



UNIVERSITEIT • STELLENBOSCH • UNIVERSITY

Off-line Signature Verification using Classifier Ensembles and Flexible Grid Features

by

Jacques Philip Swanepoel, BScHons (Stell)

*Thesis presented in partial fulfilment of the requirements
for the degree of*

Master of Science in Applied Mathematics

at Stellenbosch University

Department of Applied Mathematics,
Stellenbosch University,
Private Bag X1, 7602, Matieland, South Africa.

Supervisor: Dr Johannes Coetzer, PhD (Stell)

2009

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

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:
J.P. Swanepoel

Date:

Copyright © 2009 Stellenbosch University
All rights reserved.

Abstract

In this study we investigate the feasibility of combining an ensemble of eight continuous base classifiers for the purpose of off-line signature verification. This work is mainly inspired by the process of cheque authentication within the banking environment. Each base classifier is constructed by utilising a specific local feature, in conjunction with a specific writer-dependent signature modelling technique. The local features considered are pixel density, gravity centre distance, orientation and predominant slant. The modelling techniques considered are dynamic time warping and discrete observation hidden Markov models. In this work we focus on the detection of high quality (skilled) forgeries.

Feature extraction is achieved by superimposing a grid with predefined resolution onto a signature image, whereafter a single local feature is extracted from each signature sub-image corresponding to a specific grid cell. After encoding the signature image into a matrix of local features, each column within said matrix represents a feature vector (observation) within a feature set (observation sequence). In this work we propose a novel *flexible* grid-based feature extraction technique and show that it outperforms existing *rigid* grid-based techniques.

The performance of each continuous classifier is depicted by a receiver operating characteristic (ROC) curve, where each point in ROC-space represents the true positive rate and false positive rate of a threshold-specific discrete classifier. The objective is therefore to develop a combined classifier for which the area-under-curve (AUC) is maximised - or for which the equal error rate (EER) is minimised.

Two disjoint data sets, in conjunction with a cross-validation protocol, are used for model optimisation and model evaluation. This protocol avoids possible model overfitting, and also scrutinises the generalisation potential of each classifier. During the first optimisation stage, the grid configuration which maximises proficiency is determined for each base classifier. During the second optimisation stage, the most proficient ensemble of optimised base classifiers is determined for several classifier fusion strategies. During both optimisation stages only the optimisation data set is utilised. During evaluation, each optimal classifier ensemble is combined using a specific fusion strategy, and retrained and tested on the separate evaluation data set. We show that the performance of the optimal combined classifiers is significantly better than that of the optimal individual base classifiers.

Both score-based and decision-based fusion strategies are investigated, which includes a novel extension to an existing decision-based fusion strategy. The existing strategy is based on ROC-statistics of the base classifiers and maximum likelihood estimation. We show that the proposed *elitist* maximum attainable ROC-based strategy outperforms the existing one.

Opsomming

In hierdie projek ondersoek ons die haalbaarheid van die kombinasie van agt kontinue basis-klassifiseerders, vir statiese handtekeningverifikasie. Hierdie werk is veral relevant met die oog op die bekragtiging van tjeks in die bankwese. Elke basis-klassifiseerder word gekonstrueer deur 'n spesifieke plaaslike kenmerk in verband te bring met 'n spesifieke skrywer-afhanklike handtekeningmodelleringstegniek. Die plaaslike kenmerke sluit pik-seldigheid, swaartepunt-afstand, oriëntasie en oorheersende helling in, terwyl die modelleringstegnieke dinamiese tydsverbuiging en diskrete verskuilde Markov modelle insluit. Daar word op die opsporing van hoë kwaliteit vervalsings gefokus.

Kenmerk-onttreking word bewerkstellig deur die superponering van 'n rooster van voorafgedefinieerde resolusie op 'n bepaalde handtekening. 'n Enkele plaaslike kenmerk word onttrek vanuit die betrokke sub-beeld geassosieer met 'n spesifieke roostersel. Nadat die handtekeningbeeld na 'n matriks van plaaslike kenmerke getransformeer is, verteenwoordig elke kolom van die matriks 'n kenmerkvektor in 'n kenmerkstel. In hierdie werk stel ons 'n nuwe *buiigsame* rooster-gebaseerde kenmerk-onttrekkingstegniek voor en toon aan dat dit die bestaande *starre* rooster-gebaseerde tegnieke oortref.

Die prestasie van elke kontinue klassifiseerder word voorgestel deur 'n ROC-kurwe, waar elke punt in die ROC-ruimte die ware positiewe foutkoers en vals positiewe foutkoers van 'n drempel-spesifieke diskrete klassifiseerder verteenwoordig. Die doelwit is derhalwe die ontwikkeling van 'n gekombineerde klassifiseerder, waarvoor die area onder die kurwe (AUC) gemaksimeer word - of waarvoor die gelyke foutkoers (EER) geminimeer word.

Twee disjunkte datastelle en 'n kruisverifiëringsprotokol word gebruik vir model optimering en model evaluering. Hierdie protokol vermy potensiële model-oorpasing, en ondersoek ook die veralgemeningspotensiaal van elke klassifiseerder. Tydens die eerste optimeringsfase word die rooster-konfigurasië wat die bekwaamheid van elke basis-klassifiseerder maksimeer, gevind. Tydens die tweede optimeringsfase word die mees bekwame groepering van geoptimeerde basis-klassifiseerders gevind vir verskeie klassifiseerder fusie-strategieë. Tydens beide optimeringsfasies word slegs die optimeringsdatastel gebruik. Tydens evaluering word elke optimale groep klassifiseerders gekombineer met 'n spesifieke fusie-strategie, her-afgerig en getoets op die aparte evalueringsdatastel. Ons toon aan dat die prestasie van die optimale gekombineerde klassifiseerder aansienlik beter is as dié van die optimale individuele basis-klassifiseerders.

Beide telling- en besluit-gebaseerde fusie-strategieë word ondersoek, insluitend 'n nuwe uitbreiding van 'n bestaande besluit-gebaseerde kombinasie strategie. Die bestaande strategie is gebaseer op die ROC-statistiek van die basis-klassifiseerders en maksimum aanneemlikheidsberaming. Ons toon aan dat die voorgestelde *elitistiese* maksimum haalbare ROC-gebaseerde strategie die bestaande strategie oortref.

Acknowledgements

I would like to express my most sincere gratitude toward the following persons and institutions for their role in the successful completion of this study.

- Dr Johannes Coetzer, for his immense eagerness and invaluable insights regarding the development of both existing and novel concepts throughout this study, as well as his contribution to my development as a researcher.
- Dr Hans Dolfing, for allowing the use of his dynamic signature database.
- Stellenbosch University, for their financial assistance.
- The National Research Foundation, for their financial assistance.
- My friends and colleagues, for their moral support.
- My parents, for their patience, love and support.

Table of Contents

Declaration	i
Abstract	ii
Opsomming	iii
Acknowledgements	iv
List of Figures	viii
List of Tables	xii
List of Symbols	xiv
List of Acronyms	xvi
1 Introduction	1
1.1 Background	1
1.2 Key concepts	2
1.2.1 Pattern recognition	2
1.2.2 Combined classifiers	4
1.2.3 Automatic identification systems	4
1.2.4 Biometric authentication	6
1.2.5 Handwritten signatures	6
1.2.6 Recognition and verification	7
1.2.7 Writer-dependent and writer-independent verification	8
1.2.8 Performance measures	8
1.2.9 On-line and off-line signatures	10
1.2.10 Forgery types	11
1.3 Objectives	13
1.4 System overview	13
1.4.1 System design	15
1.4.2 Data	17
1.4.3 Results	17
1.5 Contribution of this study	18
1.5.1 A novel feature extraction technique	18
1.5.2 A novel classifier ensemble combination strategy	18

1.5.3	A novel off-line signature verification system	19
1.6	Thesis outline	19
2	Literature Study	21
2.1	Introduction	21
2.2	Simple distance classifiers	21
2.3	Dynamic time warping	22
2.4	Hidden Markov models	23
2.5	Neural networks	23
2.6	Support vector machines	24
2.7	Combined classifiers	24
2.8	Concluding remarks	25
3	Image Processing and Feature Extraction	26
3.1	Introduction	26
3.2	Noise removal	27
3.3	Signature segmentation	27
3.3.1	Rigid grid segmentation	27
3.3.2	Flexible grid segmentation	29
3.4	Feature extraction	30
3.4.1	Pixel density	31
3.4.2	Gravity centre distance	32
3.4.3	Orientation	32
3.4.4	Predominant slant	33
3.4.5	Feature vector construction	34
3.5	Vector quantisation	35
3.5.1	Overview	35
3.5.2	Implementation	36
3.6	Concluding remarks	37
4	Signature Modelling and Verification	38
4.1	Introduction	38
4.2	Overview	38
4.2.1	Modelling	39
4.2.2	Verification	39
4.3	Dynamic time warping	40
4.3.1	Overview	40
4.3.2	Model training	41
4.3.3	Verification	43
4.4	Hidden Markov models	43
4.4.1	Overview	43
4.4.2	Notation	44
4.4.3	HMM topology	45
4.4.4	Model training	46
4.4.5	Verification	47
4.5	Concluding remarks	47

