiency-cautious ensemble generation

It is possible to make the ensemble generation technique described in the previous section significantly more efficient by limiting the number of continuous classifiers from which discrete classifiers are obtained. More specifically, one can consider only the best performing N_S continuous classifiers (out of a total of N_r), while the other continuous classifiers are completely discarded. No ensembles are therefore generated that contain discrete classifiers associated with the less proficient continuous classifiers. Performance, in this case, is evaluated by calculating the AUC (see Section 6) for each continuous classifier in ROC space. For each of the X threshold values considered, only a single ensemble is therefore generated. The total number of ensembles for this efficiency-cautious approach, denoted by Ω_{ec}, is therefore given by

$$\Omega_{\text{ec}} = X. \tag{8.3.5}$$

Note that we again employ threshold parameter transformations, as described in the previous section, so that we obtain X evenly distributed values of the transformed threshold parameter ρ. It is clear that $\Omega_{\text{ec}} \ll \Omega_{\text{pc}}$ for large values of N_S. This method requires the additional step of calculating the AUC for N_r continuous classifiers. However, the computational complexity of calculating the AUC of a continuous classifier is insignificant compared to the computational complexity of evaluating the performances of all the additional combined classifiers that are generated using the performance-cautious approach. While it may seem sensible to only consider the ensembles that contain the most proficient discrete classifiers, the reader is reminded that *independence* amongst base classifiers ("independent" classifiers make conditionally independent errors) is essential to the success of a combined classifier. By generating ensembles that contain only the most proficient discrete classifiers, the independence of the discrete classifiers in the ensemble is not taken into consideration.

Example. We now illustrate this method, by considering the same $N_r = 5$ classifiers and data used to illustrate the performance-cautious approach in the previous section, where an ensemble of size $N_S = 3$ is selected. Figure 8.5a shows the performance of said continuous classifiers (evaluated on the optimisation set). The AUCs of the five continuous classifiers are shown in Table 8.1.

The continuous classifiers associated with retinas 2 and 5 have the smallest AUCs, and are therefore discarded. We therefore only generate ensembles that contain discrete classifiers associated with retinas 1, 3 and 4. Figure 8.5b shows the performance of the discrete classifiers from which the ensembles are generated. Ensembles are generated as described in the previous section, that

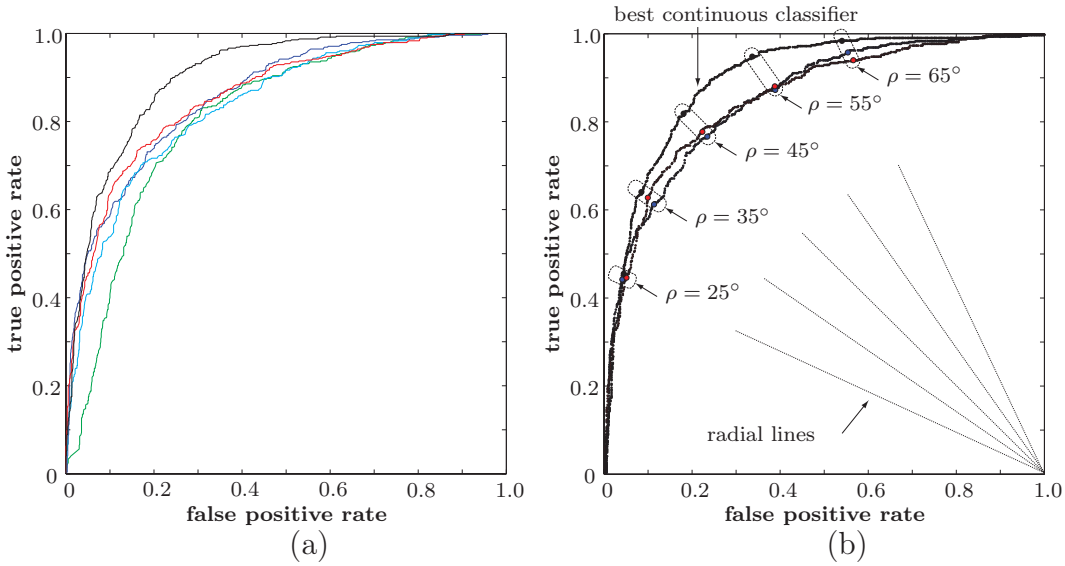


Figure 8.5: (a) Five continuous classifiers, each associated with a different retina. Since $N_S = 3$, in this example, the 2 continuous classifiers with the smallest AUCs are discarded. (b) Discrete classifiers associated with the $N_S = 3$ best performing classifiers. The discrete classifiers associated with several selected values of ρ are indicated. Note that only one ensemble is associated with each value of ρ.

Retina r	AUC
1	**0.8588**
2	0.8113
3	**0.8578**
4	**0.9037**
5	0.8382

Table 8.1: The AUCs of the continuous classifiers (each associated with a different retina r), of which the ROC curves are shown in Figure 8.5a. The $N_S = 3$ most proficient continuous classifiers, that is the classifier associated with retinas 1, 3 and 4, are selected, while the classifiers associated with retinas 2 and 5 are discarded.

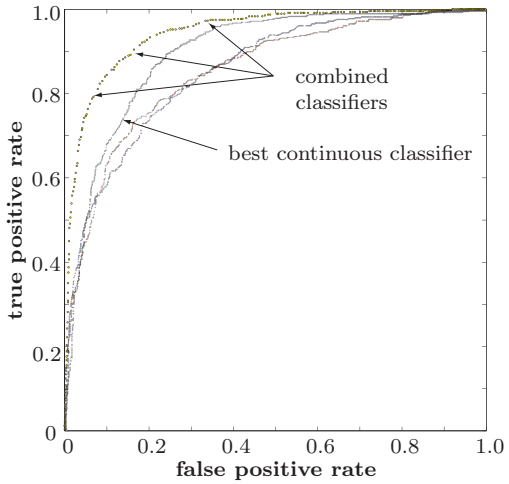


Figure 8.6: 900 candidate classifiers generated using the efficiency-cautious approach. The discrete base classifiers are also shown for comparison.

is, discrete classifiers associated with the same value of the threshold parameter are combined.

The performance of the $\Omega_{\text{ec}} = X = 900$ combined classifiers (evaluated on the optimisation set) are shown in Figure 8.6. Note that there are significantly fewer combined classifiers compared to the performance cautious method (see Figure 8.4).

8.4 Ensemble selection

In the previous section, we introduced two ensemble generation methods. We now discuss classifier *selection*, which is the process of selecting a combined classifier which is optimal for a certain operating criterion. Since we consider only a single fusion strategy, namely majority voting, selecting a combined classifier is equivalent to selecting an ensemble, since each ensemble is associated with only one combined classifier.

8.4.1 Notation

We use the notation $r = r_1, r_2, \ldots, r_{N_S}$ to denote the retinas with which the discrete classifiers in the selected ensemble are associated, where r_1 is the first selected retina in the ensemble, and r_2 the second selected retina, etcetera.

An ensemble of N_S discrete classifiers is denoted by

$$\Psi := \{C^{r_1}\{\rho\}, C^{r_2}\{\rho\}, \ldots, C^{r_{N_S}}\{\rho\}\}, \tag{8.4.1}$$

that is, an ensemble Ψ is constructed by using majority voting to combine the decisions obtained from the discrete classifiers, $C^{r_1}\{\rho\}$, $C^{r_2}\{\rho\}$, $\ldots$, $C^{r_{N_S}})\{\rho\}$.

One can also consider ensemble selection to be *retina* selection, since each discrete classifier in the selected ensemble is associated with a specific retina. Since a continuous classifier is constructed from each retina, a threshold has to be selected as well. An ensemble of size N_S is therefore associated with a set of N_S retinas, as well as a threshold value ρ, which is imposed on each continuous classifier. We can therefore use the equivalent compact notation

$$\Psi := \{r_1, r_2, \ldots, r_{N_S}; \rho\} \tag{8.4.2}$$

to denote an ensemble.

8.4.2 Operational criteria

Since the ensemble generation techniques described in the previous sections produce a pool of candidate ensembles, an appropriate ensemble needs to be *selected*. More specifically, the combined classifier which maximises performance for a certain operational criterion is selected. Although the operational criteria may vary for different practical scenarios, we illustrate classifier selection for three specific operational criteria that a bank manager may prefer to enforce:

(A) a maximum FPR of 0.05 is allowed (that is, the classifier that has the highest TPR, and a FPR of at most 0.05 is selected),

(B) a minimum EER (that is, the classifier which has the smallest EER is selected); and

(C) a minimum TPR of 0.95 is allowed (that is, the classifier which has the smallest FPR, and a TPR of at least 0.95 is selected).

Classifier selection is now illustrated by considering the same sample data used to illustrate ensemble generation in the previous section. The pools of candidate ensembles that were generated to illustrate the performance-cautious and efficiency-cautious approaches (see Figures 8.4 and 8.6) are again shown in Figure 8.7a and b, respectively. The candidate ensembles that are selected for the aforementioned operating criteria, namely a FPR of less than or equal to 0.05 (A), a minimum EER (B), and a TPR of greater than or equal to 0.95 (C) are indicated. The discrete base classifiers which constitute each of these ensembles are also shown. The properties of the selected ensembles are summarised in Table 8.2.

A comparison of the relative performances of the selected ensembles using the performance-cautious and efficiency-cautious approaches are shown in Figure 8.8. While it is clear that the ensembles selected from the performance-cautious pool perform slightly better than those selected from the efficiency-cautious pool, it is worth emphasising that the performances of the selected

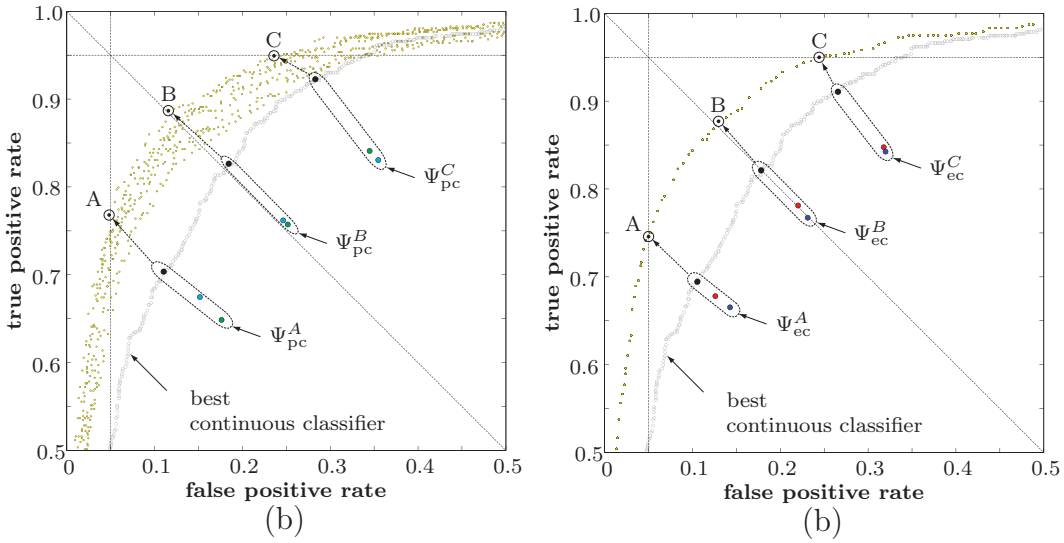


Figure 8.7: Ensemble selection. Combined classifiers generated using (a) the *performance-cautious* approach and (b) the *efficiency-cautious* approach. Three classifiers have been selected (A, B and C) based on three different operating criteria. The ensemble of base classifiers that were combined to form each of the combined classifiers (using majority voting) are also shown.

Operating point	Performance-cautious ensemble	Efficiency-cautious ensemble
FPR ≤ 0.05	$\Psi^A_{\text{pc}} := \{2, 4, 5, 38.6°\}$	$\Psi^A_{\text{ec}} := \{1, 3, 4; 37.8°\}$
EER	$\Psi^B_{\text{pc}} := \{2, 4, 5, 45.4°\}$	$\Psi^B_{\text{ec}} := \{1, 3, 4; 45.0°\}$
TPR ≥ 0.95	$\Psi^C_{\text{pc}} := \{2, 4, 5, 52.0°\}$	$\Psi^C_{\text{ec}} := \{1, 3, 4; 51.1°\}$

Table 8.2: The selected ensembles for three different operating criteria using two different ensemble generation techniques. See Figure 8.7.

classifiers reported here do not depict realistic performances in real-life scenarios, but rather performances achieved using the optimisation set. The reader is also reminded that the data used in this section is a scaled-down version of the actual data, and is used here only to demonstrate the ensemble generation and selection techniques employed in this dissertation. Actual results obtained on the evaluation set are detailed in Chapter 10.

8.4.3 Maximum attainable ROC curves

Before concluding this chapter, we briefly introduce the concept of a maximum attainable ROC (MAROC) curve. Since a conventional ROC curve depicts the trade-off between the FPR and TPR achievable by a single *continuous* classifier in ROC space (by imposing a different decision thresholds), such a ROC curve

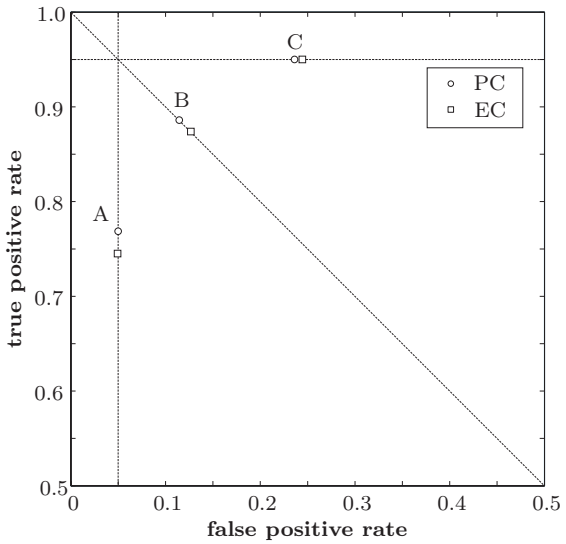


Figure 8.8: A comparison of the performances obtained on the *optimisation* set, for each of the three criteria, using the performance-cautious (PC) and the efficiency-cautious approach (EC). The performance-cautious approach results in better performance for each criterion.