5	Classifier Combination	48
5.1	Introduction	48
5.2	Score-based fusion	48
5.3	Decision-based fusion	49
5.3.1	Voting	49
5.3.2	ROC-based combination	49
5.4	Concluding remarks	55
6	Experiments	56
6.1	Introduction	56
6.2	Data	56
6.3	Experimental protocol	57
6.3.1	Data set partitioning	58
6.3.2	Model optimisation	60
6.3.3	Model evaluation	61
6.3.4	Cross-validation	62
6.4	Results	62
6.4.1	Base classifiers	63
6.4.2	Combined classifiers	68
6.5	Contributions	70
6.5.1	Flexible grid-based feature extraction	70
6.5.2	Elitist MAROC-based classifier ensemble combination	75
6.6	Comparison with previous work	80
6.7	Concluding remarks	81
7	Conclusion and Future Work	83
7.1	Conclusion	83
7.2	Future work	84
7.2.1	Adaptive grid segmentation	84
7.2.2	Conditionally independent classifier ensembles	84
7.2.3	The writer-independent approach	85
	Bibliography	86
A	Dynamic Time Warping: Key Concepts	89
A.1	Algorithm	89
B	Hidden Markov Models: Key Concepts	92
B.1	The three basic problems of HMMs	92
B.2	The Viterbi algorithm	93
B.3	Training	94
B.3.1	Parameter optimisation	94
B.3.2	Multiple observation sequences	95
B.3.3	Implementation issues	95
C	Haker's Algorithm	97

List of Figures

1.1	The pattern recognition process.	2
1.2	Categorisation of popular features associated with off-line signatures.	3
1.3	Categorisation of popular classification techniques.	4
1.4	Categorisation of automatic identification systems.	5
1.5	Categorisation of popular biometric authentication systems.	6
1.6	The Declaration of Independence of the United States of America, as signed by 56 delegates of US Congress on July 4 th , 1776.	7
1.7	Hypothetical representations of the performance measures associated with two continuous classifiers C_A and C_B . (a) The FRR, FAR and EER. (b) The ROC-curve and AUC measure, as well as the EER-based optimal discrete classifier $C_B(\tau^*)$	10
1.8	Categorisation of several off-line forgery types, increasing in quality from left to right.	11
1.9	Typical examples of (a) a genuine signature, as well as (b) professional skilled, (c) amateur skilled and (d) random forgeries.	13
1.10	Schematic representation of a combined classifier ensemble as developed in this study. Each entity C_i represents a separate base classifier, as illustrated in Figure 1.11.	14
1.11	Schematic representation of a base classifier as developed in this study. The writer model entity therefore represents either an HMM or a signature template for DTW.	14
3.1	Signature image noise removal by means of the AMF, as implemented by Swanepoel (2007). (a) A signature image containing synthetically generated impulse noise, as well as high density noise regions. (b) The corrected image, as obtained by implementing the standard median filter. Although the impulse noise is successfully removed, the median filter is incapable of correcting areas possessing high density noise. (c) The corrected image, as obtained by implementing the AMF. Practically all traces of noise have been removed	28
3.2	The rigid grid segmentation strategy. (a) The original signature image I , (b) a 3×4 rigid segmentation grid and (c) the resulting image segmentation $\{I_{ij}\}$	29

3.3	The flexible grid segmentation strategy. (a) The original signature image I . (b) A $3_{0.25} \times 4_{0.25}$ flexible grid, where the dotted lines indicate a 3×4 rigid grid, whilst the shaded areas indicate the degree of overlap between adjacent grid cells. (c) The resulting image segmentation $\{I_{ij}\}$. Note that a portion of I is shared between each pair of adjacent grid cells. Also note that each flexible grid cell is dynamically sized according to its proximity to the grid perimeter.	31
3.4	Computation of the gravity centre distance feature.	32
3.5	Computation of the orientation feature. Indicated with dotted lines alongside the segmented image region is the ellipse possessing the same second moments, as well as its major axis. The angle ϕ denotes the resulting orientation feature value.	33
3.6	Computation of the predominant slant feature. The image skeleton clearly exposes numerous straight line segments to be identified by the set of slant elements, consequently producing the predominant slant feature value. . . .	34
4.1	Hypothetical Gaussian confidence distributions for genuine signatures (G) and forgeries (F), as well as their respective misclassification subsets G^F and F^G . Also indicated is the optimal verification threshold τ^* such that $G^F \cup F^G$ is minimised.	41
4.2	Illustration of the feature vector alignment process utilised by DTW. The algorithm identifies similar features contained within the test and reference vectors. The resulting dissimilarity measure is based on the optimal path obtained between these vectors, as opposed to simply matching corresponding components. For the DTW base classifiers developed in this study, feature vectors extracted from the test and reference patterns have the same dimension d . The internal parameter H_{vec} , referred to as the <i>bandwidth</i> , is used to regulate the algorithm's <i>flexibility</i> , and is discussed in Section A.1.	42
4.3	Examples of popular HMM topologies for $N = 4$. (a)-(b) Ergodic and ring-structured HMMs, respectively. (c) A left-right HMM with $l = 2$ forward links per state. Note that only the left-right model has a designated initial state.	45
5.1	Example of a typical MAROC-curve associated with continuous classifiers C_A and C_B , comprised of $J \approx 1200$ and $K \approx 1200$ discrete classifiers, respectively. Note that, for illustrative purposes, only 0.25% of the discrete classifier combinations obtained from Haker's algorithm are presented in the figure. Also note that the resulting MAROC-curve, obtained by considering all JK possible discrete classifier combinations, is completely specified by approximately 20 ROC-points.	53
6.1	Typical examples of signature images contained in Dolfing's data set. Each image represents a genuine signature belonging to one of the 51 enrolled writers. Note that all the signatures presented have the same uniform stroke width of 5 pixels. This property is not clearly illustrated in the figure, however, since the scale of the images differ.	57

6.2	Schematic representation of the experimental protocol considered in this study. Note that each process utilises a disjoint subset of signature data. Each entity M_i , C_i and C_i^* denotes an untrained, trained and optimised base classifier, respectively. The entities O_{1-N}^* and E_{1-N}^* denote the optimal combined classifier, as obtained from the optimisation set and evaluation set, respectively, whilst M_{1-N}^* denotes the collection of untrained base classifiers associated with the optimal classifier ensemble. Note that n denotes the number of available base classifiers, whilst N denotes the number of base classifiers utilised in the optimal classifier ensemble. Detailed discussions on model optimisation and model evaluation are presented in Sections 6.3.2 and 6.3.3, respectively.	59
6.3	The 3-fold cross-validation procedure considered in this study. Each rectangular segment, within the context of a single fold, represents the signature set belonging to one of the 51 writers contained in Dolfig's data set. As the fold index increases, the optimisation set and evaluation set are shifted 17 writers to the right, thereby considering all 51 writers for evaluation over the 3 folds.	63
6.4	ROC-based performance achieved by the set of DTW base classifiers, using both the optimisation set and the evaluation set.	64
6.5	ROC-based performance achieved by the set of HMM base classifiers, using both the optimisation set and the evaluation set.	66
6.6	ROC-based performance achieved by the set of combined classifiers, using both the optimisation set and the evaluation set.	69
6.7	ROC-based comparison between results achieved by the set of DTW base classifiers, when considering a rigid grid (RG) and a flexible grid (FG) for feature extraction. Results are obtained using the evaluation set.	72
6.8	ROC-based comparison between results obtained for the set of HMM base classifiers, when considering either a rigid grid (RG) or a flexible grid (FG) for feature extraction. Results are obtained using the evaluation set.	75
6.9	ROC-based performance of the combined classifier, using only the two most proficient base classifiers, constructed using Haker's algorithm. Only the optimisation set associated with fold 1 is used. Note the significant difference between predicted and estimated performance - indicating an insufficient degree of independence between the two classifiers submitted for combination.	76
6.10	ROC-based performance of the set of intermediate MAROC combined classifiers constructed during iterations 1-6. Only the optimisation set associated with fold 1 is used. The final iteration is presented in more detail in Figure 6.11.	78
6.11	ROC-based performance of the MAROC combined classifier constructed during the seventh and final iteration. Only the optimisation set associated with fold 1 is used. Note that, similar to iterations 2-6, the combination of a ROC-curve with a MAROC-curve is associated with significantly fewer discrete classifier combinations than Haker's original algorithm. It is this property that renders the MAROC-based combination strategy computationally feasible for a much larger number of continuous classifiers, as discussed in Section 5.3.2.	79

6.12	ROC-based comparison between results obtained for the combined classifier using Haker’s algorithm with the MAROC combined classifier developed in this study. Results are obtained using only the evaluation set associated with fold 1.	80
B.1	Conceptualisation of the observation sequence probability $P(\mathbf{O} \lambda_i)$ as a function of the model configuration λ_i . From this conceptualisation it is clear that the convergence of $P(\mathbf{O} \lambda_i)$, to a local or global maximum $P(\mathbf{O} \bar{\lambda}_i)$, is based solely on the corresponding parameter initialisation λ_i	94