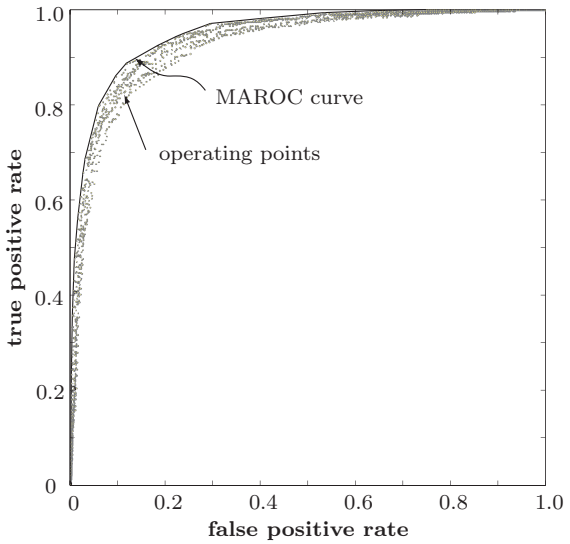


Figure 8.9: MAROC curve. The MAROC curve for all operating points is shown.

cannot be *generated* after classifier combination and selection strategies have been employed. However, the convex hull of the operating points (in ROC space) depicting the respective performances of a set of candidate classifiers can be used to depict the optimal attainable (selectable) performance associated with this set, for several operational criteria. We refer to this convex hull as the so-called MAROC curve.

8.4.4 Conclusion

In this chapter, we provided a general overview of classifier combination strategies. Performance-cautious ensemble generation, as well efficiency-cautious ensemble generation were introduced. We also discussed how a suitable ensemble is selected based on a specific operating criterion. In the next chapter we discuss the data set and experimental protocol used to evaluate the performance of the system developed in this dissertation. The results obtained for these experiments are provided in Chapter 10.

Chapter 9

Data and Experimental Protocol

9.1 Introduction

The implementation of the proposed system, as well the associated computational requirements are discussed in Section 9.2. In Section 9.3 we present a general discussion on data partitioning protocols found in the literature, before introducing the hybrid partitioning protocol utilised in this dissertation. The data set used in this dissertation ("Dolfing's data set") is discussed in Section 9.4. In Section 9.5 we discuss the issue of performance evaluation in multi-iteration experiments. Finally, in Section 9.6 we define the system parameters employed for each experiment.

9.2 Implementation issues

9.2.1 Implementation

The proposed system was implemented in MATLAB. No external libraries, other than an implementation of the DRT, were utilised. The external implementation of the DRT forms part of the *image processing toolbox* developed by MathWorks.

9.2.2 Computational requirements

An in-depth analysis of the computational requirements of the system developed in this dissertation is beyond the scope of this study. We do however, provide a brief overview here. When considering the computational requirements of a signature verification system, it is sensible to consider said requirements for (1) optimising the system parameters, (2) enrolling a new user/client, and (3) classifying a questioned signature, separately. The computational requirements of stage (1) are not critical, since the system parameters need to be optimised only once, and not in real-time. The computational requirements

Figure 9.1: Notation used to denote a data set containing N_w writers. Each block indicates the signatures associated with a specific writer w.

of stage (2) are more important than those of stage (1) but again, stage (2) does not need to occur in real-time. The computational requirements of stage (3) are the most critical, since each questioned signature needs to be classified in real-time, in order for the proposed system to be feasible in practice. We therefore focus our attention on the computational requirements of classifying a signature (stage (3)).

In Coetzer (2005), it is shown that calculating the DRT and matching the resultant observation sequence with the appropriately trained HMM, uses $7.69e7$ floating point operations. Since our system needs to perform this operation $N_S = 11$ times (once for each retina in the optimal selected ensemble), the total number of floating point operations required is $8.46e8$. Since the dimension of the local features is less than half of the dimension of the global features (making both the DRT-based feature extraction and the subsequent matching less complex), $8.46e8$ floating point operations is a conservative estimate.

9.3 Data partitioning

In Chapter 1, we described how a data set is partitioned into an optimisation set $\boldsymbol{O}$ and an evaluation set $\boldsymbol{E}$, after which it is further subdivided into training sets ($\boldsymbol{T}_O^+$, $\boldsymbol{T}_E^+$) and testing sets ($\boldsymbol{E}^+$, $\boldsymbol{E}^-$, $\boldsymbol{O}^+$ and $\boldsymbol{O}^-$) (see Figures 1.4 and 1.5). In this section, we explain in more detail how this partitioning is achieved, that is, how to decide which writers are assigned to set $\boldsymbol{O}$ and which writers are assigned to set $\boldsymbol{E}$. We use the notation in Figure 9.1 to represent a data set with N_w writers. Each block indicates a different writer (and his/her associated signatures). Note that the labels $w = 1$ (writer 1), $w = 2$ (writer 2), etc, are *arbitrary* labels assigned to the respective writers in the data set.

In an ideal scenario, sufficient signature data will be available so that a large optimisation set $\boldsymbol{O}$, as well as a large evaluation set $\boldsymbol{E}$ can be utilised. Both of these sets can then be considered representative of the general population. In this scenario the subsequent performance when evaluating data set $\boldsymbol{E}$ can be trusted as indicative of a potential real-world performance.

Unfortunately, no available signature data set is adequately large to reach the above-mentioned conclusion. We therefore use data partitioning protocols that allow the reuse of data, generally by running multiple iterations of the

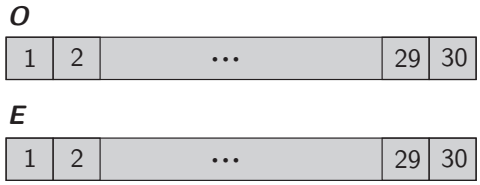


Figure 9.2: Resubtitution method for $N_w = 30$. The entire data set (that is, all the writers) is assigned to both the optimisation set and evaluation set. The size of each set is maximised, but overfitting inevitably occurs.

same experiment using different subsets of the data, in order to get a reliable *estimate* of the potential real-world performance. A further advantage of running multiple iterations, is that it also gives an indication of variance. The number of iterations performed for each experiment is denoted by L. In Sections 9.3.1 - 9.3.4, we present a brief overview of several popular data partitioning protocols encountered in the literature. These protocols are summarised in Kuncheva (2004). In Section 9.3.5, we introduce the hybrid protocol used in this dissertation, which aims to address certain weaknesses found in existing protocols.

9.3.1 Resubstitution (R-method)

The resubstitution method maximises the size of the optimisation and evaluation sets by assigning the entire data set (that is, all the writers) to both of these sets. The resubtitution method therefore only allows $L = 1$ iteration. Since the system parameters are essentially optimised using the *evaluation set*, overfitting inevitability occurs, and the reported results (using this protocol) should always be considered optimistically biased. The resubtitution method is therefore an undesirable data partitioning protocol. The resubtitution method is illustrated in Figure 9.2, with $N_w = 30$.

9.3.2 Hold-out (H-method)

The hold-out method splits the data set into two halves (other proportions may be used). Traditionally, one half (that is, half the writers) is assigned to the optimisation set, while the other half is assigned to the evaluation set. Optionally, the experiment can be repeated with the evaluation and optimisation sets swapped (allowing for $L = 2$ iterations), and the average of the two results is reported. The hold-out method successfully keeps the evaluation and optimisation sets separate, however, since only two iterations are possible, only a mean result can be reported. The hold-out method is illustrated in Figure 9.3, with $N_w = 30$.

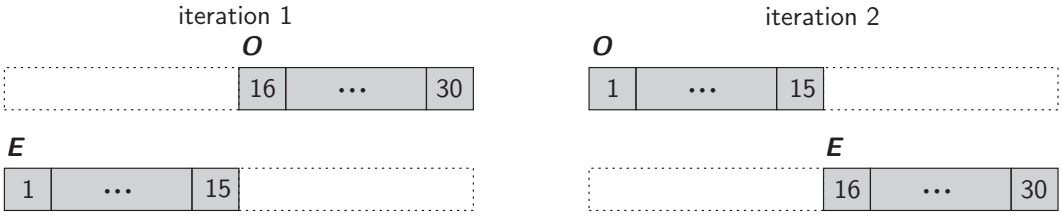


Figure 9.3: Hold-out method for $N_w = 30$. The data set is split into two halves, where one half is used as the optimisation set, while the other half is used as the evaluation set (iteration 1). The experiment can be repeated (iteration 2) with the two halves swapped.

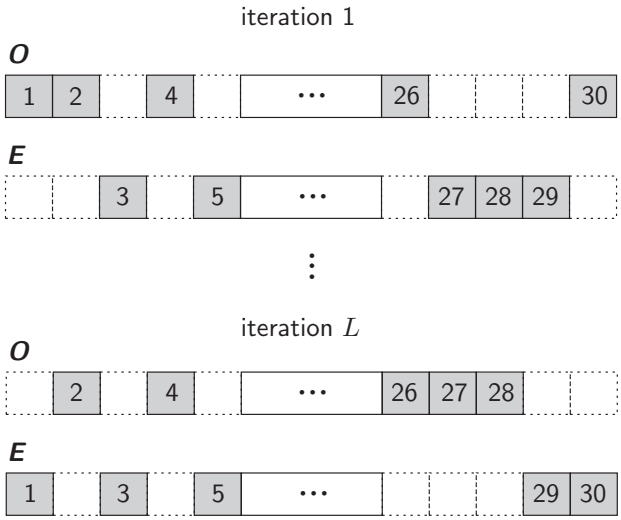


Figure 9.4: The data shuffling method for $N_w = 30$. Writers are randomly assigned to either the evaluation or optimisation set, according to a fixed proportion - in this example, *half* the writers are assigned to the evaluation set. The experiment is then repeated L times, by randomly reassigning the writers.

9.3.3 Data shuffling

The so-called *data shuffling* method randomly assigns each writer to either the optimisation set or the evaluation set, in a fixed, predefined proportion. The data is then *shuffled* (that is, reassigned at random) L times, to produce L experimental iterations. The data shuffling method, with $N_w = 30$, is illustrated in Figure 9.4.

9.3.4 k-fold cross-validation (π-method)

In k-fold cross-validation, the data is partitioned into k sections of equal size (k should ideally be a factor of N_w). Each of the k sets, in turn, is used as the evaluation set, while the union of the remaining sets is used as the optimisation set, to produce k iterations. Conventionally, each iteration L is

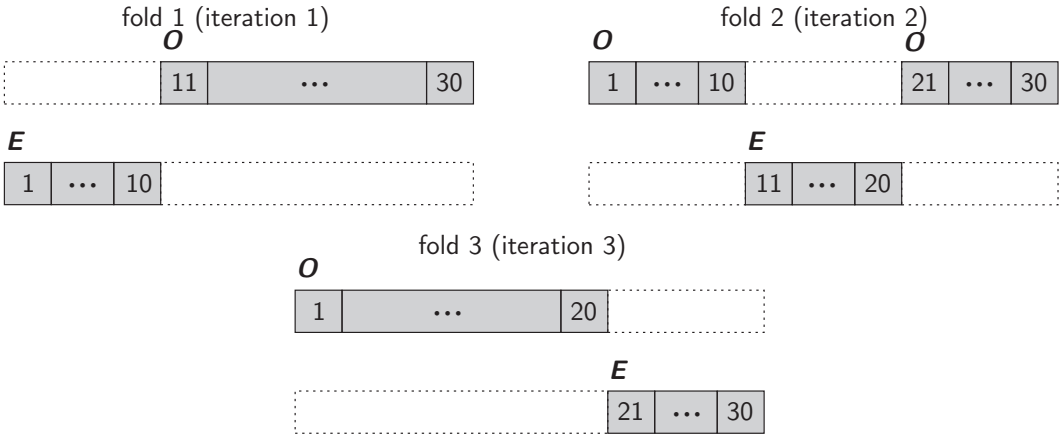


Figure 9.5: k-fold cross validation for $N_w = 30$ and $k = 3$. The data set is split into three sections. Each section (ten writers) is, in turn, used as the evaluation set, while the union of the remaining sets (twenty writers) is used as the optimisation set.

referred to as a *fold*, and the number of iterations L is equal to the number of folds (conventionally denoted by k). Note that when $k = 2$, k-fold cross-validation is equivalent to the hold-out method (with two iterations). If $k = N_w$, the method is referred to as the *leave-one-out* method, since each writer is individually, in turn, used as the evaluation "set", while the remaining writers are used as the optimisation set. Figure 9.5 illustrates 3-fold cross-validation ($k = 3$) on a data set containing $N_w = 30$ writers.

9.3.5 k-fold cross-validation with shuffling

In this section we introduce the hybrid data partitioning protocol utilised in this dissertation, which aims to combine two separate protocols, namely k-fold cross-validation and data shuffling, in order to address their respective weaknesses.

In theory, the data shuffling method, allows for a maximum of $\binom{N_w}{h}$ possible iterations (where h denotes the proportion of the split). In practice, however, only a few iterations are typically performed, due to the limitations of the available computational resources. If both the size of the data set N_w, and the number of iterations L, are small (which is generally the case in practice), the influence of "outliers" (that is, writers that perform atypically well or poorly) is significant. For example, if several writers that perform very well (that is, writers with weak forgeries and relatively invariant genuine signatures) are, by chance, assigned to the evaluation set during most of the iterations, the results will consequently be optimistically biased[1]. By increasing the number

[1]Experiments on Dolfing's data set, which is used in this dissertation, have shown that this effect is significant.

of iterations, L, the reliability of the results can be improved. However, this also greatly increases the computational requirements.

By employing k-fold cross-validation (as discussed in Section 9.3.4), we are assured that each writer is used exactly once as an evaluation writer, therefore minimising the influence of outliers. However, we are limited to $L = k$ iterations using the standard k-fold cross-validation protocol. Furthermore, by simply reassigning writer labels ($w = 1, 2, \ldots, N_w$) and repeating the experiment (using the same k-fold cross validation protocol), different results can be obtained[2]. We therefore use a hybrid protocol, namely *k-fold cross-validation with shuffling*, in order to get reliable results. The proposed protocol is as follows:

1. split the data into k folds and perform k-fold cross-validation;
2. randomly reassign labels ($w = 1, 2, \ldots, N_w$) to all the writers in the data set;
3. repeat steps 1 and 2, R times, so that a total of $L = R \cdot k$ iterations (experiments) are performed.

The above protocol ensures that each writer is used as an evaluation writer exactly R times.

9.4 Dolfing's data set

We evaluate the system proposed in this dissertation using Dolfing's data set. This data set was originally captured on-line for Hans Dolfing's Ph.D. thesis (Dolfing (1998)). The signatures were converted to static (that is, off-line) signature images by Johannes Coetzer for his Ph.D. thesis (Coetzer (2005)). The pen-position data (that is, the pen-tip position coordinates on the tablet) is used to render each signature image. This is achieved through morphological dilation (see Gonzalez and Woods (2002)) of the pixels corresponding to the pen-tip positions in the on-line data, using a suitable structuring element. The signature images are therefore "ideal" in the sense that they contain no background noise and exhibit no variation in pen-stroke width. Each signature has a uniform stroke width of approximately five pixels.

Dolfing's data set contains 4530 unique signatures across $N_w = 51$ different writers. For each writer, thirty positive signatures and sixty negative signatures (amateur-skilled forgeries) are provided, with the exception of two writers, for which only thirty negative signatures are provided. For the sake of uniformity, the negative signatures for said two writers are duplicated, so that sixty negative signatures are used for *each* writer. As a result, the data

[2]Again, experiments on Dolfing's data set have shown that this effect is significant.

		Signatures per writer	Signatures per partition
Optimisation set ($\boldsymbol{O}$)	$\boldsymbol{T}_O^+$	15	510
(34 writers)	$\boldsymbol{O}^+$	15	510
	$\boldsymbol{O}^-$	60	2040
Evaluation set ($\boldsymbol{E}$)	$\boldsymbol{T}_E^+$	15	255
(17 writers)	$\boldsymbol{E}^+$	15	255
	$\boldsymbol{E}^-$	60	1020

Table 9.1: Partitioning of Dolfing's data set. The number of signatures in each partition is shown.

set contains 4590 signatures. In addition to amateur-skilled forgeries, Dolfing's data set also contains *professional* forgeries, but said signatures are not considered in this dissertation, due to the fact that a very limited number of these signatures are available.

Since the data set contains $N_w = 51$ writers, we conveniently employ 3-fold cross-validation ($k = 3$), so that for each fold $\frac{51}{3} = 17$ writers are assigned to the evaluation set, while 34 writers are assigned to the optimisation set. We choose $R = 10$ (that is, ten repetitions), so that $L = 30$ iterations per experiment are performed in total.

The number of signatures assigned to each partition (during a single experimental iteration) is summarised in Table 9.1.

9.5 Performance evaluation in multi-iteration experiments

In Section 6.3, we introduced several performance evaluation measures, namely the FPR, the TPR and the AUC. The FPR, TPR, TNR and FNR are suitable for describing the performance of a single *discrete* classifier, whereas ROC curves are suitable for visualising the performance of a single *continuous* classifier. However, the performance evaluation protocol adopted in this dissertation, as well as the protocols generally employed by researchers, utilise *multiple* experimental iterations to evaluate the performance of a *single* classifier. We therefore need to address the issue of performance evaluation in multi-iteration experiments.

Note that, in the following discussion, we make the assumption that the performance associated with each experimental iteration can be depicted by a ROC curve. This discussion is therefore only relevant to performance evaluation scenarios in which generative classifiers are used. If a *single* generative classifier is evaluated, said ROC curve is generated directly by employing a sliding threshold (see Chapter 6). However, when classifier combination and

selection strategies are employed, as is the case in this dissertation, the ROC curve obtained from each experimental iteration is in fact a MAROC curve, as explained in Section 8.4.3.

In the next section we discuss so-called "traditional" averaging methods, that are generally employed by researchers. We explain in the next section that these traditional averaging methods can be optimistically biased. We then introduce the concept of operating point stability, before discussing operating point-based averaging, which aims to address the issue of optimistic bias inherent in traditional averaging methods.

9.5.1 Traditional averaging methods

Researchers have adopted various approaches to performance evaluation in multi-iteration experiments, but said approaches generally involve reporting the average performance achieved for a certain operating criterion across all ROC curves. For example, if it is desirable to report the TPR achieved for a certain FPR (that is, an operating criterion that is based on a maximum allowable FPR), the *average* TPR (for said imposed FPR) obtained across L ROC curves, generated for L experimental iterations, is reported. This scenario is illustrated in Figure 9.6, in which $L = 5$ ROC curves are depicted. The five points which are averaged to report the performance at a FPR of 0.05 are indicated by $\circ$'s. Similar approaches have been adopted for reporting the performance based on other operating criteria. For example, the operating points used to obtain an average EER, are depicted by $\square$'s in Figure 9.6. If it is desirable to depict the average performance of a classifier based on a *single* ROC curve, generated across L experimental iterations, the L individual ROC curves may be averaged, using (for example) *vertical* averaging (see Fawcett (2006)), to generate an "average" ROC curve.

While the above "traditional" averaging approaches may seem reasonable, the reported results do not constitute a reliable performance measure. We now substantiate this assertion.

Consider a scenario in which L ROC curves are generated by employing the *leave-one-out* data partitioning protocol (see Section 9.3.4). Each ROC curve (or MAROC curve, for the case where classifier selection is employed) will therefore depict the performance of a single writer only (the signatures of $N_w - 1$ writers are used for optimisation and the signatures of only one writer are used for evaluation). The task of generating an "average" ROC curve therefore reduces to a score normalisation problem (see Chapter 7). The constraints (that is, knowledge of expected score distributions for each writer) that apply to score normalisation must therefore also be enforced when generating an average ROC curve.

Said constraints are not adhered to when employing the above averaging methods. Vertical averaging, for example, is equivalent to applying Ross's method (see Section 7.3.3) for score normalisation, where *perfectly* estimated

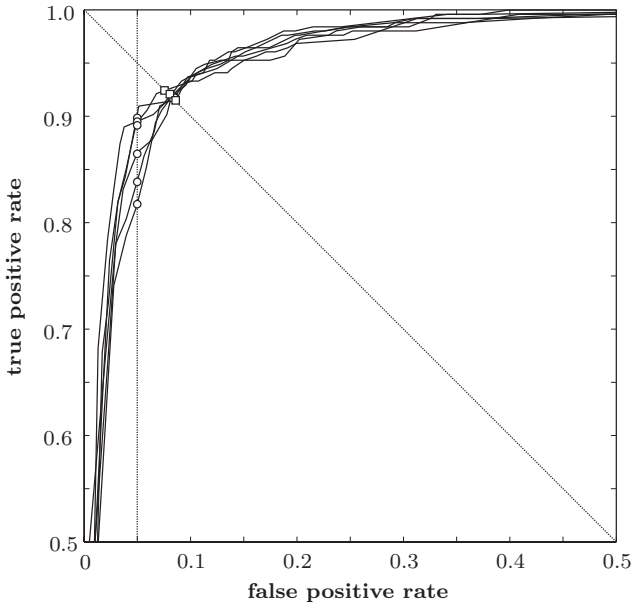


Figure 9.6: Traditional averaging approach. Five ROC curves are shown, each depicting the performance achieved for a specific experimental iteration. The ∘'s indicate the points that are averaged to report the TPR achieved for an FPR of 0.05. The □'s indicate the points that are averaged to report an EER.

negative score distributions are assumed, for *each* writer. The obtained ROC curve may be optimistically biased, since it is equivalent to the "ideal" ROC curve that can only be obtained when infinitely large optimisation and evaluation sets are available, as discussed in Section 7.3.3. Similar arguments can be used to show the inadequacy of other averaging methods. For example, reporting an EER for several experimental iterations, by averaging the EER for the iteration-specific ROC curves (generated by using the leave-one-out method) is equivalent to the R-score normalisation strategy introduced in Section 7.3.4, again assuming *perfectly* estimated positive and negative score distributions for each writer, which constitutes the "ideal" case.

The optimistic bias that arises when employing traditional averaging is most pronounced when utilising the *leave-one-out* method, although it does diminish as the number of writers in each evaluation set increases. Traditional averaging approaches to performance evaluation, although common in the literature, are directly influenced by the size of the evaluation set, with *significant* performance "improvements" made by simply decreasing the size of the evaluation set used in each experimental iteration (by, for example, increasing the size of k in k-fold cross validation). This effect is demonstrated in Section 9.5.4.

In addition to the optimistic bias inherent in traditional averaging techniques, said techniques are further disadvantaged by their inability to provide an indication of *operating point stability*. The concept of operating point sta-

bility is introduced in the next section.

9.5.2 Operating point stability

We define operating point stability (OPS) as a measure of how reliably an *imposed* operational criterion ξ and associated combined classifier's performance ζ achieved on an optimisation set can be reproduced on an evaluation set, across several experimental iterations. We define two orthogonal axes in ROC space. The first axis, referred to as the operational axis, constitutes the axis on which an operational criterion is imposed. The second axis, referred to as the performance axis, is orthogonal to the operational axis.

The reliability of an operational criterion is quantified by its stability $\frac{1}{\sigma_\xi^2}$ (referred to as *operational stability* (OS)[3]), and predictability $\mu_\xi - \xi$ (referred to as *operational predictability* (OP)[4]) , where σ_ξ and μ_ξ denote the standard deviation and mean, respectively, of the operational error rate achieved on an evaluation set across L experimental iterations. These quantities are depicted in Figure 9.7a. The vertical line indicated by ξ represents an operational criterion, for example, an imposed FPR of 0.1, or an imposed TPR of 0.8. The PDF represents the distribution of error rates achieved on the L evaluation sets, along the operational axis, and is defined by the values μ_ξ and σ_ξ. The absolute value of the OP is defined as the distance between the imposed criterion (error rate) ξ, and the average error rate (along the operational axis) μ_ξ calculated across L iterations. Note that, an operational criterion is usually imposed as a constraint, for example "a FPR of no greater than 0.1" or "a TPR of at least 0.95". A negative OP is therefore desirable in the cases where a "maximum" constraint is imposed, whereas a positive OP is desirable when a "minimum" constraint is imposed.

Analogues of the above measures can be defined in a similar way for the performance axis, as shown in Figure 9.7b. Note that the line indicated by ζ represents the average performance obtained (for the imposed criterion ξ) across the L optimisation sets, while the measures μ_ζ and $\frac{1}{\sigma_\zeta^2}$ denote the *mean performance* and *performance stability* (PS), respectively, where σ_ζ denotes the standard deviation of the performance achieved on the evaluation set, across L experimental iterations. The "performance predictability" (PP) $(\mu_\zeta - \zeta)$ is therefore equivalent to the generalisation error. A negative PP is therefore typically reported when the performance axis coincides with a the TPR, whereas a positive PP is typically reported when the performance axis coincides with

[3] This measure is often termed "precision" in statistics, but we refrain from using this terminology in order to avoid confusion with the measure $\frac{\text{TP}}{\text{TP+FP}}$, which is also termed "precision" in the machine leaning literature.

[4] The absolute value of this measure is often termed "accuracy" in statistics, but we refrain from using this terminology in order to avoid confusion with the measure $\frac{\text{TP+FP}}{\text{TP+FP+FN+TN}}$, which is also termed "accuracy" in the machine learning literature.

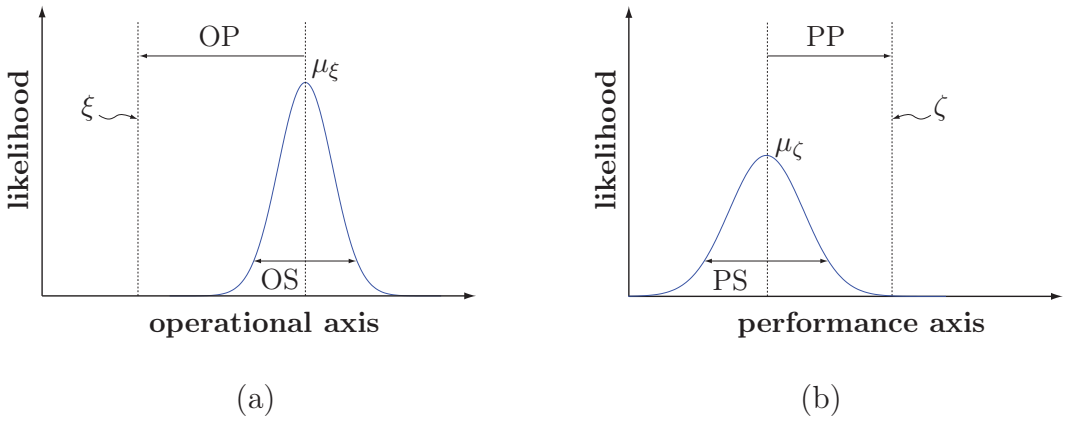


Figure 9.7: Operating point stability. (OP = operational predictability, OS = operational stability, PP = performance predictability, PS = performance stability)

the FPR or EER. That is, the performance of the selected combined classifier is typically better (lower in the case of the FPR or EER and higher in the case of the TPR) on the optimisation set than on the corresponding evaluation set. A large generalisation error (indicated by a large PP) indicates that significant overfitting has occurred. A PP of close to zero is therefore generally preferred.

The interpretation of the above parameters is now clarified with an example. Consider a scenario in which a human operator imposes a FPR-based operating constraint, for example, the classification system must operate with a maximum FPR of 0.1 (that is, no more than 10% of fraudulent signatures should be accepted). The procedure for estimating the system's performance and operational stability is as follows.

For each experimental iteration−in this dissertation, 3-fold cross-validation with 10 repetitions is employed, so that $L = 30$ iterations are performed−the optimisation set is used to select the optimal combined classifier (and associated ensemble) with the highest TPR (and a FPR of less than 0.1) as discussed in Section 8.4.2. Said ensemble is then used to classify all of the signatures in the corresponding evaluation set, by employing majority vote fusion. The performance of said classifier is subsequently depicted by a single point in ROC space. The above process is repeated L times, each time utilising the signatures of different writers in the optimisation and evaluation sets, respectively, so that L operating points are generated (see Figure 9.8a).