List of Tables

1.1	Summary of results obtained for the set of base classifiers.	17
1.2	Summary of results obtained for the set of combined classifiers.	18
5.1	Maximum likelihood combination of the binary output of classifiers C_A and C_B	51
5.2	Maximum likelihood combination, in terms of the associated TPR and FPR, of the binary output of classifiers C_A and C_B	51
5.3	Combination rules considered for the output of classifiers C_A and C_B	52
5.4	Calculation of the predicted combined TPR and FPR associated with the set of MLE combination rules.	52
5.5	Decision rules considered by the combined classifier, given the base classifier scores s_A and s_B	52
6.1	The number of signatures used in the partitioning of Dolfing’s data set into a separate optimisation set and evaluation set. The specific tasks associated with the subsets T_O , O_B , O_C , T_E and E are discussed in Sections 6.3.2 and 6.3.3.	58
6.2	The set of fixed model hyper-parameter values. These values are fixed prior to model optimisation, thereby greatly decreasing the number of model configurations to consider. Note that reference is made to the section in which each parameter is introduced. Also note that d and T refer to the feature vector dimension and observation sequence length, respectively, as introduced in Section 1.2.1.	60
6.3	The set of flexible grid parameter values considered for model optimisation. Since $F_x = F_y = 0$ is included, the combination of these parameter values results in a total of 4 rigid segmentation grids and 320 flexible segmentation grids considered per base classifier.	60
6.4	Optimal grid configurations obtained for the set of DTW base classifiers. . .	63
6.5	Results obtained for the set of DTW base classifiers.	64
6.6	Generalisation errors obtained for the set of DTW base classifiers, using both the optimisation set and the evaluation set.	65
6.7	Optimal grid configurations obtained for the set of HMM base classifiers. . .	65
6.8	Results obtained for the set of HMM base classifiers.	66
6.9	Generalisation errors obtained for the set of HMM base classifiers, using both the optimisation set and the evaluation set.	67

6.10	Performance-based ranking of the DTW and HMM base classifiers, as obtained during model optimisation. For each base classifier, the classification technique (C), feature extraction technique (F) and resulting AUC (%) performance measure (P) is specified.	67
6.11	Optimal combination levels for the set of combined classifiers.	68
6.12	Results obtained by the set of combined classifiers.	68
6.13	Generalisation errors obtained for the set of combined classifiers, using both the optimisation set and the evaluation set.	69
6.14	Optimal grid configurations obtained for the set of DTW base classifiers, when only rigid grid-based feature extraction is used.	70
6.15	Results obtained for the set of DTW base classifiers, when considering only rigid grid-based feature extraction.	71
6.16	Generalisation errors obtained for the set of DTW base classifiers, when considering only rigid grid-based feature extraction, using both the optimisation set and the evaluation set.	71
6.17	Comparison of the results obtained for the set of DTW base classifiers, when considering either a rigid grid (RG) or a flexible grid (FG) for feature extraction. The μ_{AUC} and μ_{EER} measures are obtained using the evaluation set, whilst μ_{ϵ} is obtained using both the optimisation set and the evaluation set.	72
6.18	Optimal grid configurations obtained for the set of HMM base classifiers, when considering only rigid grid-based feature extraction.	73
6.19	Results obtained for the set of HMM base classifiers, when considering only rigid grid-based feature extraction.	74
6.20	Generalisation errors obtained for the set of HMM base classifiers, when considering only rigid grid-based feature extraction, using both the optimisation set and the evaluation set.	74
6.21	Comparison of the results obtained for the set of HMM base classifiers, when considering either a rigid grid (RG) or a flexible grid (FG) for feature extraction. The μ_{AUC} and μ_{EER} measures are obtained using the evaluation set, whilst μ_{ϵ} is obtained using both the optimisation set and the evaluation set.	74
6.22	Results obtained for the set of intermediate MAROC combined classifiers constructed during iterations 1–7. Only the optimisation set associated with fold 1 is used.	77
6.23	Predicted performance of the set of intermediate MAROC combined classifiers constructed during iterations 1–7. Only the optimisation set associated with fold 1 is used. These predictions assume conditionally independent classifier decisions.	79
6.24	Comparison of the results obtained for the SA and MV combined classifiers developed in this study, after considering amateur forgeries only, with previous systems evaluated using Dolfing’s data set.	81

List of Symbols

This list provides a collection of recurrent symbols used throughout this study, not including notation introduced in Appendices A and B.

Pattern recognition	
\mathbf{X}	Continuous observation sequence
\mathbf{x}	Continuous observation
x_f	Feature of type f
d	Feature vector dimension
\mathbf{O}	Discrete observation sequence
o	Discrete observation
T	Observation sequence length
ω	Pattern class
Ω	Number of pattern classes
C_i	Continuous base classifier i
C_{1-N}	Combined classifier utilising N individual base classifiers

Signature segmentation	
I	Signature image
m	Vertical dimension of signature image
n	Horizontal dimension of signature image
M	Number of rows in segmentation grid
N	Number of columns in segmentation grid
I_{ij}	Signature sub-image extracted by segmentation grid cell (i,j)
F_x	Horizontal flexibility of segmentation grid cell boundaries
F_y	Vertical flexibility of segmentation grid cell boundaries
l_x	Maximum allowable horizontal flexibility of segmentation grid cell boundaries
l_y	Maximum allowable vertical flexibility of segmentation grid cell boundaries

Vector quantisation	
Q	Vector quantiser
\mathbf{V}	Codebook
v_k	Codeword k
R_k	Region in feature space associated with codeword k
ϵ_k	Distortion associated with codeword k
K	Maximum allowable codebook size

Signature modelling	
\mathbf{X}_q	Questioned signature pattern
\mathbf{X}_k	Reference signature pattern
M_ω	Signature model for pattern class ω
K_ω	Number of training samples for pattern class ω
μ_ω	Mean dissimilarity between M_ω and the training set for pattern class ω
σ_ω	Standard deviation of the dissimilarities between M_ω and the training set for pattern class ω

Dynamic time warping	
H_{vec}	Vector alignment bandwidth
$D(\mathbf{X}_i, \mathbf{X}_j)$	Dynamic time warping-based dissimilarity between observation sequences \mathbf{X}_i and \mathbf{X}_j

Hidden Markov models	
λ	Hidden Markov model
N	Number of states
M	Number of symbols
q_t	State at time t
l	Number of allotted forward links per state
π	Initial state distribution
\mathbf{A}	State transition probability distribution
\mathbf{b}	Observation symbol probability distribution
$D(\mathbf{O}, \lambda)$	Dissimilarity between observation sequence \mathbf{O} and hidden Markov model λ

Verification	
δ	Dissimilarity score
s_c	Confidence score
τ	Decision threshold
D	Verification decision
P	Partial verification decision

Performance measures	
$f_i^+(\tau)$	False positive rate associated with discrete classifier $C_i(\tau)$
$t_i^+(\tau)$	True positive rate associated with discrete classifier $C_i(\tau)$
ϵ	Generalisation error

List of Acronyms

AER	Average error rate
AMF	Adaptive median filter
AUC	Area under curve
DRT	Discrete Radon transform
DTW	Dynamic time warping
EER	Equal error rate
ES	Evaluation set
FAR	False acceptance rate
FPR	False positive rate
FRR	False rejection rate
GCD	Gravity centre distance
HMM	Hidden Markov model
MAROC	Maximum attainable ROC
MLE	Maximum likelihood estimation
MV	Majority vote
NN	Neural network
ORT	Orientation
OS	Optimisation set
PD	Pixel density
PDF	Probability density function
PS	Predominant slant
RBF	Radial basis function
RBP	Resilient back-propagation
ROC	Receiver operating characteristic
SA	Score averaging
SDC	Simple distance classifier
SVM	Support vector machine
TPR	True positive rate
VQ	Vector quantisation

Chapter 1

Introduction

“He who seeks for methods without having a definite problem in mind seeks in the most part in vain.”
- David Hilbert (1862–1943)

1.1 Background

The field of automatic signature verification has intrigued researchers the world over during recent decades, as it not only serves as an exciting platform for the development of innovative mathematical modelling techniques, but also holds undeniable economic potential. As a result, signature verification systems have experienced quantum leaps regarding both complexity and efficiency at a continuous and relentless pace.

As the world population continues to increase¹, so too does the potential for ill-intentioned individuals to perpetrate identity fraud. Such efforts are further supported by the relatively recent paradigm shifts regarding point-of-sale payment options. The use of cheques and especially credit cards have quickly become the preferred method of payment for most individuals, particularly in the developed world. Even though this monetary evolution holds obvious benefits, as it all but eradicates the need for individuals to carry large amounts of cash on their person, it is entirely based on the notion that these tokens would be of no use whatsoever to anyone other than the owner, as a transaction cannot be completed without a valid signature.

This is simply not the case, as both cheque and credit card fraud cost financial institutions an unfathomable amount of money on an annual basis. Reports by the American Bankers Association (2007) suggest that annual attempted cheque fraud in the United States increased from \$5.5 billion to \$12.2 billion during the period 2003–2007, whilst actual losses increased from \$677 million to \$969 million during the same period. Also, the Association for Payment Clearing Services (2008) report that during the first semester of 2008, losses due to cheque fraud in the United Kingdom reached £20.4 million, whilst losses due to point-of-sale credit card fraud reached £47.4 million. These levels constitute increases of 35% and 26%, respectively, when compared to the same period in 2007.

¹According to the United States Census Bureau (2008), world population increased from 3 billion to 6 billion during the period 1959–1999, whilst current estimations indicate it will reach 7 billion in 2012.

All the aforementioned factors suggest that effective automatic handwritten signature verification systems are no longer a technological luxury as in years past, but have in fact become a true necessity in the modern document processing environment.

1.2 Key concepts

The successful implementation of an off-line handwritten signature verification system represents a highly specialised amalgamation of various concepts throughout the mathematical sciences. In this section, several of the most important concepts relevant to such an application are discussed.