Since, in this example, a constraint based upon the FPR is imposed, the FPR-axis constitutes the operational axis, while the TPR-axis constitutes the performance axis. The cluster of operating points is then modelled using a binormal (bivariate Gaussian) distribution, with the respective axes coinciding with the operational and performance axes. The statistics μ_ζ, μ_ξ, σ_ζ and σ_ξ are illustrated in Figure 9.8b. For the scenario depicted in Figure 9.8, the measures OS, OP, PS and PP (which are based on the statistics μ_ζ μ_ξ, σ_ζ and σ_ξ), as well as the individual statistics, μ_ζ and μ_ξ, are tabulated in Table 9.2

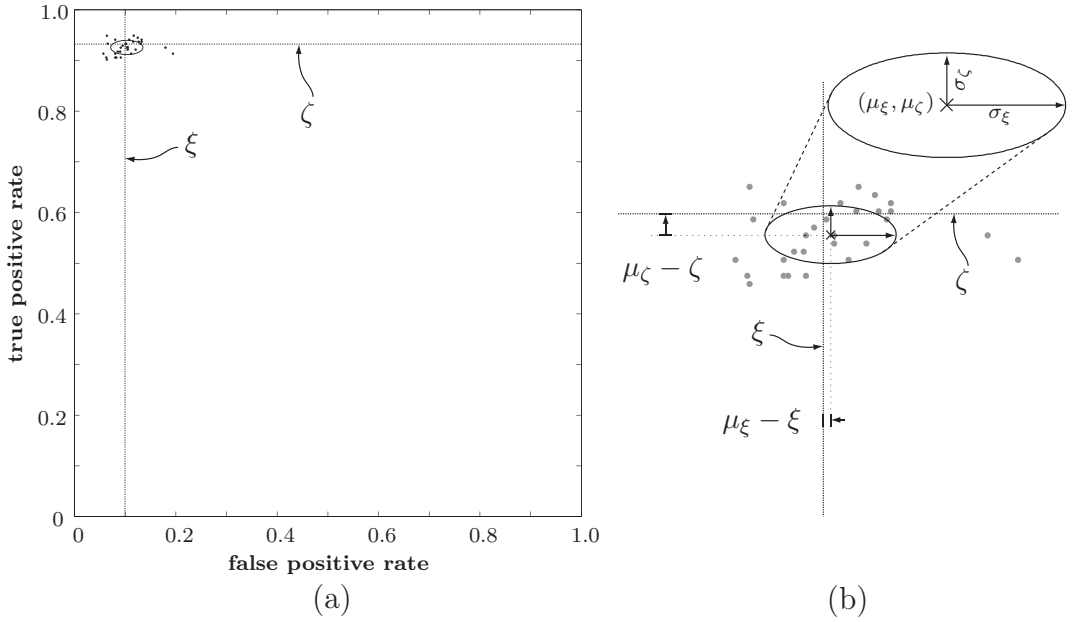


Figure 9.8: (a) A cluster of $L = 30$ operating points produced when an operating constraint of a FPR < 0.1 is imposed. The ellipse visually represents the distribution of said operating points. (b) A closer look at the $L = 30$ operating points in (a), with the performance evaluation measures indicated. The cluster of points is modelled using a binormal (bivariate Gaussian) distribution, with the respective axes coinciding with the operational and performance axes. The distribution ellipse is centred on the mean (μ_ξ, μ_ζ), and has dimensions equivalent to two standard deviations in each direction.

Operational axis (FPR)			Performance axis (TPR)		
ξ	μ_ξ(OS)	OP	ζ	μ_ζ(PS)	PP
FPR < 0.1	0.103(988)	+0.003	0.935	0.926(5102)	−0.009

Table 9.2: Performance evaluation measures for the experiment depicted in Figure 9.8. In this scenario, a maximum FPR of 0.1 has been imposed.

and are interpreted as follows.

In this example, the OP is positive but small, indicating that on average, a FPR of 0.103, which is slightly greater than the maximum imposed criterion of 0.1, will be obtained. The OS of 988 is relatively low, indicating that the FPRs achieved for individual writers will vary significantly, with some writers experiencing significantly higher FPRs than the imposed maximum of 0.1. This is further clarified by examining the distribution of operating points in Figure 9.8[5]. The mean performance μ_ζ is 0.926, which indicates that, on average, a TPR of 0.926 will be achieved across all writers in the

[5]Each operating point depicts the *average* performance of all the writers in the evaluation set. Since several operating points have FPRs of greater than 0.1, it necessarily follows that individuals writers experience FPRs of greater than 0.1.

evaluation set, when the aforementioned constraint is imposed. The relatively high PS indicates that the respective TPRs achieved for the individual writers do not vary significantly (see Figure 9.8). Finally, the small negative PP (or generalisation error) is indicative of slight overfitting.

In this section, we have shown how the performance of a classifier can be evaluated (when a specific operating criterion is imposed) by considering distributions along the operational and performance axes. In the following section, we show how this procedure can be extended to generate an average ROC *curve* which provides a *visual* indication of a classifier's performance for an *entire range* of operating criteria.

9.5.3 Operating point-based averaging

The procedure for generating an average ROC curve using operating point-based averaging is as follows. The FPR (or TPR) is swept from 0 to 1, at arbitrarily small, uniform intervals. For each operational criterion, an average operating point ((μ_ξ, μ_ζ) for a FPR-based criterion, or (μ_ζ, μ_ξ) for a TPR-based criterion) is calculated (as described in the previous section). Said operating point then constitutes a point on the average ROC curve. An average ROC curve, which is constructed by sweeping a FPR-based criterion from 0 to 1, in increments of 0.1, is shown in Figure 9.9. Distribution ellipsoids are also shown.

Operating point-based averaging is closely related to *threshold* averaging (see Fawcett (2006) and Macskassy and Provost (2004)), where an average ROC curve is obtained by averaging those points on the individual ROC curves that are associated with the same threshold parameter value (as discussed in Chapter 7). While threshold averaging is well-suited to averaging ROC curves that are generated from continuous classifiers, operating point-based averaging, on the other hand, constitutes a similar process that is applicable to scenarios where classifier combination is employed.

9.5.4 Traditional averaging versus operating point-based averaging

We now demonstrate the optimistic bias inherent to traditional averaging approaches when compared to the performances reported when using operating point-based averaging. We also illustrate that the performances reported using traditional averaging methods are dependent on the size of the evaluation set used during each iteration (as discussed in Section 9.5.1), and that this is not the case when operating point-based averaging methods are used.

The experimental protocol used to generate the operating points depicted in Figure 9.10 is as follows. Three experiments are conducted using k-fold cross-validation, with $k = 3$, $k = 17$ and $k = 51$ (the leave-one-out method)

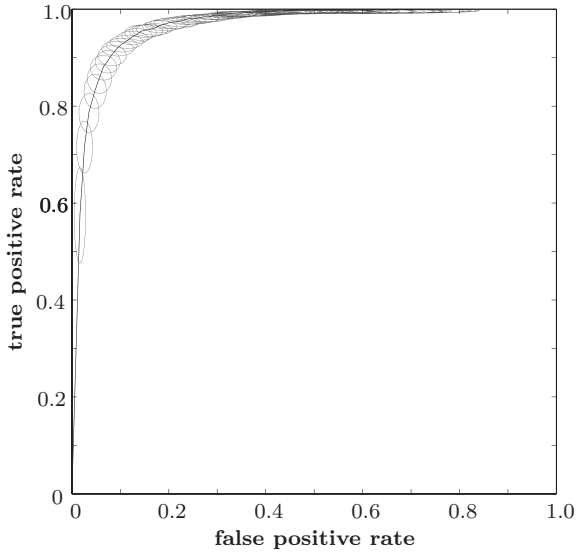


Figure 9.9: An average ROC curve obtained from L experimental iterations using operating point-based averaging. Distribution ellipsoids are also shown.

respectively. Except for the value of k, the experimental protocol, data set and system design are identical for each experiment. We consider the performance at the EER for each case. When traditional averaging is used, we generate L MAROC curves for each case. The EERs achieved on said MAROC curves (for each case) are averaged (traditional averaging) in order to obtain an average EER (depicted by $\square$'s in Figure 9.10a and b).

The experiments are repeated (for $k = 3$, $k = 17$ and $k = 51$) using operating point-based averaging−instead of generating L MAROC curves, an EER-based selection criterion is used to generate L operating points (each with approximately an EER). For each case, the L operating points are averaged to produce an average performance (depicted by $\circ$'s in Figure 9.10a and b). Note that said operating points have a non-zero OP.

By examining Figure 9.10b, it is clear that the EERs (depicted by $\square$'s) obtained when using traditional averaging approaches are lower (better) than when using operating point-based averaging. Furthermore, the EER obtained when using traditional averaging is correlated with the size of the evaluation set. When $k = 51$, each of the L evaluation "sets" contains the signatures of only one writer, and an EER of approximately 10.8% is reported. When $k = 3$, each evaluation set contains the signatures of seventeen writers, resulting in an EER of approximately 12.5%−constituting a signifcant performance descripency. When we consider the EERs (depicted by $\circ$'s) obtained when operating point-based averaging is utilised, it is clear that the EER performance remains relatively constant when k changes. Furthermore, by examining the actual EERs (of approximately 13.0%) the optimistic bias inherent to traditional averaging methods becomes evident.

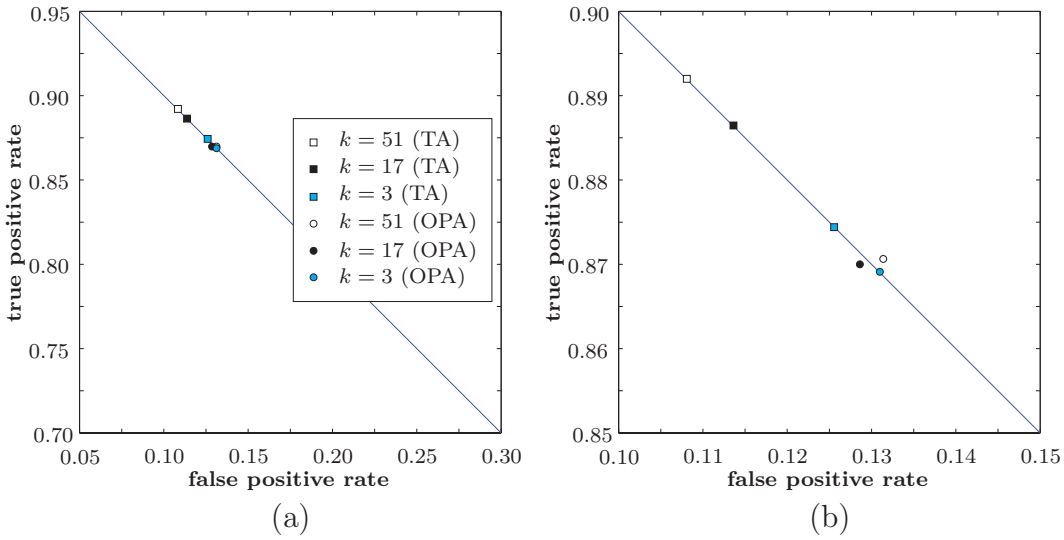


Figure 9.10: (a) Traditional averaging (TA) versus operating point-based averaging (OPA) for $k = 3$, $k = 17$ and $k = 51$. (b) A closer look at the EER for each curve in (a). The legend in (a) also applies to (b).

9.6 Employed system parameters

In this section, we discuss the specific values assigned to each of the system parameters used in this dissertation. For each parameter, the relevant section in which said parameter was originally introduced is provided for reference.

9.6.1 Feature extraction

We use $Z_h = \{0, 99\}$ horizontal intervals, and $Z_v = \{0, 50, 99\}$ vertical intervals, in order to define $N_r = 16$ (15 local and 1 global) retina centroids (see Sections 3.3.3 and 3.3.4). The fifteen local retinas are constructed with radii of $\gamma = 120$ pixels[6]. An example of a signature image with retinas constructed using the above values is shown in Figure 9.11a. The retina numbering scheme utilised in this dissertation is shown in Figure 9.11b. Although the layout of the retina centroids will change when a different signature is considered (since a flexible zoning scheme is employed), the numbering scheme is universal. For example, "retina 1" will always be the top-left retina and "retina 8" will always be the retina centred on the gravity centre of a signature, etc.

We use $N_\theta = 128$ angles, and $d = 2\gamma = 240$ beams per angle to calculate the DRT of local retinas, for the purpose of generating an observation sequence (see Sections 3.4 and 3.5), and $d = 2\gamma = 512$ beams per angle for global retinas.

[6]Since the dimensions of the signatures in Dolfing's data set have already been normalised, we can define the retina size using a fixed number of pixels. Should this not have been the case, we could have easily defined the retina size as a fraction of the width/height of each signature image, as explained in Chapter 5.

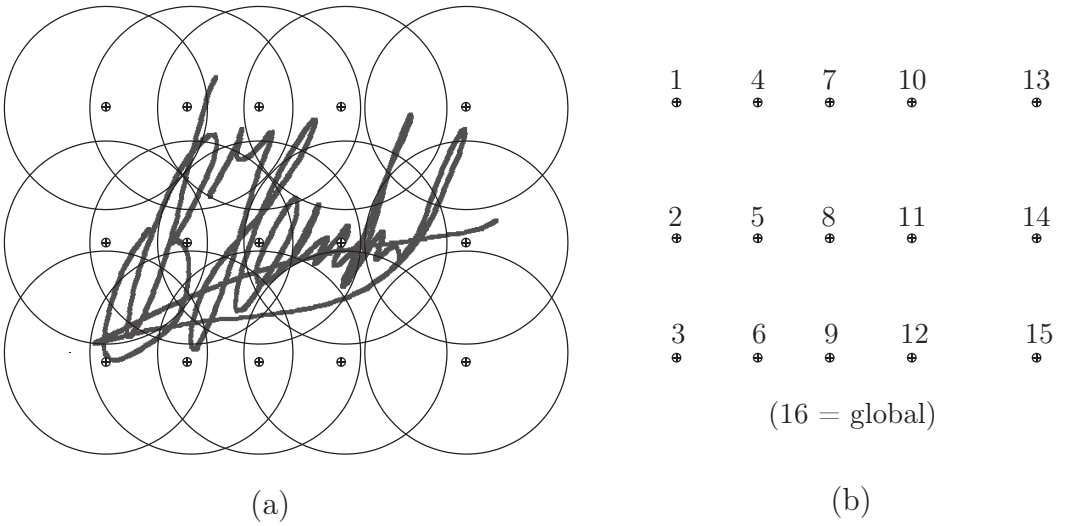


Figure 9.11: Signature zoning and retina construction. (a) An example of a signature image with retinas constructed using the parameters employed in this dissertation. (b) The retina numbering scheme used in this dissertation.