1.2.1 Pattern recognition

The process of *pattern recognition* constitutes the intelligent foundation of any decision-making process. In order to perform anything from menial tasks to complex data analysis, human beings rely greatly on the ability of the brain to perform pattern recognition on a daily basis. Consider, for example, attempting to drive a vehicle without the ability to recognise and interpret traffic signals. Such real-time pattern recognition processes govern nearly every scenario in modern society.

From a mathematical perspective, pattern recognition involves classifying a *pattern*, represented by an *observation sequence* \mathbf{X} , as belonging to one of Ω finite *pattern classes* $\{\omega_1, \omega_2, \dots, \omega_\Omega\}$. An observation sequence is constructed from a set of T , d -dimensional *feature vectors* $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$, where each element of \mathbf{x}_i denotes a measurement of arbitrary origin, referred to as a *feature*.

The pattern recognition process, as illustrated in Figure 1.1, consists primarily of two phases, namely *feature extraction* and *classification*. In some cases, depending on the nature of the data being modelled and the classification technique utilised, certain preprocessing and/or post-processing of the system data may be required.

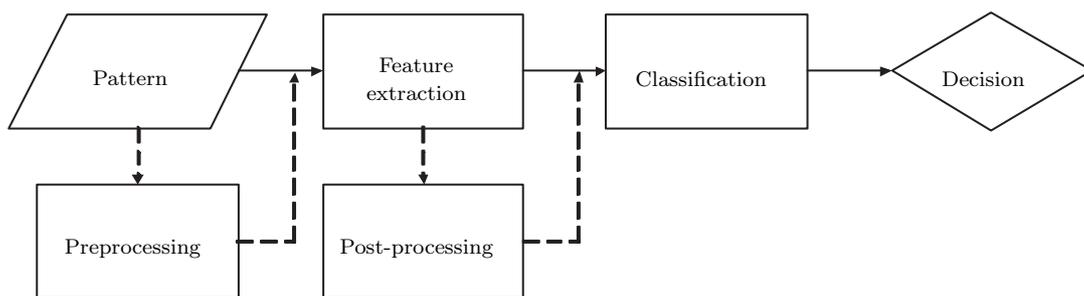


Figure 1.1: The pattern recognition process.

Feature extraction

During the feature extraction phase, the system analyses a given pattern and records certain features, in order to yield structured data in the form of an observation sequence.

Any measurable quantity may constitute a feature. However, since the ultimate aim is to classify a test pattern based solely on such features, it becomes advisable to select a feature set such that patterns belonging to different pattern classes are maximally separated² in the *feature space*. A selection of popular feature types, in the context of signature verification, is categorised in Figure 1.2.

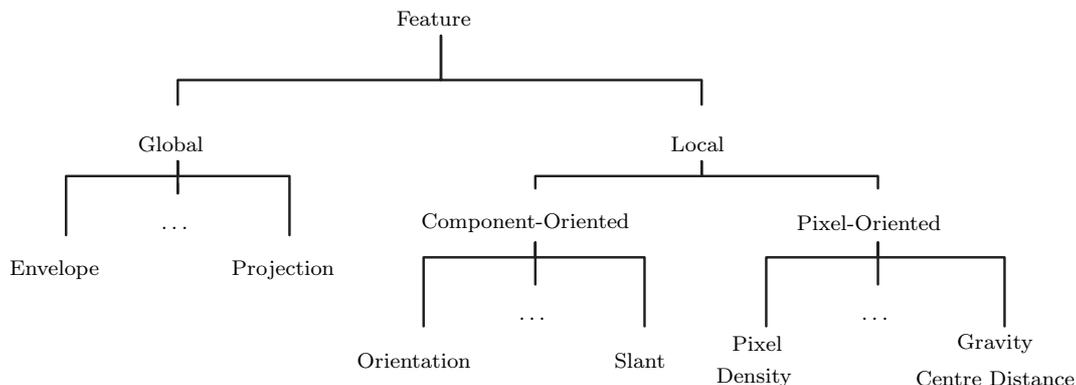


Figure 1.2: Categorisation of popular features associated with off-line signatures.

The base classifiers developed in this study employ a selection of local features, including *pixel density* (PD), *gravity centre distance* (GCD), *orientation* (ORT) and *pre-dominant slant* (PS). These features have been used to great effect in the literature, as each enables signature analysis on either stroke or sub-stroke level, thereby generating robust observation sequences. Furthermore, each base classifier developed in this study employs a novel feature extraction technique, combining the efforts of the aforementioned local features with a flexible grid-based signature segmentation strategy.

Classification

During the classification phase, the feature space is partitioned into Ω disjoint regions, where each region is representative of a pattern class. If the pattern classes represented within the training set are known beforehand, this process is referred to as *supervised learning*. Conversely, during *unsupervised learning*, the system is required to define these pattern classes, prior to delivering a classification result. Subsequently, if an observation sequence \mathbf{X} is yielded by the feature extraction phase and found to be contained within region R_j , the pattern is classified as belonging to pattern class ω_j .

As is the case with feature selection, there exists a wide variety of classification techniques available for incorporation into a successful signature verification system. Selected examples of such techniques are categorised in Figure 1.3.

The base classifiers developed in this study construct writer models using two fundamentally different classification techniques, namely the *dynamic time warping* (DTW)

²In an optimal scenario, each pattern class would occupy a compact and disjoint region.

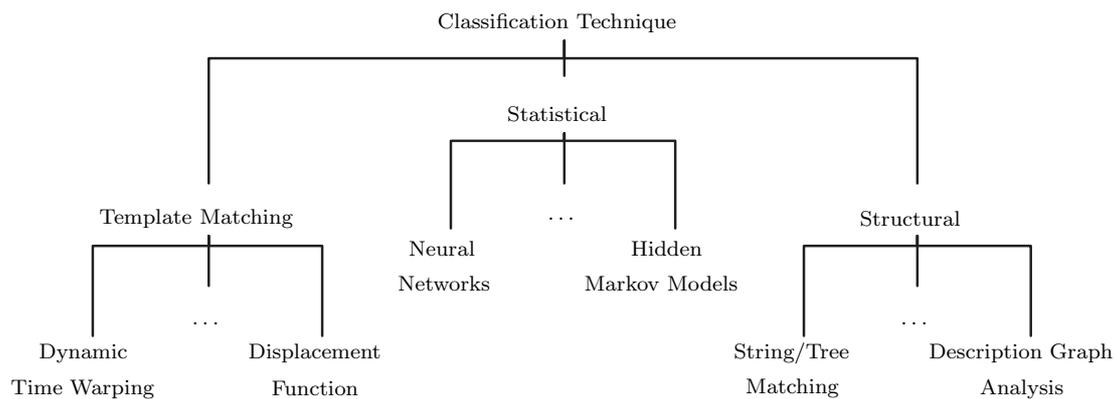


Figure 1.3: Categorisation of popular classification techniques.

algorithm and the discrete observation *hidden Markov model* (HMM). DTW is a commonly used template matching technique, whilst the HMM represents a powerful generative model. Both the DTW and HMM base classifiers are trained, given a set of genuine signature patterns per writer, using supervised learning.

1.2.2 Combined classifiers

In general, a pattern recognition system is constructed by utilising one or more feature extraction techniques, in conjunction with a single classification technique. The use of several feature extraction techniques is recommended, as this ensures greater separation of different pattern classes in the feature space.

Given a set of two or more classifiers, referred to as a *classifier ensemble*, it is logical to expect an improvement in performance when combining the separate efforts of each into a single classifier, referred to as a *combined classifier*. The combination process is performed either on *score level* or *decision level*.

In this study, we consider both score fusion and decision fusion, in order to develop fundamentally different combined classifiers. These combined classifiers utilise the efforts of the set of HMM and DTW base classifiers developed in this study.

1.2.3 Automatic identification systems

The process of manual signature verification is a laborious one. This is especially true in the commercial and financial sectors, where vast quantities of cheques and other official documents are processed on a daily basis. Furthermore, advances achieved in the computer industry over the past few decades have not only introduced computer systems to a wide range of previously unconsidered locations, but have also rendered the use of computer-based systems for identity verification a computationally and economically viable option.

A well-developed automatic identification system holds two cardinal advantages over human verifiers, namely efficiency and accuracy. According to Coetzer *et al.* (2006), where human and machine performance is directly compared within the context of off-

line signature verification, a bank clerk is likely to take 3–5 seconds in verifying the authenticity of a signed cheque. For this reason, only cheques valued over a certain threshold are usually submitted for manual verification. In contrast, assuming an efficient design and implementation, one may typically expect a machine to perform the same task in a matter of milliseconds, once a suitable digital representation of the signature is acquired. This increased efficiency therefore allows the processing of a much greater number of cheques, such that cheques of a significantly increased range of values can be submitted for verification. In addition, Coetzer *et al.* report that the probability that a human verifier will outperform their HMM-based verification system, 1–4 times in 22 trials, is 0.22%. This statistic undeniably confirms the enormous potential associated with deploying a machine-based handwritten signature verification system, either as an alternative to manual verification or simply as a reliable aid.

In general, automatic identification systems are categorised as being either *knowledge-based*, *possession-based* or *biometric*, as illustrated in Figure 1.4.

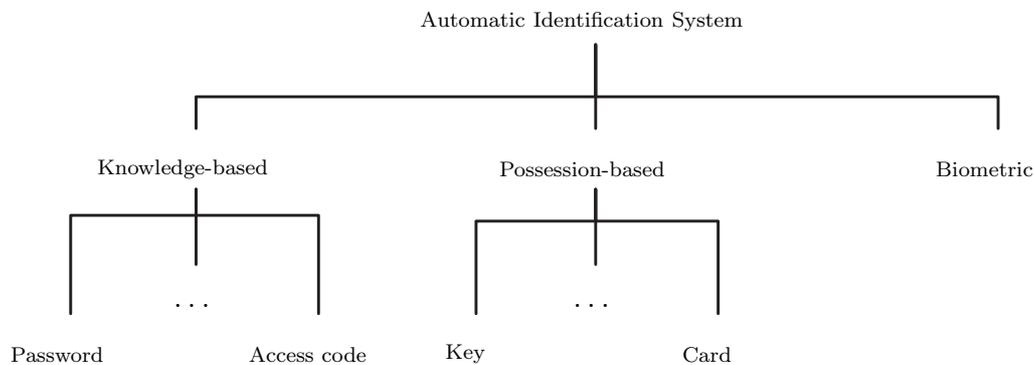


Figure 1.4: Categorisation of automatic identification systems.

Knowledge-based identification systems require an individual to produce some form of information, usually a password or access code, for verification purposes. As we are constantly reminded, though, we currently live in the *information age*, where entities such as the internet provide human beings with constant, and potentially unrestricted, access to practically any information desired. This concept greatly diminishes the level of security offered by a knowledge-based identification system.

Possession-based identification systems attempt to eradicate this defect by requiring an individual to produce a physical token, such as a key or card. Such tokens may of course be lost or stolen, thereby nullifying the security provided by the associated possession-based identification system.

Biometric identification systems generally avoid both the aforementioned pitfalls by performing *ad hoc* verification on the basis of a physiological or behavioural attribute unique to the person in question, thereby strictly requiring the presence of an authorised individual. Such systems are discussed further in the next section.

1.2.4 Biometric authentication

The use of handwritten signatures as a means of identity verification constitutes a subclass of what is known as *biometric authentication*. The success of biometric authentication relies on the belief that it is significantly more difficult to mimic a physiological or behavioural human trait than, for example, to obtain a key or uncover a password. A selection of popular biometric authentication systems is categorised in Figure 1.5.

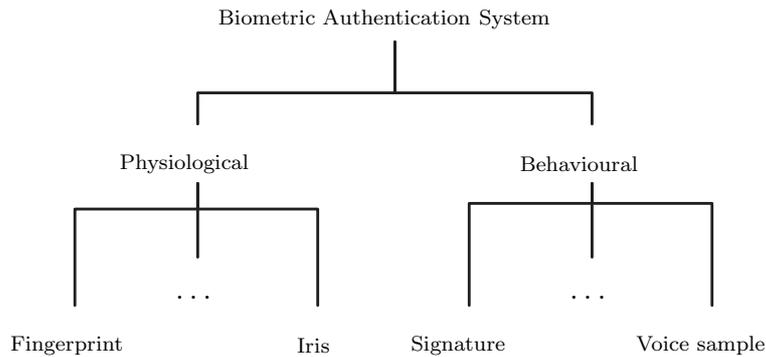


Figure 1.5: Categorisation of popular biometric authentication systems.

Although biometric systems that are based on physiological traits (such as a face, iris or fingerprint) provide a much greater level of accuracy than those based on behavioural traits (such as a handwritten signature or voice sample), the implementation of a physiological biometric authentication system is often economically or computationally infeasible. In addition, due to the invasive nature of a physiological system, use on the general population is often considered inappropriate. As a result, the deployment of such sophisticated systems is usually reserved for high-level security applications.

1.2.5 Handwritten signatures

Handwritten signatures, henceforth referred to only as signatures, have been considered valid proof of identity and consent for centuries. The signing of the US Declaration of Independence, presented as Figure 1.6, is epitomic of this social credence. Even in our present day and age, dominated by advanced technological systems and protocols, signatures remain the preferred method for identity verification, as they are both non-intrusive and easily collectable.

According to Schmidt (1994), an individual's signature is usually composed of stroke sequences much unlike those used in ordinary handwriting and, in addition, tends to evolve towards a single, unique design. This is not only as a result of repetition³, but also the innate desire of each person to create a unique signature. Signatures are therefore able to reflect a writer's subtle idiosyncrasies to a much greater extent than ordinary handwriting.

³Sustained repetition of a physical activity leads to the development of so-called *muscle memories*, ensuring improved consistency in the case of signatures.

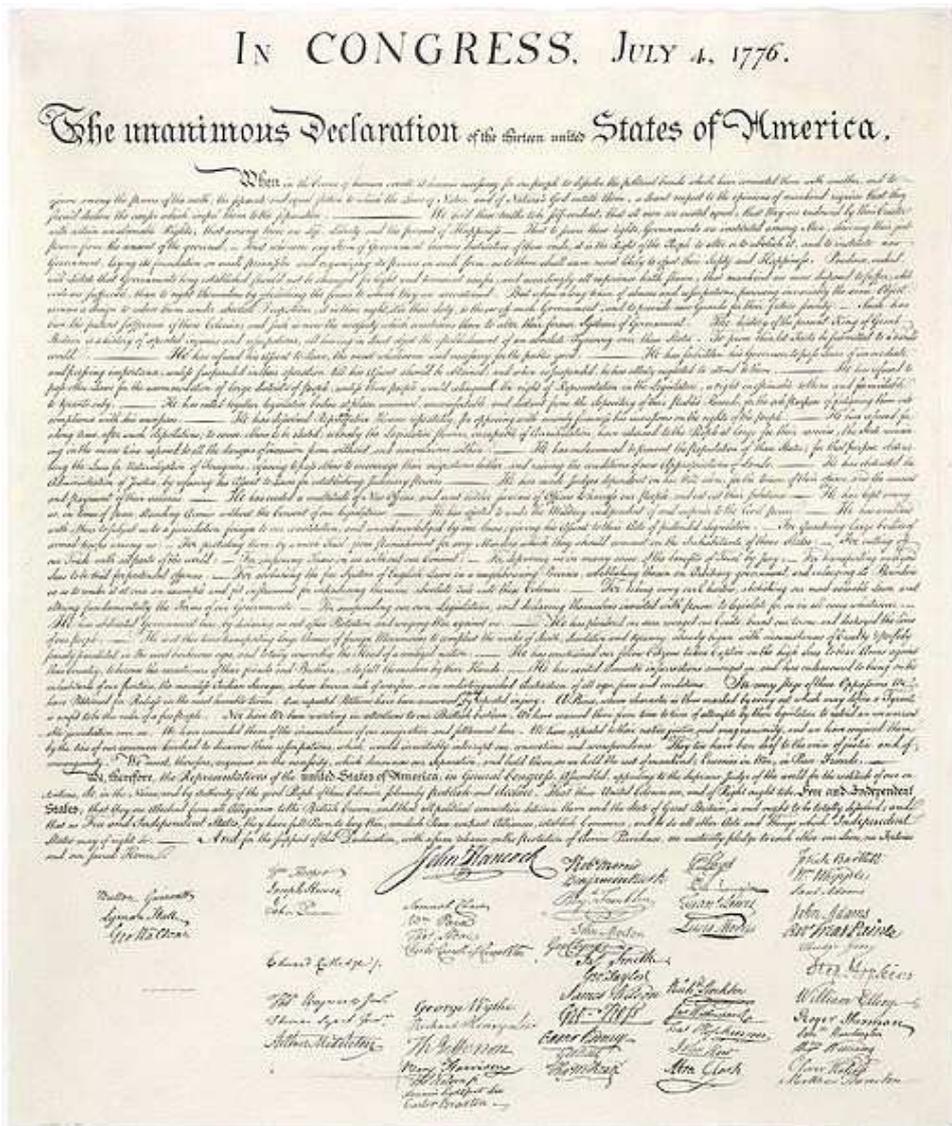


Figure 1.6: The Declaration of Independence of the United States of America, as signed by 56 delegates of US Congress on July 4th, 1776.

1.2.6 Recognition and verification

A clear distinction should be made between systems developed for signature *recognition* and those intended for signature *verification*.

A recognition system receives as input a signature of unknown origin. The system then has to determine to which one of its finite number of enrolled writer classes the input signature is a closest match. A Ω -class recognition system therefore needs to compare its input to samples representative of each of its Ω writer classes before delivering a probabilistic output as to the origin of the input signature.

A verification system, on the other hand, receives as input a signature of unknown origin, but also a claim of ownership. The system then has to either confirm or reject

the validity of this claim. In order to achieve this, the system compares the signature to samples of the claimed owner, before delivering an output as to its certainty concerning the validity of the claim of ownership.

Verification systems may therefore be viewed as a specific subclass of recognition systems, namely bi-class recognition systems that classify input as belonging to either the positive (genuine) or negative (forgery) class. As a result, there exists certain terminology that is used to describe either system indiscriminately. During the course of this study, for example, there are references to classifiers and classes, whilst it is implied that reference is being made to verifiers and genuine/forged signatures, respectively.

1.2.7 Writer-dependent and writer-independent verification

In a writer-dependent verification scenario, there exists a unique, trained model M_ω for each writer ω enrolled into the system database. When the system receives a questioned signature pattern \mathbf{X} and claim of ownership ω , the pattern is matched with M_ω , subsequently yielding a score reflecting the (dis)similarity between \mathbf{X} and a typical signature pattern used to train M_ω . It should be made clear, however, that a *global* decision threshold τ , as discussed in the next section, is used for verification purposes.