Local observation sequences therefore contain $T = 2N_\theta = 256$ feature vectors, each with a dimension of $d = 240$. The global observation sequence contains $T = 256$ feature vectors, each with a dimension of $d = 512$. Note that the values chosen for N_θ and T are based on the research done in Coetzer (2005), in which said values were found to be optimal.

9.6.2 Signature modelling

Each retina is modelled using a ring-structured HMM with $N = 64$ states. Each HMM has a uniform initial state distribution ($\pi_i = 1/64, i = 1, \ldots, 64$), and is initialised using uniform state transition probabilities: the probability of transitioning to the next state is initialised to 0.8, and the probability of staying in the same state is initialised to 0.2, as explained in Chapter 4.

9.6.3 Classifier selection

Since each signature is modelled using $N_r = 16$ retinas, we select ensembles of sizes $N_S = 1, 3, 5, 7, 9, 11, 13, 15, 16$. We usually consider ensembles containing an odd number of base classifiers, in order to avoid the possibility of a tie occurring when using majority voting. We do, however, include an ensemble of size 16 (that is, an ensemble that contains a discrete classifiers associated with each of the retinas).

9.7 Conclusion

In this chapter we discussed the data set, data partitioning protocol, system parameters and performance evaluation protocol used to conduct suitable experiments in order to investigate the proficiency of the system developed in this dissertation. In the next chapter we present and discuss the results obtained for these experiments.

Chapter 10

Results

10.1 Introduction

In Section 10.2 we present the results for the performance-cautious ensemble generation approach (see Section 8.3.2), for which three different selection criteria, namely (1) "a FPR of at most 0.1", (2) "a TPR of at least 0.9" and (3) the EER, are considered. For each selection criterion, we present separate results for the different ensemble sizes considered. In Section 10.3 we present the results, as described above, when the efficiency-cautious ensemble generation approach (see Section 8.3.3) is utilised. A comparison between the respective results obtained for the performance-cautious and efficiency-cautious approaches follows in Section 10.4.

10.2 Performance-cautious ensemble generation

We present the results for each selection criterion in tabular form, in which the operational mean (μ_ξ), operational stability (OS), operational predictability (OP), mean performance across the evaluation sets (μ_ζ), performance stability (PS) and performance predictability (PP) are shown for each ensemble size N_S. The operational criterion (ξ) and mean performance across the optimisation sets (ζ) are also tabulated. Note that the operational criterion (ξ) is set by a human operator, whereas all the other measures (that is, μ_ξ, OS, OP, μ_ζ and PS) are estimated using the $L = 30$ *evaluation* sets, with the exception of ζ, which indicates the mean performance of the selected ensembles across the *optimisation* sets.

In each table, the ensemble size (and corresponding row) which results in the best mean performance across the optimisation sets (ζ) is shown boldface. The corresponding value of μ_ζ (also in boldface) therefore estimates the best attained performance across the evaluation sets, which are representative of

the general public. Note that, in some cases the ensemble size which results in the best mean performance across the optimisation sets (ζ) may not result in the best mean performance across the evaluation sets (μ_ζ) (underlined). It is worth emphasising that, in the aforementioned scenario, the *evaluation* performance μ_ζ *corresponding to* the best *optimisation* performance ζ should be considered an estimate of the best *attainable* performance. The best *evaluation* performance underlined in each table, is therefore not *attainable* due to the fact that a human operator is unable to make this assessment based only on the optimisation sets.

In addition to each table, a figure illustrating the mean performance across the optimisation sets (ζ), as well as the mean performance across the evaluation sets (μ_ζ) is plotted against the ensemble size N_S.

10.2.1 FPR-based constraint

Table 10.1 shows the results for the system using performance-cautious ensemble generation, when a maximum FPR of 0.1 is imposed. For the majority of the ensemble sizes, the maximum imposed FPR is slightly exceeded, as indicated by positive OP-values in the fourth column.

The mean evaluation performance μ_ζ achieved for each ensemble size is also shown. Note that the best performance (underlined and in boldface) of 0.929 (indicating that, on average, 92.9% of positive signatures are accepted), is obtained for an ensemble of size $N_S = 9$. Also note that, for each ensemble size, the mean performance achieved across the evaluation sets (μ_ζ) is less than or equal to the mean performance achieved across the corresponsding optimisation sets (ζ), which is indicative of slight overfitting.

While the best performance (TPR = 0.929) is obtained when an ensemble of size $N_S = 9$ is utilised, this is not necessarily the optimal result, since the corresponding FPR of 0.108 exceeds the imposed criterion. Although the result obtained for an ensemble of size $N_S = 13$ may be considered more desirable−the TPR is *slightly* lower, while the FPR is *much* closer to the imposed criterion−it is worth emphasising that a human operator is unable to make this judgement, since he/she is only able to select the optimal ensemble based on the objective function ζ. Since the ensemble of size $N_S = 9$ has the highest average performance across the *optimisation* set, the results obtained for this ensemble size must be considered the best obtainable result.

In Figure 10.1, the mean evaluation performance (μ_ζ), as well as the average performance obtained across the optimisation sets (ζ) are plotted against the ensemble size N_S. The vertical bars indicate the PS, and are constructed in such a way that they extend by one standard deviation (σ_ζ) from the mean value in both directions.

	Operational axis (FPR)			Performance axis (TPR)		
N_S	ξ	μ_ξ(OS)	OP	ζ	μ_ζ(PS)	PP
1	FPR < 0.1	0.099(1024)	−0.010	0.821	0.821(1059)	−0.000
3	FPR < 0.1	0.115(982)	+0.015	0.888	0.880(2537)	−0.008
5	FPR < 0.1	0.118(1243)	+0.018	0.921	0.906(3592)	−0.015
7	FPR < 0.1	0.111(1510)	+0.011	0.934	0.919(4203)	−0.015
9	**FPR < 0.1**	**0.108(997)**	**+0.080**	**0.941**	**0.929(3592)**	**−0.012**
11	FPR < 0.1	0.107(874)	+0.007	0.940	0.926(4239)	−0.014
13	FPR < 0.1	0.103(986)	+0.003	0.935	0.926(5087)	−0.009
15	FPR < 0.1	0.100(950)	+0.000	0.922	0.919(3062)	−0.003
16	FPR < 0.1	0.100(810)	+0.000	0.908	0.906(1756)	−0.002

Table 10.1: Results obtained for an imposed constraint of FPR < 0.1, using performance-cautious ensemble generation. The optimal ensemble size, based on the average performance achieved across the optimisation sets (ζ), is indicated in boldface. The best mean performance (μ_ζ) theoretically achievable across the evaluation sets is underlined.

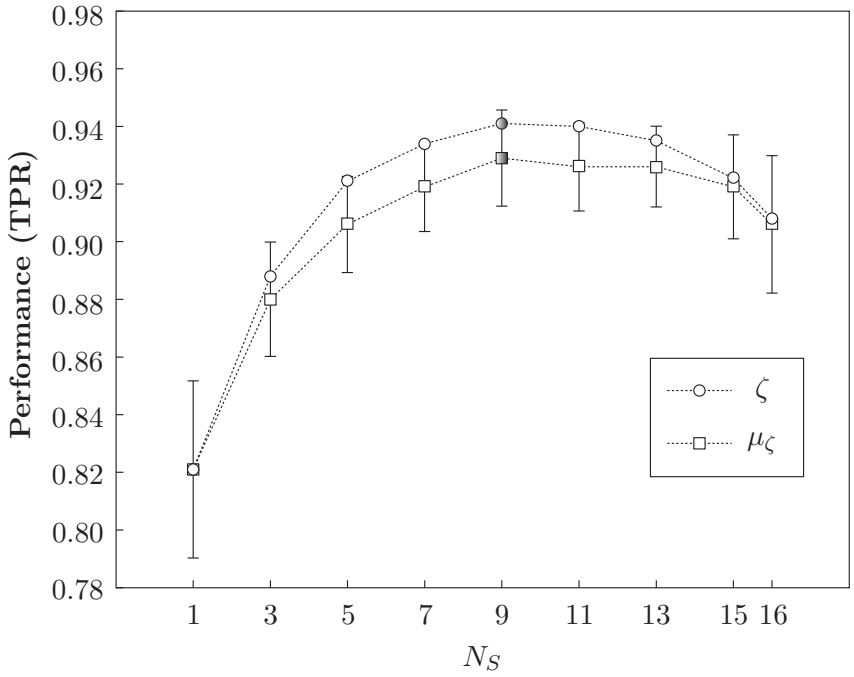


Figure 10.1: Performance for an imposed constraint of FPR < 0.1, using performance-cautious ensemble generation. The best mean performance achieved across the optimisation sets (ζ) and the best mean performance achieved across the evaluation sets (μ_ζ) are indicated by shading. The vertical bars indicate the PS, and are constructed in such a way that they extend by one standard deviation (σ_ζ) from the mean value in both directions. For each value of N_S, the vertical distance between ζ and μ_ζ indicates the PP (generalisation error).

	Operational axis (TPR)			Performance axis (FPR)		
N_S	ξ	μ_ξ(OS)	OP	ζ	μ_ζ(PS)	PP
1	TPR $>$ 0.9	0.904(400)	+0.004	0.169	0.169(1303)	+0.000
3	TPR $>$ 0.9	0.891(1686)	−0.009	0.113	0.133(1249)	+0.020
5	TPR $>$ 0.9	0.890(1977)	−0.010	0.079	0.092(2448)	+0.013
7	TPR $>$ 0.9	0.887(2770)	−0.013	0.069	0.086(3319)	+0.017
9	TPR $>$ 0.9	0.887(1461)	−0.013	0.064	0.075(7588)	+0.011
11	**TPR $>$ 0.9**	**0.889(1665)**	**−0.011**	**0.064**	**0.074(3261)**	**+0.010**
13	TPR $>$ 0.9	0.888(1697)	−0.012	0.068	0.072(2513)	+0.004
15	TPR $>$ 0.9	0.895(1361)	−0.050	0.080	0.081(2081)	+0.001
16	TPR $>$ 0.9	0.903(1522)	+0.030	0.097	0.097(1382)	−0.000

Table 10.2: Results obtained for an imposed constraint of TPR $>$ 0.9, using performance-cautious ensemble generation. The optimal ensemble size, based on the average performance achieved across the optimisation sets (ζ), is indicated in boldface. The best mean performance (μ_ζ) theoretically achievable across the evaluation sets is underlined.

10.2.2 TPR-based constraint

Table 10.2 shows the results for the system using performance-cautious ensemble generation, when a minimum TPR of 0.9 is imposed. For the majority of the ensemble sizes, the minimum imposed TPR is not met, indicated by negative OP-values in the fourth column.

When a TPR-based constraint is imposed, the performance is represented by the FPR. Lower FPRs therefore indicate better performance and a *positive* PP is expected. The optimal ensemble size ($N_S = 11$), which results in a FPR of 0.074, based upon the objective function ζ is indicated in boldface. Note that, by selecting an ensemble of size $N_S = 13$, a better mean performance across the evaluation sets (underlined) is *theoretically* possible, but since the mean optimisation performance (ζ), when ensembles of size $N_S = 11$ are selected, is better than the mean optmisisation performance (ζ) when ensembles of size $N_S = 13$ are selected (0.064 versus 0.068), the result obtained for $N_S = 11$ represents the best *practically* obtainable result.

In Figure 10.2, the mean evaluation performance (μ_ζ), as well as the average performance obtained across the optimisation sets (ζ) are plotted against the ensemble size N_S.

10.2.3 EER-based constraint

Table 10.3 shows the results for the system using performance-cautious ensemble generation, when an EER is imposed.

The operational mean (μ_ξ) is defined as the perpendicular distance between the operating point and the EER line in ROC space ($\mu_\xi = 0$ therefore indicates that the mean performance achieved across the evaluation sets has an EER). A positive operational mean is associated with an operating point to the upper-

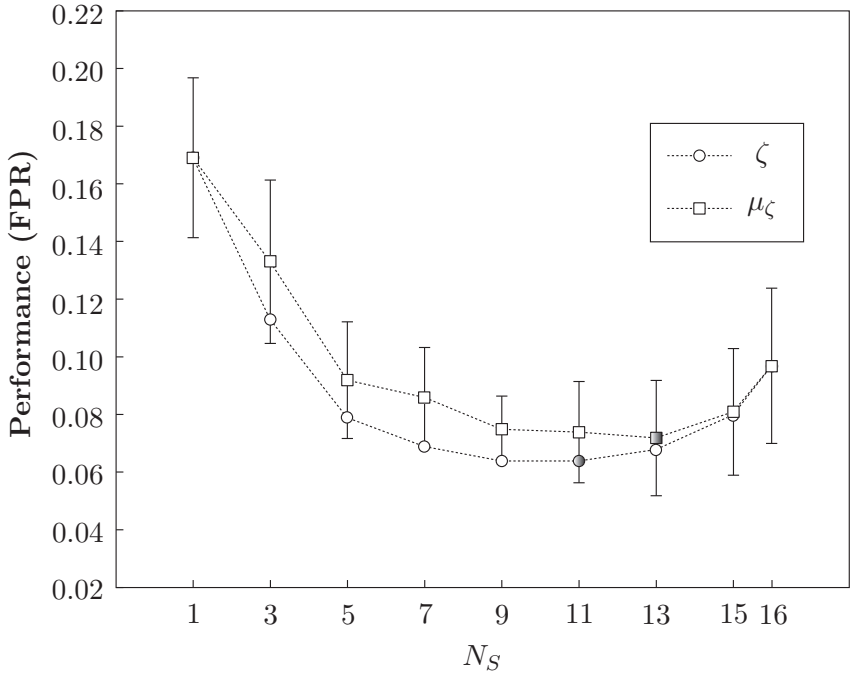


Figure 10.2: Performance for an imposed constraint of TPR > 0.9, using performance-cautious ensemble generation. The best mean performance achieved across the optimisation sets (ζ) and the best mean performance achieved across the evaluation sets (μ_ζ) are indicated by shading. The vertical bars indicate the PS, and are constructed in such a way that they extend by one standard deviation (σ_ζ) from the mean value in both directions. For each value of N_S, the vertical distance between ζ and μ_ζ indicates the PP (generalisation error).

right of the EER line in ROC space, whereas a negative operational mean is associated with an operating point to the lower-left of the EER line. Note that the OP and (μ_ξ) are equivalent in this case. The best practically obtainable result (an EER of 0.088) is achieved when ensembles of size $N_S = 11$ are selected, although an EER of 0.087 is theoretically possible when $N_S = 13$.

In Figure 10.3, the mean evaluation performance (μ_ζ), as well as the average performance obtained across the optimisation sets (ζ) are plotted against the ensemble size N_S.

10.3 Efficiency-cautious ensemble generation

We now discuss the results obtained for the efficiency-cautious ensemble generation method, as introduced in Section 8.3.3. The results are tabulated and plotted for the same criteria that were considered for the performance-cautious scenario. Note that, when $N_S = 1$ or $N_S = 16$, the performance-cautious and efficiency-cautious ensemble techniques are equivalent−the performances reported for these two scenarios are therefore identical.

	Operational axis (FPR = TPR)			Performance axis (FPR = FNR)		
N_S	ξ	μ_ξ(OS)	OP	ζ	μ_ζ(PS)	PP
1	EER	+0.000(1101)	+0.000	0.119	0.129(2107)	+0.010
3	EER	+0.006(1014)	+0.006	0.098	0.121(2180)	+0.023
5	EER	+0.004(1410)	+0.004	0.083	0.105(1779)	+0.022
7	EER	+0.005(1504)	+0.005	0.077	0.097(3897)	+0.020
9	EER	+0.000(2221)	+0.000	0.074	0.090(2609)	+0.015
11	**EER**	**−0.002(2061)**	**−0.002**	**0.074**	**0.088(3307)**	**+0.014**
13	EER	−0.001(1681)	−0.001	0.077	0.087(3062)	+0.010
15	EER	−0.001(1235)	−0.001	0.083	0.092(3307)	+0.009
16	EER	+0.000(925)	+0.000	0.090	0.095(2544)	+0.005

Table 10.3: Results obtained for an EER-based constraint, using performance-cautious ensemble generation. The optimal ensemble size, based on the average performance achieved across the optimisation sets (ζ), is indicated in boldface. The best mean performance (μ_ζ) theoretically achievable across the evaluation sets is underlined.

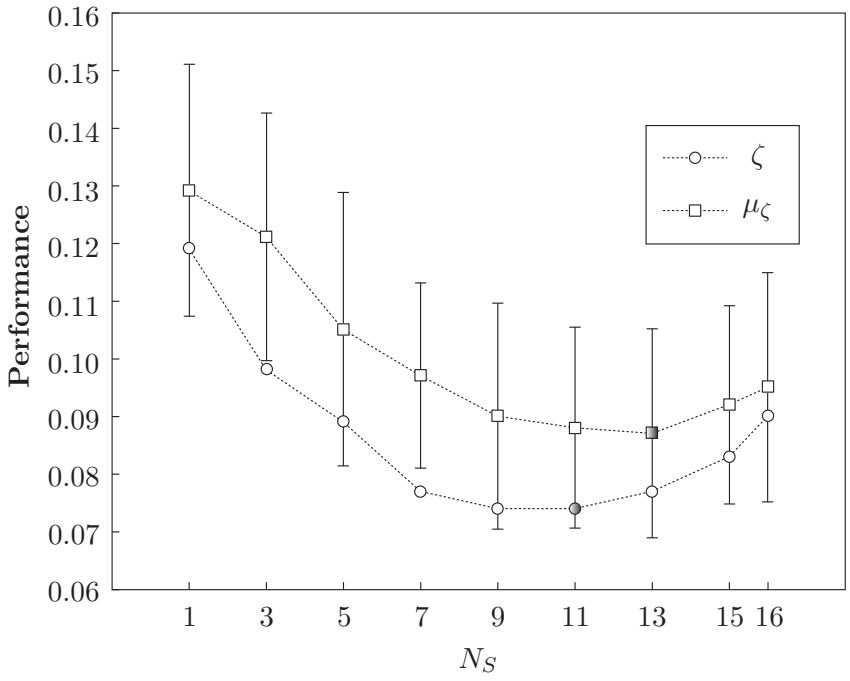


Figure 10.3: Performance for an EER-based constrant, using performance-cautious ensemble generation. The best mean performance achieved across the optimisation sets (ζ) and the best mean performance achieved across the evaluation sets (μ_ζ) are indicated by shading. The vertical bars indicate the PS, and are constructed in such a way that they extend by one standard deviation (σ_ζ) from the mean value in both directions. For each value of N_S, the vertical distance between ζ and μ_ζ indicates the PP (generalisation error).

	Operational axis (FPR)			Performance axis (TPR)		
N_S	ξ	μ_ξ(OS)	OP	ζ	μ_ζ(PS)	PP
1	FPR < 0.1	0.099(1024)	−0.001	0.821	0.821(1059)	−0.000
3	FPR < 0.1	0.097(1386)	−0.003	0.860	0.860(2472)	−0.000
5	FPR < 0.1	0.100(1575)	−0.000	0.891	0.892(4178)	−0.001
7	FPR < 0.1	0.102(2262)	+0.002	0.917	0.915(3664)	−0.002
9	FPR < 0.1	0.104(1987)	+0.004	0.922	0.920(3512)	+0.008
11	**FPR < 0.1**	**0.102(1473)**	**+0.002**	**0.931**	**0.930(2966)**	**−0.001**
13	FPR < 0.1	0.098(1217)	−0.002	0.917	0.916(3510)	−0.001
15	FPR < 0.1	0.096(1232)	−0.004	0.914	0.910(2569)	−0.004
16	FPR < 0.1	0.100(810)	−0.000	0.908	0.906(1756)	−0.002

Table 10.4: Results obtained for an imposed constraint of FPR < 0.1, using efficiency-cautious ensemble generation. The optimal ensemble size, based on the average performance achieved across the optimisation sets (ζ), is indicated in boldface. The best mean performance (μ_ζ) theoretically achievable across the evaluation sets is underlined.

10.3.1 FPR-based criterion

Table 10.4 shows the results for the system using efficiency-cautious ensemble generation, when a maximum FPR of 0.1 is imposed. In Figure 10.4, the mean evaluation performance (μ_ζ), as well as the mean performance obtained across the optimisation sets (ζ) are plotted against the ensemble size N_S.

10.3.2 TPR-based criterion

Table 10.5 shows the results for the system using efficiency-cautious ensemble generation, when a minimum TPR of 0.9 is imposed. In Figure 10.5, the mean evaluation performance (μ_ζ), as well as the mean performance obtained on the optimisation sets (ζ) are plotted against the ensemble size N_S.

10.3.3 EER-based constraint

Table 10.6 shows the results for the system using efficiency-cautious ensemble generation, when an EER is imposed. In Figure 10.6, the mean evaluation performance (μ_ζ), as well as the mean performance obtained on the optimisation sets (ζ) are plotted against the ensemble size N_S.

10.4 Discussion

10.4.1 Performance-cautious versus efficiency-cautious ensemble selection

Since the ensembles generated using the efficiency-cautious strategy constitute a subset of the ensembles generated using the performance-cautious strat-

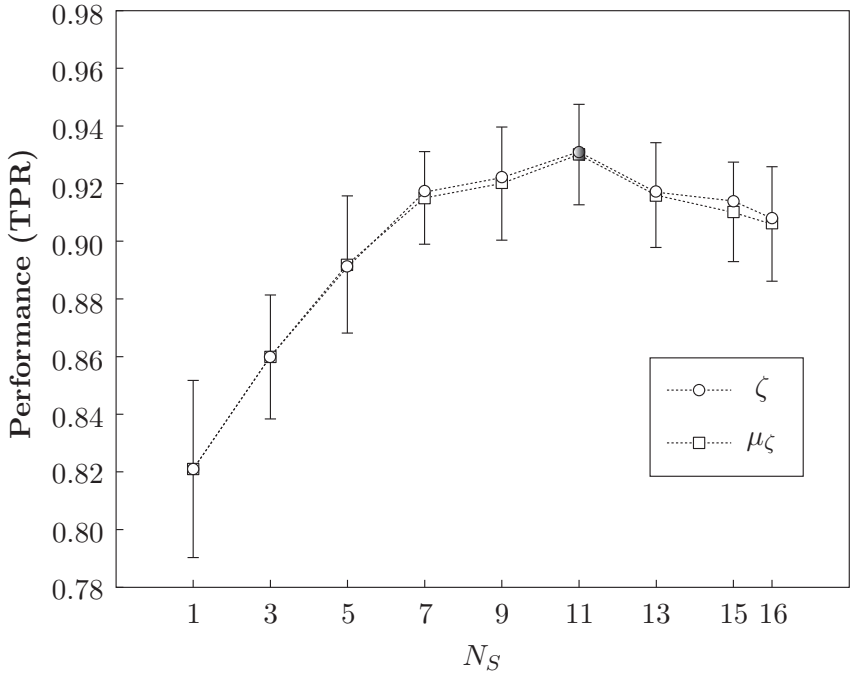


Figure 10.4: Performance for an imposed constraint of FPR < 0.1, using efficincy-cautious ensemble generation. The best mean performance achieved across the optimisation sets (ζ) and the best mean performance achieved across the evaluation sets (μ_ζ) are indicated by shading. The vertical bars indicate the PS, and are constructed in such a way that they extend by one standard deviation (σ_ζ) from the mean value in both directions. For each value of N_S, the vertical distance between ζ and μ_ζ indicates the PP (generalisation error).

	Operational axis (TPR)			Performance axis (FPR)		
N_S	ξ	μ_ξ(OS)	OP	ζ	μ_ζ(PS)	PP
1	TPR > 0.9	0.904(1400)	+0.004	0.169	0.169(1304)	+0.000
3	TPR > 0.9	0.906(1597)	+0.006	0.138	0.137(2351)	−0.001
5	TPR > 0.9	0.903(2607)	+0.003	0.110	0.111(2866)	+0.001
7	TPR > 0.9	0.901(1484)	+0.001	0.083	0.087(4287)	+0.004
9	TPR > 0.9	0.902(2979)	+0.002	0.078	0.083(4084)	+0.005
11	**TPR > 0.9**	**0.902(3037)**	**+0.002**	**0.071**	**0.075(3037)**	**+0.004**
13	TPR > 0.9	0.904(1760)	+0.004	0.081	0.081(2472)	+0.000
15	TPR > 0.9	0.900(1182)	+0.000	0.084	0.085(2253)	+0.001
16	TPR > 0.9	0.903(1522)	+0.003	0.097	0.097(1382)	+0.000

Table 10.5: Results obtained for an imposed constraint of TPR > 0.9, using efficiency-cautious ensemble generation. The optimal ensemble size, based on the average performance achieved across the optimisation sets (ζ), is indicated in boldface. The best mean performance (μ_ζ) theoretically achievable across the evaluation sets is underlined.

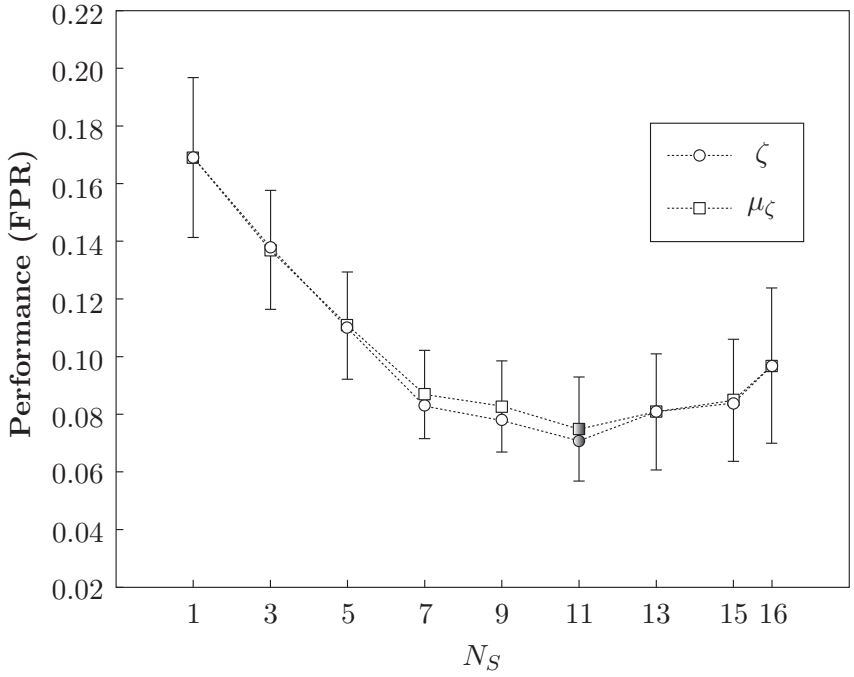


Figure 10.5: Performance for an imposed constraint of TPR > 0.9, using efficiency-cautious ensemble generation. The best mean performance achieved across the optimisation sets (ζ) and the best mean performance achieved across the evaluation sets (μ_ζ) are indicated by shading. The vertical bars indicate the PS, and are constructed in such a way that they extend by one standard deviation (σ_ζ) from the mean value in both directions. For each value of N_S, the vertical distance between ζ and μ_ζ indicates the PP (generalisation error).