The writer-independent approach, on the other hand, performs verification using a single model M , regardless of the number of writers enrolled in the system database. This is achieved by attempting to model the difference between genuine signatures and forgeries in general. Any classifier employing the writer-independent approach is therefore trained using a set of modified feature vectors, known as *difference vectors*. In order to construct such difference vectors, each writer ω provides a genuine signature pattern $\mathbf{X}_k^{(\omega)}$ as reference. Any pattern $\mathbf{X}^{(\omega)}$ belonging to or claimed to belong to writer ω , subsequently presented to the system, is converted to the difference vector $\mathbf{Z}^{(\omega)}$ by computing

$$\mathbf{Z}^{(\omega)} = D(\mathbf{X}_k^{(\omega)}, \mathbf{X}^{(\omega)}), \quad (1.1)$$

where $D(\cdot)$ denotes any suitable distance measure.

In order to effectively model the difference between genuine signatures and forgeries by using the writer-independent approach, though, one typically requires the efforts of a discriminative classifier such as a neural network (NN) or support vector machine (SVM), as both genuine signatures and forgeries are used during model training. For this reason, the DTW and HMM base classifiers developed in this study employ a writer-dependent modelling strategy.

1.2.8 Performance measures

During the course of this study, the performance measures *false rejection rate* (FRR), *false acceptance rate* (FAR), *average error rate* (AER), *equal error rate* (EER), *true positive rate* (TPR), *false positive rate* (FPR) and *area-under-curve* (AUC) are considered.

In order to define these measures, it is first necessary to define the set of possible classification events. A *false positive* event occurs when a forgery, or *negative instance*, is misclassified as belonging to the positive class, whilst a *false negative* event occurs when a genuine signature, or *positive instance*, is misclassified as belonging to the negative class.

Similarly, *true positive* and *true negative* events are indicative of the correct classification of genuine signatures and forgeries, respectively. In an experimental scenario, we denote the number of instances delivering the outcomes false positive, false negative, true positive and true negative by F^+ , F^- , T^+ and T^- , respectively. Furthermore, we denote the number of genuine test signatures by n^+ , whilst the number of forged test signatures are denoted by n^- .

The FRR, or *Type I error*, refers to the number of false negatives in relation to the number of genuine test signatures, or

$$\text{FRR} = \frac{F^-}{n^+}, \quad (1.2)$$

whilst the FAR, or *Type II error*, refers to the number of false positives in relation to the number of forged test signatures, or

$$\text{FAR} = \frac{F^+}{n^-}. \quad (1.3)$$

The AER simply refers to the average of the FRR and FAR. For *continuous*⁴ classifiers, both the FRR and FAR may be manipulated by adjusting a global decision threshold τ (see Section 4.2.2). As the FRR is decreased, the FAR increases, and vice versa. It is therefore logical to expect that, for a certain τ -value, the FRR and FAR will coincide, as illustrated in Figure 1.7 (a). This value, known as the EER, is a commonly used quality performance measure throughout the literature.

Another platform used to gauge system performance, which has gained considerable popularity during recent years, is the *receiver operating characteristic* (ROC) curve. A ROC-curve is obtained by plotting the TPR, defined as

$$\begin{aligned} \text{TPR} &= \frac{T^+}{n^+} \\ &= 1 - \text{FRR}, \end{aligned} \quad (1.4)$$

against the FPR, for all values of τ . The FPR is synonymous to the FAR. Each point in ROC-space, denoted by $(f_i^+(\tau), t_i^+(\tau))$, therefore represents the FPR-TPR pair associated with a τ -specific discrete classifier $C_i(\tau)$. It should be clear that one may also obtain the EER associated with a continuous classifier from its corresponding ROC-curve, as illustrated in Figure 1.7 (b).

One of the fundamental ROC-based performance measures associated with a continuous classifier is its corresponding AUC. This measure is defined as the area spanned by the convex hull of each point on the ROC-curve and the ROC-point (1,0). The AUC associated with a continuous classifier may be interpreted as the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance.

Therefore, by employing the EER and AUC as performance measures, an ROC-curve enables one to graphically represent a system's typical discriminative ability, whilst simultaneously being capable of illustrating overall system stability. Furthermore, it is

⁴A classifier is said to be continuous if it produces a probabilistic estimate regarding a test pattern's class membership, to which different thresholds may be applied, in order to assign a class label. A discrete classifier, in contrast, produces a predicted class label only, and is associated with a single threshold.

very convenient to represent performance as a function of the FPR, since it is often desirable to predefine a maximum allowable FPR, especially in cost-sensitive scenarios. For a comprehensive overview regarding ROC analysis, the reader is referred to Fawcett (2006). The various quality performance measures considered in this study are illustrated in Figure 1.7.

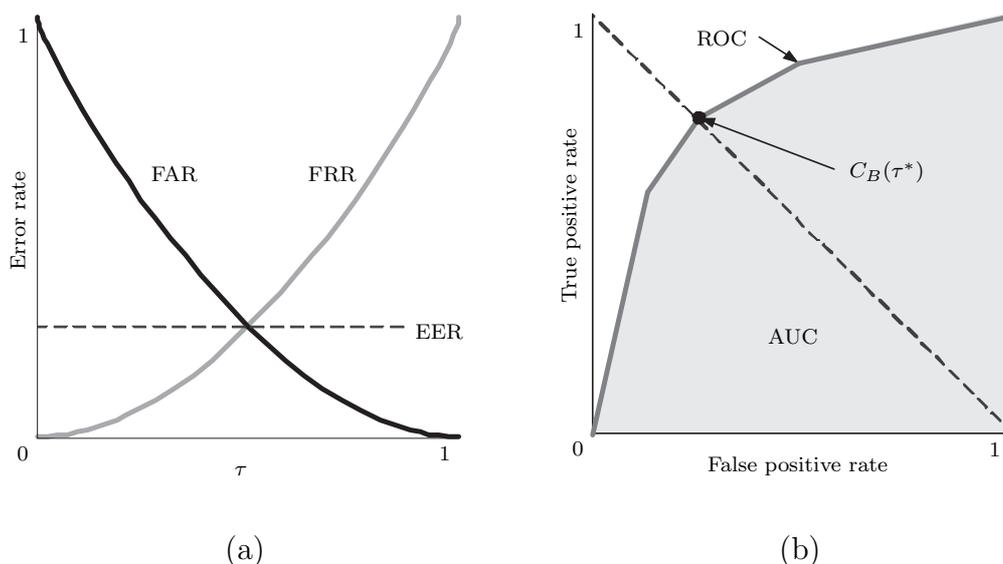


Figure 1.7: Hypothetical representations of the performance measures associated with two continuous classifiers C_A and C_B . (a) The FRR, FAR and EER. (b) The ROC-curve and AUC measure, as well as the EER-based optimal discrete classifier $C_B(\tau^*)$.

1.2.9 On-line and off-line signatures

The field of automatic signature verification may currently be divided into two distinct sub-categories, namely those systems concerned with *on-line* signature verification and those concerned with *off-line* signature verification.

In the on-line scenario, signature data is captured in real time by means of an electronic pen and digitising tablet, yielding not only pen stroke coordinates, but also dynamic signature data such as pen pressure, velocity and acceleration. On-line signatures are therefore also commonly referred to as *dynamic* signatures.

In the off-line scenario, ink-signed documents require digitisation by means of a scanning device. The obtained signature image therefore only provides the coordinates of pixels representative of pen strokes. During the course of this study, it is assumed that all signature images are in binary format. All pen stroke pixels are therefore represented by 1, whilst a pixel value of 0 denotes the image background. Various static and pseudo-dynamic features may subsequently be extracted from the obtained image. For this reason, off-line signatures are also referred to as *static* signatures.

Apart from the fact that a static signature yields considerably less information than its dynamic counterpart, it may also suffer from the presence of background noise generated during the digitising process. A greater degree of variability also exists with regard to the physical structure of a static signature, as the effect of using different writing instruments and surfaces becomes apparent.

Due to the nature of static features and the adverse effect of background noise, on-line verification systems are, in general, a great deal more reliable than off-line systems. The incorporation of intelligent image processing and feature extraction techniques, as well as robust classification models, are therefore key to the success of any off-line verification system.

1.2.10 Forgery types

In the context of off-line signatures, forgeries may generally be categorised as either *random*, *simple* or *skilled*, in increasing order of quality. Furthermore, skilled forgeries may be sub-categorised as either *amateur* or *professional*, as illustrated in Figure 1.8.

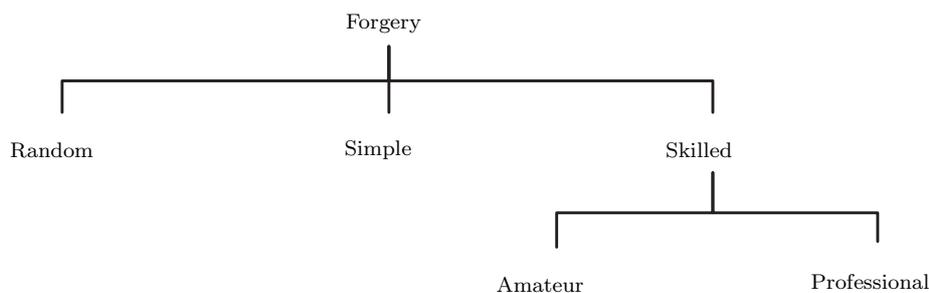


Figure 1.8: Categorisation of several off-line forgery types, increasing in quality from left to right.