	Operational axis (FPR = TPR)			Performance axis (EER)		
N_S	ξ	μ_ξ(OS)	OP	ζ	μ_ζ(PS)	PP
1	EER	+0.000(1101)	+0.000	0.119	0.129(2107)	+0.010
3	EER	+0.002(1737)	+0.002	0.107	0.116(2503)	+0.009
5	EER	+0.002(2689)	+0.002	0.097	0.104(2929)	+0.007
7	EER	+0.004(2236)	+0.004	0.086	0.092(5734)	+0.006
9	EER	+0.004(1907)	+0.004	0.083	0.091(9312)	+0.008
11	**EER**	**+0.002(2059)**	**+0.002**	**0.080**	**0.086(4849)**	**+0.006**
13	EER	+0.002(1828)	+0.002	0.085	0.090(3197)	+0.005
15	EER	−0.001(1255)	−0.001	0.086	0.092(3440)	+0.006
16	EER	+0.000(925)	+0.000	0.090	0.095(2544)	+0.005

Table 10.6: Results obtained for an EER-based constraint, using efficiency-cautious ensemble generation. The optimal ensemble size, based on the average performance achieved across the optimisation sets (ζ), is indicated in boldface. The best mean performance (μ_ζ) theoretically achievable across the evaluation sets is underlined.

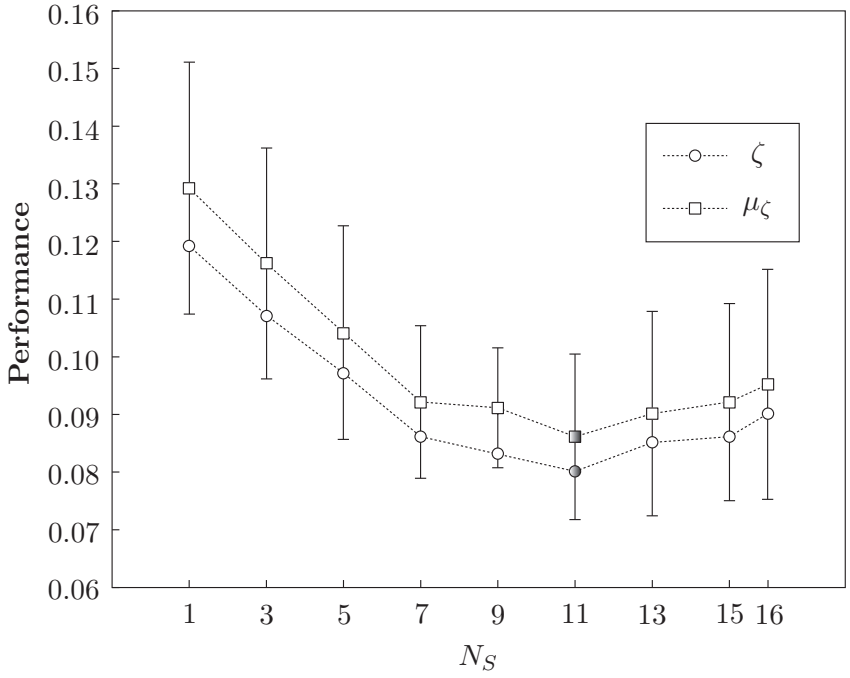


Figure 10.6: Performance for an EER-based constraint, using efficiency-cautious ensemble generation. The best mean performance achieved across the optimisation sets (ζ) and the best mean performance achieved across the evaluation sets (μ_ζ) are indicated by shading. The vertical bars indicate the PS, and are constructed in such a way that they extend by one standard deviation (σ_ζ) from the mean value in both directions. For each value of N_S, the vertical distance between ζ and μ_ζ indicates the PP (generalisation error).

egy, the mean performance (ζ) obtained across the optimisation sets for the performance-cautious approach will necessarily be better than or equal to the mean performance (ζ) obtained when the efficiency-cautious approach is utilised. This is illustrated clearly in Figure 8.8, and further evidenced by examining the tables presented in this chapter. Note that, when $N = 1$ or $N = 16$, the performance-cautious and efficiency-cautious approaches are identical.

In Table 10.7 a comparison of the best result (that is, the mean evaluation performance (μ_ζ) corresponding to the optimal ensemble size, N_S, and best mean optimisation performance (ζ)) obtained for each criterion for the performance-cautious and efficiency-cautious approaches is presented. The best result, for each criterion, is shown in boldface. Note that, for two of the three criteria, the efficiency-cautious approach produces a better mean performance (μ_ζ) across the evaluation sets. Furthermore, the operational mean μ_ξ obtained for two of the three criteria is closer to the imposed criterion when the efficiency-cautious approach is utilised.

In Figure 10.7, the mean performances (μ_ζ and ζ) for the EER-based cri-

	Performance-cautious			Efficiency-cautious		
ξ	N_S	μ_ξ	μ_ζ	N_S	μ_ξ	μ_ζ
FPR < 0.1	9	0.108	0.929	11	0.102	**0.930**
TPR > 0.9	11	0.889	**0.074**	11	0.902	0.075
EER	11	0.002	0.088	11	0.002	**0.086**

Table 10.7: A comparison of the best result (that is, the mean evaluation performance μ_ζ corresponding to the optimal ensemble size, N_S, and best mean optimisation performance ζ) obtained for each criterion for the performance-cautious and efficiency-cautious approaches.

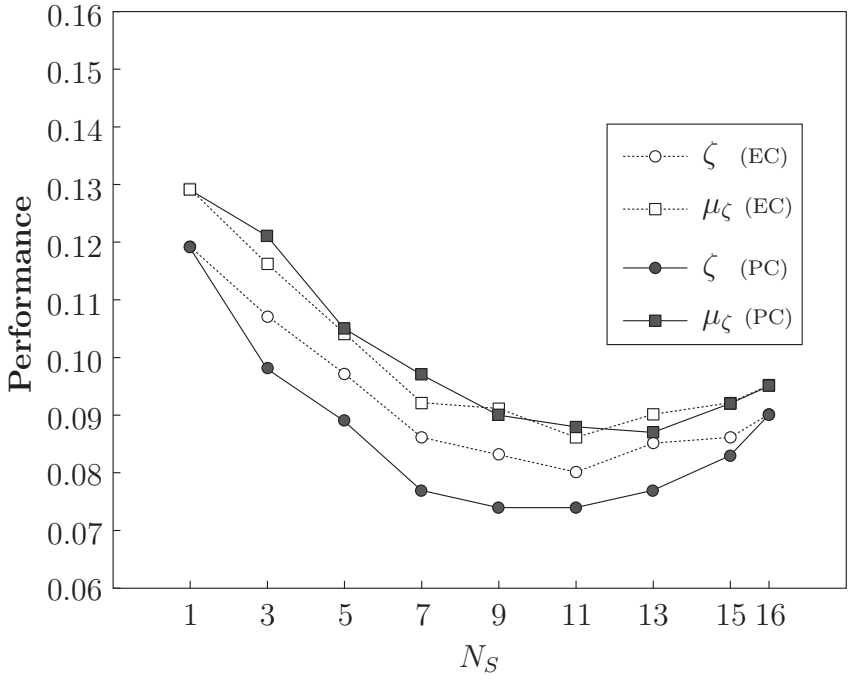


Figure 10.7: A comparison of the EERs achieved using performance-cautious and efficiency-cautious approach. Note that significant overfitting occurrs when the performance-cautious approach is utilised.

terion, for both the performance-cautious and efficiency-cautious approaches, are plotted against N_S. As expected, the mean optimisation performance (ζ) obtained when using the performance-cautious approach is better (lower in the case of an EER) than the mean optimisation performance (ζ), obtained when using the efficiency-cautious approach, for all ensemble sizes, except for $N_S = 1$ and $N_S = 16$. However, for several ensemble sizes ($N_S = 3, 5, 7, 11$) the mean performance obtained across the evaluation sets (μ_ζ), is superior when the efficiency-cautious approach is employed.

It is clear that significant overfitting occurs when the performance-cautious approach is utilised, resulting in a large generalisation error (PP). This over-

Ensemble Ψ_{pc}	Freq.	Ensemble Ψ_{pc}	Freq.
$\{1,2,3,4,5,6,7,8,11,14,16\}$	2	$\{2,3,4,5,6,7,8,10,11,14,16\}$	3
$\{1,2,3,4,5,7,8,9,10,11,16\}$	1	$\{2,3,4,5,6,7,8,11,13,14,16\}$	2
$\{1,2,3,4,5,7,8,9,11,14,16\}$	1	$\{2,3,4,5,7,8,10,11,12,14,16\}$	3
$\{1,2,3,4,5,7,8,10,11,13,16\}$	2	$\{2,3,4,5,7,8,10,11,13,15,16\}$	3
$\{1,2,3,4,5,7,8,10,11,14,16\}$	1	$\{2,3,4,5,7,8,10,11,14,15,16\}$	3
$\{1,2,3,4,5,7,8,10,12,14,16\}$	1	$\{2,3,5,6,7,8,10,11,13,14,16\}$	1
$\{1,2,4,5,6,8,10,11,13,15,16\}$	1	$\{2,4,5,6,7,8,10,11,12,14,16\}$	1
$\{1,2,4,5,7,8,10,11,12,14,16\}$	1	$\{2,4,5,6,7,8,10,11,13,14,16\}$	1
$\{1,2,4,5,7,8,10,11,13,15,16\}$	1	$\{2,4,5,6,7,8,10,11,13,15,16\}$	1
$\{2,3,4,5,6,7,8,10,11,13,16\}$	1		

Table 10.8: Ensembles selected for an EER-based criterion using the performance-cautious approach, for $L = 30$ iterations. The frequency (Freq.) column indicates the number of times that each ensemble is selected. The majority of the ensembles are selected only once.

fitting also explains the relatively poor OP achieved when using the performance-cautious approach, in the sense that the operational mean (μ_ξ) is relatively far from the imposed criterion (ξ).

When significant overfitting occurs (as is the case with the performance-cautious approach) one would expect different ensembles to be selected during each iteration. This is a reasonable expectation, since, if the optimal ensembles selected for each optimisation set (utilised during a specific iteration) are the same, it would imply that said ensemble is adept at accurately classifying all the signatures (for all the iterations), and no overfitting occurs. Conversely, if a different ensemble is selected for each optimisation set (utilised during a specific iteration), it implies that the optimal ensemble in each case, while being optimal for a specific set of writers, is not optimal for a different subset of writers. This results in overfitting.

Since ensembles of size $N_S = 11$ are optimal in most cases, we now take a closer look at said ensembles. In Table 10.8, the selected ensembles (for an EER-based criterion), that is, Ψ_{pc}, for the $L = 30$ optimisation sets are shown. The number of times (frequency) that each ensemble is selected is also shown. Note that there are nineteen unique ensembles (out of thirty), and that the majority (twelve) of these ensembles are only optimal for a single optimisation set (iteration), in the sense that the frequency equals one.

In Table 10.9 the selected ensembles (for an EER-base criterion) when the efficiency-cautious approach is employed are shown. Note that there are only three unique ensembles, with one ensemble selected for twenty-two of the thirty optimisation sets (iterations). This suggests that the AUC-based rank of retinas is stable across the writers (and therefore across the optimisation sets), which results in minimal overfitting.

Ensemble Ψ_{ec}	Freq.
$\{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 16\}$	5
$\{2, 3, 4, 5, 6, 7, 8, 10, 11, 14, 16\}$	22
$\{2, 4, 5, 6, 7, 8, 9, 10, 11, 14, 16\}$	3

Table 10.9: Ensembles selected for an EER-based criterion using the efficiency-cautious approach, for $L = 30$ iterations. The frequency (Freq.) column indicates the number of times that each ensemble is selected. Three ensembles are selected, with one ensemble clearly favoured.

Retina	16	5	8	11	2	7	4	6
Average rank	1.0	2.3	2.7	4.0	5.6	5.7	7.6	8.0
Retina	10	14	3	9	1	13	12	15
Average rank.	8.8	10.3	10.5	11.6	13.3	13.9	14.8	16.0

Table 10.10: The average AUC-based rank (from 1 to 16, where 1 indicates that the continuous classifier associated with the retina has the highest AUC) of each retina, for the efficiency-cautious ensemble generation approach.

10.4.2 Selected retinas

We now take a closer look at the performance of the individual retinas for the optimal EER result (that is, the EER achieved when the efficiency-cautious approach is used to select ensembles of size $N_S = 11$). The average AUC-based rank (from 1 to 16, where 1 indicates that the continuous classifier associated with the retina has the highest AUC) of each retina is shown in Table 10.10. The global retina (retina 16) has an average rank of 1.0, which implies that the global retina is the highest ranked retina for each optimisation set, without exception. Similarly, retina 15 (the lower-rightmost retina) is the lowest ranked retina for each optimisation set (see Figure 9.11). Note that this rank order provides an indication of which ensembles are selected for smaller ensemble sizes. For example, when $N_S = 3$, the first three retinas in the table, that is retinas 16, 5 and 8, constitute the optimal (efficiency-cautious) ensemble, across the *majority* of the iterations (optimisation sets).

In Figure 10.8, the local retinas that form part of the optimal ensemble $\{2, 3, 4, 5, 6, 7, 8, 10, 11, 14, 16\}$ are superimposed onto several signatures associated with different writers. The global retina (retina 16) is not shown.

Figure 10.8: Local retinas that form part of the optimal ensemble $\{2, 3, 4, 5, 6, 7, 8, 10, 11, 14, 16\}$ superimposed onto several signatures associated with different writers. The global retina (retina 16) is not shown.

Chapter 11

Conclusion and Future Work

11.1 Comparison with previous work

The main objective of this dissertation was to investigate whether a significant improvement in system performance is possible by also utilising *local* DRT-based features. An EER of 12.2% is reported in Coetzer (2005). The system proposed in Coetzer (2005) is similiar to the system developed in this dissertation, in the sense that it also uses DRT-based feature extraction techniques and continuous observation, ring-structured HMMs. However, in Coetzer (2005), only global features are extracted, and the resubstituion method is used for performance evaluation (see Section 9.3.1) which is optimistically biased. We therefore repeated Coetzer's experiments (by setting $N_S = 1$ and therefore selecting only the global retina) using the protocol adopted in this dissertation (that is, 3-fold cross-validation with shuffling). An EER of 12.9% is subsequently obtained for Coetzer's system, which constitutes a directly comparable result.

We have therefore shown that the inclusion of local features and classifier combination, when evaluated on the same data set and when using the same evaluation protocol, improves the EER reported in Coetzer (2005) by 33.33% (from 12.90% to 8.6%).

Furthermore, we evaluated our system using a robust evaluation protocol, thereby ensuring that the results reported are not optimistically biased, and may therefore be trusted as indicative of potential real-world performance.

We also demonstrated that the system is robust with respect to variations in rotation, scale and translation of questioned signatures.

Three other off-line signature verification systems (including the system proposed in Coetzer (2005)) have previously been evaluated on Dolfing's data set. We are therefore able to directly compare the proficiency of our system (when operating with an EER) with said systems. We present the EER achieved for each of the above-mentioned systems in Table 11.1. Note that two of these systems are presented in Swanepoel and Coetzer (2010), and we

Author(s)	EER
J. Dolfing (1998)	13.3%
J. Coetzer, B. Herbst, and J. du Preez (2005)	12.9%
J. Swanepoel and J. Coetzer (2010) (SA)	11.21%
J. Swanepoel and J. Coetzer (2010) (MV)	10.23%
M. Panton and J. Coetzer (2010)	**8.6%**

Table 11.1: A comparison of EERs achieved by off-line signature verification systems evaluated on Dolfing's data set.

include both results. Our system significantly outperforms all of these systems.

Although the systems developed in Swanepoel and Coetzer (2010) are similar to the system developed in this dissertation (in the sense that both of these systems also utilise HMMs and local features), the features utilised in Swanepoel and Coetzer (2010) are significantly different from those employed in this dissertation. It may therefore be possible to combine the above-mentioned systems in order to obtain a superior hybrid system.

11.2 Future work

In this section, we address a number of strategies we believe are worth investigating in future research.

11.2.1 Fusion strategies

In this dissertation, we only investigated the combination of classifiers by majority voting. In possible future work, we may consider several more sophisticated decision-level fusion strategies (for example, weighted majority voting (Kuncheva (2004)), Haker's algorithm (Haker *et al.* (2005)), and iterative boolean combination (Khreich *et al.* (2010)), as well as score-level fusion strategies (for example, simple averaging, weighted averaging, the trimmed mean, etc. (Kuncheva (2004)).

11.2.2 Genetic search algorithms

In Section 8.3.1 we showed that the total number of possible ensembles is given by the expression $\binom{N_r}{N_S} \cdot X^{N_S}$, where N_r denotes the number of defined retinas, N_S denotes the size of the selected ensemble and X denotes the number of imposed threshold values. The number of candidate ensembles to consider becomes prohibitively large as more retinas are considered. We introduced two approaches to reduce the total search space, namely "performance-cautious ensemble generation" and "efficiency-cautious ensemble generation".

The utilisation of a performance-based single objective genetic search algorithm (see Mitchell (1998) and Coello Coello *et al.* (2007)) instead of the ensemble generation and selection strategies implemented in this dissertation, may result in superior performance.

Since a genetic search algorithm allows for a locally optimal solution to be found relatively efficiently in an expansive search space, genetic search algorithms will make an investigation into utilising significantly more (than 16) retinas computationally feasible. This can be accomplished by either increasing the number of centroids defined in the zoning process, or by defining multiple, different sized retinas for each centroid.

11.2.3 Adaptive ensemble selection

The overfitting that occurs when the performance-cautious ensemble generation approach is utilised implies that the ensembles selected using the optimisation set, in each case, are not optimal when used to classify the signatures in the corresponding evaluation set. Although this overfitting is undesirable, it suggests that superior performance can be obtained by simply modifying the ensemble selection strategy.

An *adaptive* ensemble selection strategy that utilises *writer-specific* attributes should prove beneficial. The system proposed in this dissertation makes *writer-independent* ensemble selections.

In a theoretical scenario where negative signatures are available for each evaluation writer at the time of enrolment (thereby constituting a set $\boldsymbol{T}_E^-$) it will possible to select writer-specific ensembles by simply employing the ensemble generation and selection strategies described in Chapter 8 for *each* writer. However, the absence of a set $\boldsymbol{T}_E^-$ in our (practical) scenario makes this approach to writer-specific ensemble selection infeasible.

In our scenario, a writer-specific ensemble selection approach must necessarily be based on the information contained in the set $\boldsymbol{T}_E^+$. Although such a strategy is beyond the scope of this dissertation, we outline an adaptive selection protocol that can be investigated in future research.

By examining the superimposed retinas in Figure 10.8, it is clear that retinas that *generally* correspond to areas of a signature image that contain the least amount of signature information are *not* selected. For example, the poorest performing retina (retina 15) corresponds to the lower-righthand region of a signature image—a region that generally contains little or no signature information. An adaptive approach may use the training signatures $\boldsymbol{T}_E^+$ to rank the black pixel-density associated with each retina, for a *specific* writer. These statistics can then be incorporated into the writer model. The ensemble generation and selection strategies will be similar to the strategies described in this dissertation, with the exception that the retina labels ("retina 1", "retina 2", etc), instead of the retina's location, refer to the black pixel density of each retina (which is not writer-specific). If, for example, the optimal ensemble

selected using the optimisation set consists of base classifiers associated with retinas 1, 2 and 6, this implies that, for each writer in the evaluation set, decisions obtained from the 1^{st}, 2^{nd} and 6^{th} densest retinas (by imposing the appropriate selected threshold) should be combined.

In addition to ranking each retina based on black pixel-density, the "consistency" of each retina (among the training samples associated with a specific writer) can be exploited[1]. An optimal selected ensemble may then, for example, imply that "the decision obtained from the two most "consistent" retinas should be combined with the decisions obtained from the three "densest" retinas".

Although an adaptive ensemble selection approach has not been implemented in this dissertation, a superior performance can be expected.

[1]In fact, the statistic σ_w^r defined in Equation 7.5.2 already quantifies this "consistency" to a certain extent.

List of References

Armand, S., Blumenstein, M. and Muthukkumarasamy, V. (2006). Off-line signature verification using the enhanced modified direction feature and neural-based classification. In: *Proceedings of the International Joint Conference on Neural Networks*, pp. 1663–1669.

Baltzakis, H. and Papamarkos, N. (2001). A new signature verification technique based on a two-stage neural network classifier. *Engineering Applications of Artificial Intelligence*, vol. 1, pp. 95–103.

Bastista, L., Rivard, D., Sabourin, R., Granger, E. and Maupin, P. (2007). *Pattern Recognition Technologies and Applications: Recent Advances*, chap. 3. 1st edn. IGI Global.

Batista, L., Granger, E. and Sabourin, R. (2009). Improving performance of HMM-based off-line signature verification systems through a multi-hypothesis approach. *International Journal on Document Analysis and Recognition*, vol. 13, pp. 33–47.

Bertolini, D., Oliveira, L.S., Justino, E. and Sabourin, R. (2008). Ensemble of classifiers for off-line signature verification. *Systems, Man and Cybernetics*, pp. 283–288.

Bertolini, D., Oliveira, L.S., Justino, E. and Sabourin, R. (2009). Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers. *Pattern Recognition*, vol. 43, pp. 387–396.

Bortolozzi, E.J.R.J.F. and Sabourin, R. (2004). A comparison of SVM and HMM classifiers in the off-line signature verification. *Pattern Recognition Letters*, vol. 26, pp. 1377–1385.

Coello Coello, C.A., Lamont, G.B. and Veldhuizen, D.A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer.

Coetzer, J. (2005). *Off-line signature verification*. Ph.D. thesis, Stellenbosch University.

Coetzer, J., Herbst, B. and du Preez, J. (2004). Off-line signature verification using the discrete Radon transform and a hidden Markov model. *EURASIP Journal on Applied Signal Processing*, vol. 4, pp. 559–571.

Coetzer, J. and Sabourin, R. (2007). A human-centric off-line signature verification system. In: *Ninth International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 153–157.

Deng, P.S., Liao, H.Y.M., Ho, C.W. and Tyan, H.R. (1999). Wavelet-based off-line handwritten signature verification. *Computer Vision and Image Understanding*, vol. 76, pp. 173–190.

Dolfing, J. (1998). *Handwriting Recognition and Verification - A hidden Markov approach.* Ph.D. thesis, Eindhoven University of Technology.

Einsele, F., Ingold, R. and Hennbert, J. (2008). A language-independent, open-vocabulary system based on HMMs for recognition of ultra low resolution words. *Journal of Universal Computer Science*, vol. 14, no. 18, pp. 2982–2997.

El-Yacoubi, A., Justino, E., Sabourin, R. and Bortolozzi, F. (2000). Off-line signature verification using HMMs and cross-validation. In: *IEEE Worskhop on Neural Networks for Signal Processing*, pp. 859–868.

Fang, B., Leung, C.H., Tand, Y.Y., Tse, K.W., Kwok, P.C.K. and Wong, Y.K. (2003). Off-line signature verification by tracking of feature and stroke positions. *Pattern Recognition*, vol. 36, pp. 91–101.

Fang, B., Leung, C.H., Tang, Y.Y., Kwok, P.C.K., Tse, K.W. and Wong, Y.K. (2002). Off-line signature verification with generated training samples. *IEE Proceedings - Vision, Image and Signal Processing*, vol. 149, pp. 85–90.

Fang, B., Wang, Y.Y., Leung, C.H. and Tse, K.W. (2001). Off-line signature verification by the analysis of cursive strokes. *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, pp. 659–673.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition letters*, vol. 27, pp. 861–874.

Ferrer, M., Alonso, J. and Travieso, C. (2005). Off-line geometric parameters for automatic signature verification using fixed-point arithmetic. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 993–997.

Forney, G. (1973). The viterbi algorithm. In: *Proceedings of the sixth international conference on handrwriting and drawing.*, vol. 61, pp. 268–278.

Gonzalez, R.C. and Woods, R.E. (2002). *Digital Image Processing*. Prentice Hall.

Guo, J.K., Doermann, D. and Rosenfeld, A. (2001). Forgery detection by local correspondence. *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 4, pp. 579–641.

Haker, S., Wells III, W.M., Warfield, S.K., Talos, I., Bhagwat, J.G., Goldberg-Zimring, D., Mian, A., Ohno-Machado, L. and Zou, K.H. (2005). Combining classifiers using their receiver operating characteristics and maximum likelihood estimation. *Medical Image Computing and Computer-Assisted Intervention*, vol. 3749, pp. 506–514.

Jain, A., Nandakumara, K. and Ross, A. (2005). Score normalization in multimodal biometric systems. *Pattern Recognition*, vol. 38, p. 2270 2285.

Justino, E.J.R., Bortolozzi, F. and Sabourin, R. (2001). Off-line signature verification using HMM for random, simple and skilled forgeries. In: *International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 105–110.

Khreich, W., Granger, E., Miri, A. and Sabourin, R. (2010). Iterative boolean combination of classifiers in the roc space: An application to anomaly detection with HMMs. *Pattern Recognition*, vol. 43.

Kuncheva, L.I. (2004). *Combining Pattern Classifiers*. John Wiley and Sons, Inc.

Macskassy, S.A. and Provost, F. (2004). Confidence bands for roc curves: Methods and an empirical study. In: *Proceedings of the FirstWorkshop on ROC Analasis in AI*.

Madusa, V.K., Yusof, M.H.M., Hanmandlu, M. and Kubik, K. (2003). Automatic extraction of signatures from bank cheques and other documents. In: *Proceedings of the seventh conference on Digital Image Computing: Techniques and Applications drawing*.

Maiorana, E., Campisi, P. and Neri, A. (2007). Biometric signature authentication using radon transform-based watermarking techniques. pp. 1–6.

Martinez, L., Travieso, C., Alonso, J. and Ferrer, M. (2004). Parametrization of a forgery hand-written signature verification system using SVM. pp. 193–196.

Mehta, M., Choudhary, V., Das, R. and Khan, I. (2010). Offline signature verification and skilled forgery detection using HMM and sum graph features with ANN and knowledge based classifier. *Proceedings of SPIE, the International Society for Optical Engineering*, vol. 2.

Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. The MIT Press.

Mizukami, Y., Yoshimura, M., Miike, H. and Yoshimura, I. (2002). An off-line signature verification system using an extracted displacement function. *Pattern Recognition Letters*, vol. 23, pp. 1569–1577.

Ozgunduz, E., Senturk, T. and Karsligil, E. (2005). Off-line signature verification and recognition by support vector machine. In: *Proceedings of the European Signal Processing Conference*.

Plamondon, R. and Shihari, S.N. (2000). On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 63–84.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceeding of the IEEE.*, vol. 77, pp. 257 – 286.

Ross, A.A. (2003). *Information Fusion in Fingerprint Authentication*. Ph.D. thesis, Michigan State University.

Shapiro, V. and Bakalov, I. (1993). Static signature verification as a dynamic programming problem. In: *Proceedings of the sixth international conference on handrwriting and drawing.*

Swanepoel, J. and Coetzer, J. (2010). Off-line signature verification using flexible grid features and classier fusion. In: *Proceedings of the 12th International Conference on the Frontiers of Handwriting Recognition (ICFHR).*, pp. 297–302.

Toft, P. (1996). *The Radon Transform - Theory and Implementation*. Ph.D. thesis, Technical University of Denmark.

Vapinik, V. (1998). *Statistical Learning Theory*. Wiley.

Varga, A.P. and Moore, R.K. (1990). Hidden markov model decomposition of speech and noise. In: *Acoustics, Speech, and Signal Processing*, vol. 2, pp. 845–848.

Velez, J., Sanchez, A. and Moreno, A. (2003). Robust off-line signature verification using compression networks and position cuttings. pp. 627–636.

Wang, C., Li, Y., Zhang, H. and Wang, L. (2008). Multi-modal biometric based on FAR-score normalization. *International Journal of Computer Science and Network Security*, vol. 8, no. 4, pp. 250–254.

You, J., Liu, G. and Perkis, A. (2010). A semantic framework for video genre classification and event analysis. *Signal Processing: Image Communication*, vol. 25, no. 4, pp. 287–302.