In this section we discuss the key requirements for forgery categorisation. Each discussion also provides a typical example of when such a forgery type may be encountered in practise, within the context of cheque fraud. Graphical examples of selected forgery types are provided in Figure 1.9.

Random forgeries

Random forgeries encompass any arbitrary attempt at forging a signature, generally without prior knowledge of the owner's name. This type of forgery may constitute random pen strokes and is usually easy to detect. For experimental purposes, genuine signatures from writers other than the legitimate owner are commonly used to represent random forgeries.

A random forgery is typically expected when a cheque book is registered to a company or institution, rather than a specific individual. The forger therefore has no information regarding the name of an authorised signer. This impediment, however, usually applies

to the cheque's recipient as well, as the unauthorised signing is only detected upon submission to the appropriate banking institution.

This type of forgery is, however, not limited only to cheques withholding personal information. In some instances, random forgeries are produced by *casual* criminals who, unbelievable as it may seem, simply do not go through the effort of looking at the owner's name.

Simple forgeries

In the case of simple forgeries, the forger's knowledge is restricted to the name of the signature's owner. Due to the arbitrary nature of signature design, simple forgeries may in some cases bear an alarming resemblance to the writer's genuine signature. In such cases, more sophisticated systems, able of detecting subtle stylistic differences, are required in order to distinguish between genuine signatures and forgeries of this type.

Simple forgeries usually result from forging a cheque, lost or stolen, registered to an unknown individual. As the name of the legitimate owner is printed on the cheque itself, an effort can be made to produce a realistically expected representation of the genuine signature. No writer stylistic information can be incorporated, though. This type of forgery is generally associated with a brief period of forged cheques, each with a relatively small value. This is due to the fact that a simple forger generally attempts to avoid the attention associated with processing exceedingly large cheques or the usage of a cheque book reported as lost/stolen.

Skilled forgeries

In some instances, the forger is not only familiar with the writer's name, but also has access to samples of genuine signatures. Given ample time to practice signature reproduction, he is able to produce so-called skilled forgeries.

The vast majority of skilled forgeries may be categorised as amateur, as this type of forgery may be produced by any given individual. In contrast, to produce a professional skilled forgery, the forger typically requires a certain amount of knowledge regarding forensic document analysis. This enables the forger to mimic subtle writer-specific idiosyncrasies, thereby producing a forgery far beyond the capabilities of the average individual.

Skilled forgeries are undoubtedly the most difficult to detect, especially by untrained humans. As the production of a skilled forgery involves both planning and effort, similar effort is required to enforce sufficient countermeasures - typically a sophisticated automatic signature verification system. The ability to produce skilled forgeries constitutes the greatest threat to legitimate cheque processing, as an unacceptable number of forged cheques go undetected. Furthermore, the involvement of professional skilled forgers may facilitate large-scale corporate fraud, potentially causing crippling losses to high-profile businesses.

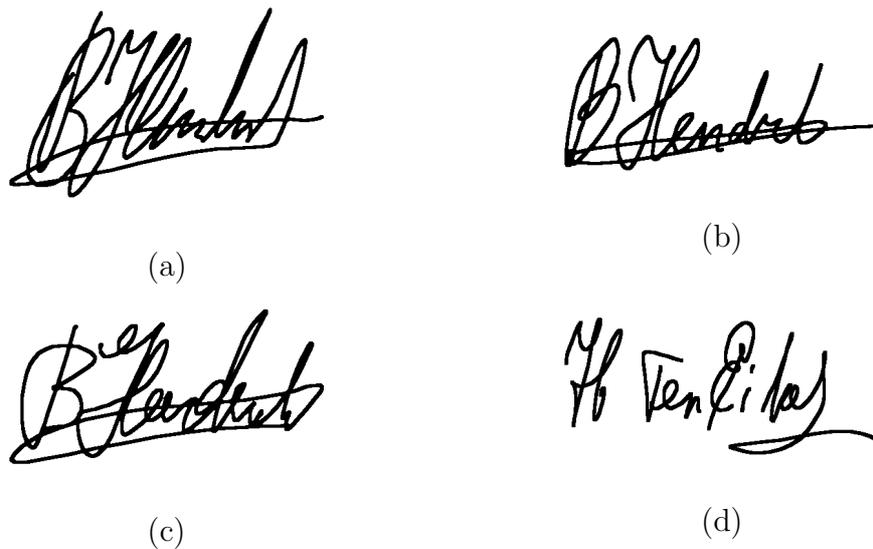


Figure 1.9: Typical examples of (a) a genuine signature, as well as (b) professional skilled, (c) amateur skilled and (d) random forgeries.

1.3 Objectives

During the course of this study, we aim to achieve two primary objectives, namely the successful design and implementation of:

- a novel feature extraction technique, utilising the flexible grid segmentation strategy proposed in this study.
- a robust off-line signature verification system, utilising the efforts of either a score-based or decision-based combined classifier. This combined classifier is to be constructed from an ensemble of DTW and HMM base classifiers.

Furthermore, we investigate the feasibility and significance of a novel classifier ensemble combination strategy proposed in this study. This strategy performs ROC-based combination of an ensemble of continuous classifiers by utilising an existing classifier combination algorithm that is designed for continuous classifier pairs only.

1.4 System overview

In this study we combine an ensemble of continuous base classifiers, in order to obtain a superior combined classifier. Each base classifier utilises a different type of feature, as well as a different modelling strategy.

This section provides a condensed review of the DTW and HMM base classifiers developed in this study, as well as the strategies employed to combine said base classifiers. The general schematics of such a combined classifier is provided in Figure 1.10.

Each base classifier, as illustrated in Figure 1.11, provides fundamentally different capabilities regarding signature analysis, thereby facilitating greatly superior ensemble performance.

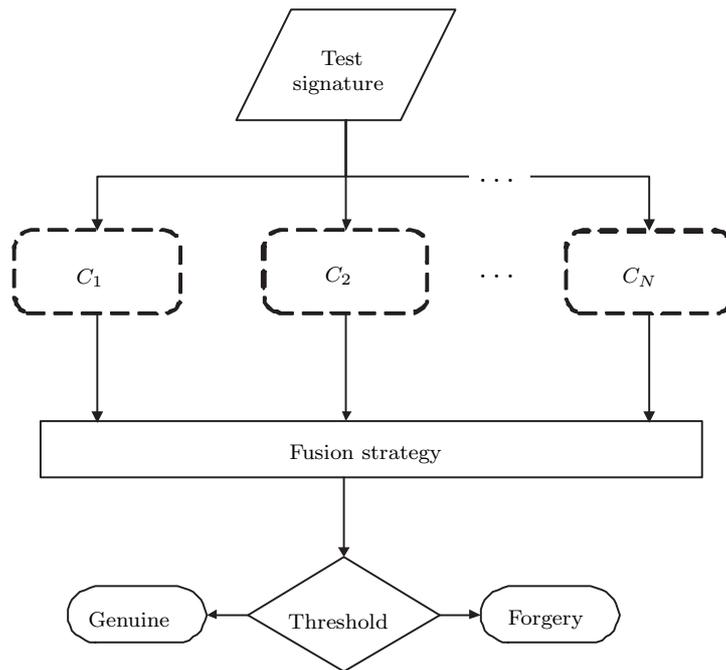


Figure 1.10: Schematic representation of a combined classifier ensemble as developed in this study. Each entity C_i represents a separate base classifier, as illustrated in Figure 1.11.

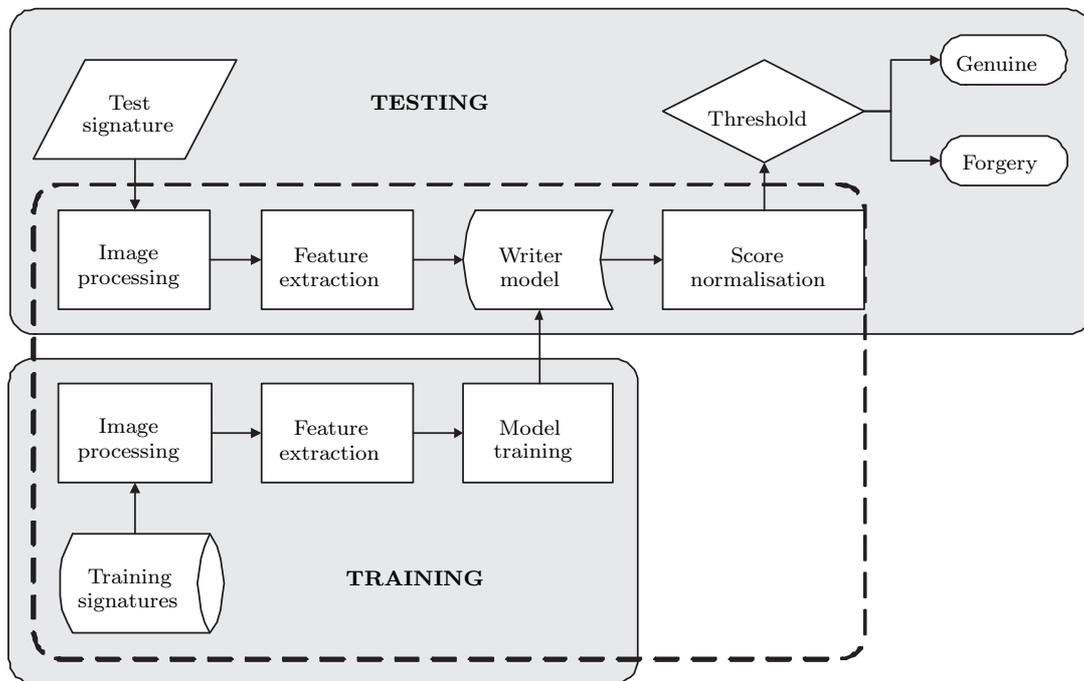


Figure 1.11: Schematic representation of a base classifier as developed in this study. The writer model entity therefore represents either an HMM or a signature template for DTW.

1.4.1 System design

The DTW and HMM base classifiers employ the same basic approach to the pattern recognition process, differing only in their signature matching techniques. Unless stated otherwise, the topics reviewed in this section may therefore be viewed as part of either a DTW-based or HMM-based approach to signature verification. As a result, reference is often made to “the base classifiers”, which may denote either DTW or HMM base classifiers.

Image processing

In order to achieve efficient signature modelling for each writer enrolled into the system, certain image preprocessing is required. Grid-based image segmentation is performed on each signature image, consequently yielding suitable input for the feature extraction process. Two segmentation strategies are considered, namely traditional rigid grid segmentation, as found in the literature, as well as flexible grid segmentation, a novel strategy proposed in this study. Both of these segmentation strategies ensure feature vector representations that are invariant with respect to translation and scale.

Feature extraction

The base classifiers consider the same set of grid-based features for signature modelling. These features include pixel density, gravity centre distance, orientation and predominant slant. By employing grid-based feature extraction techniques, complete and robust feature-specific profiles are created in \mathfrak{R}^d for each signature pattern presented to a base classifier.

In addition, since the HMM base classifiers are designed for discrete observation sequences only, *vector quantisation* (VQ) is performed on the feature set during post-processing, by means of the K -means clustering algorithm.

Signature modelling

The base classifiers consider two fundamentally different approaches to modelling a writer’s signature.

The DTW base classifiers construct writer-dependent models based on template matching techniques. As a result, each writer is modelled using the single observation sequence found to be the most representative during training. Writer-dependent models constructed using the DTW-based approach are well equipped to compensate for intra-class variability, as feature vectors are non-linearly aligned prior to matching.

The HMM base classifiers adopt a stochastic approach to signature modelling, constructing writer-dependent models on the basis of minimum distance statistics. Each writer is modelled using a left-right discrete HMM. Writer-dependent models constructed using an HMM-based approach generally possess a discriminative ability which is superior to their DTW-based counterparts, as the relationships between consecutive observations within a sequences are also modelled.

All signature models constructed in this study include a set of training statistics based on the mean and standard deviation of classifier scores observed during model training.

These statistics describe the level of variability present in each writer's feature profile and play a critical role during score normalisation.

Verification

The base classifiers are aimed at the detection of amateur skilled forgeries. When a questioned signature, along with a claim of ownership, is submitted for verification, a base classifier matches said signature to the model trained for the claimed owner. This process provides a measure of dissimilarity between the test signature and a typical genuine signature used to train the writer-dependent model.

The DTW base classifiers calculate dissimilarity by computing the average distance between the set of non-linearly aligned feature vectors belonging to the questioned signature and the reference signature for the claimed owner.

The HMM base classifiers match a questioned signature to a writer model by means of Viterbi alignment, consequently yielding a probabilistic measure of ownership. By taking the negative log-likelihood of this probability, a dissimilarity measure is obtained.

The base classifiers subsequently convert the obtained dissimilarity measure into a confidence score, by using a sigmoidal score normalisation function. This normalisation technique utilises the writer statistics determined during model training.

Finally, a global decision threshold is imposed. If and only if the confidence score obtained for a questioned signature is equal to or greater than the required threshold value, the claim of ownership is accepted.

Classifier combination

In order to combine the classifier ensemble constructed from these base classifiers, several classifier combination strategies are considered.

One score fusion technique, namely *score averaging* (SA), is investigated. The efficiency of this method is greatly increased by the sigmoidal score normalisation function utilised in this study. Two decision fusion techniques are also investigated, namely the popular *majority vote* (MV) rule and a novel elitist maximum attainable ROC (MAROC) classifier ensemble combination strategy.

In constructing a classifier ensemble, any number of base classifiers may be utilised, regardless of their feature extraction or signature modelling techniques. The optimal ensemble composition is determined experimentally.

Performance evaluation

The success of classifiers developed in this study is evaluated using two fundamentally different performance measures, namely the AUC and EER.

The AUC is used as primary performance measure, as it represents an accurate measure regarding overall performance of a continuous classifier. During experimentation, one classifier is said to outperform another if it yields a greater AUC-value.

The reasoning behind utilising the EER as secondary performance measure is two-fold. Firstly, the EER is used to rank classifiers possessing equal AUC measures. Secondly, as the EER is currently the most common indication of system performance found in the

literature, it enables us to place the performance of the systems developed in this study into a familiar context.

An important issue which is addressed during experimentation is that of *over-fitting*, which is averted by using separate subsets of signature data for model training, optimisation and evaluation. This data partitioning, used in conjunction with a *k*-fold *cross-validation* protocol, greatly increases the credibility of the reported results, as each classifier’s generalisation potential is also scrutinised.

1.4.2 Data

The classifiers developed in this study are optimised and evaluated using the signature database henceforth referred to as *Dolfing’s data set*. This ideal⁵ data set, containing approximately 4800 signatures collected from 51 different writers, is composed of on-line signature data, originally used by Dolfing (1998), and subsequently converted into a suitable off-line representation by Coetzer (2005).

Since the system developed by Coetzer *et al.* has previously been evaluated using this data set, it also provides a relevant benchmark for the performance of the classifiers developed in this study. Dolfing’s data set is discussed in Section 6.2.

1.4.3 Results

The performance achieved by the base classifiers and combined classifiers developed in this study, when assessed using *Dolfing’s evaluation set* (see Section 6.3.1), is summarised in Tables 1.1 and 1.2, respectively. Performance is measured using the AUC and EER, as well as the *generalisation error* ϵ (see Section 6.3.3).

Performance	DTW				HMM			
	PD	GCD	ORT	PS	PD	GCD	ORT	PS
AUC (%)	89.90	89.98	89.62	92.35	92.81	90.07	90.40	90.85
EER (%)	18.30	18.14	18.55	14.24	14.43	17.82	16.21	16.73
ϵ (%)	1.22	1.04	1.77	0.96	0.74	-0.08	1.06	1.50

Table 1.1: Summary of results obtained for the set of base classifiers.

The DTW (HMM) base classifier utilising the PS (PD) feature significantly outperforms its peers. Furthermore, the set of HMM base classifiers generally outperforms the set of DTW base classifiers, both in terms of verification proficiency and generalisation potential.

The majority vote combined classifier slightly outperforms its peers. Furthermore, the set of combined classifiers significantly outperforms the set of base classifiers.

⁵Dolfing’s data set is considered ideal, as it was originally captured on-line. Each signature image is therefore free of background noise, whilst also possessing uniform stroke width. Furthermore, each writer’s set of training signatures share a similar baseline orientation.

Performance	SA	MV	MAROC
AUC (%)	95.06	95.80	94.04
EER (%)	11.21	10.23	12.54
ϵ (%)	1.15	1.05	3.65

Table 1.2: Summary of results obtained for the set of combined classifiers.

1.5 Contribution of this study

During the course of this study, a collection of novel concepts and techniques are developed. Each of these techniques constitutes an extension of the current state of the art, thereby providing a contribution to the field of off-line signature verification. We experimentally verify the contribution made by each presented technique in Section 6.5.

1.5.1 A novel feature extraction technique

The use of grid-based feature extraction techniques has proved very popular during recent years, as the extraction of local features allows signature analysis on a stroke and sub-stroke level.

In this study we propose an extension of the current fixed-resolution rigid grid-based feature extraction technique, which we refer to as the flexible grid-based feature extraction technique. In this strategy, after constructing a traditional rigid segmentation grid, each grid cell boundary is dilated by a predefined factor, thereby allowing adjacent grid cells to overlap. In this manner, the flexible grid-based feature extraction technique not only allows stroke and sub-stroke signature analysis, but also inherently provides information regarding signature progression on a global scale.

This simulated time-evolution, not sufficiently provided by the rigid grid-based approach, greatly increases the robustness of modelling techniques such as DTW, where no information regarding signature progression is incorporated into model training. The improvement achieved when using an HMM base classifier is less significant, as the issue of time-evolution is sufficiently addressed during model training. Nevertheless, a consistent improvement is observed.

1.5.2 A novel classifier ensemble combination strategy

The set of three combined classifiers developed in this study utilise both score-based and decision-based fusion strategies. The SA and MV combined classifiers represent popular fusion techniques and are well documented throughout the literature. The third, an elitist MAROC-based classifier ensemble combination strategy, although based on an existing ROC-based combination strategy, is yet to be reported in the literature. To the best of our knowledge, this strategy may therefore be considered novel.

The ROC-based combination of a continuous classifier pair, originally proposed by Haker *et al.* (2005), combines every threshold-specific discrete classifier contained in one continuous classifier with every threshold-specific discrete classifier contained in the other. Each discrete classifier pair is combined by either adopting one of their decisions exclu-

sively, or by using the well-known AND or OR rule. The appropriate combination scheme is determined by maximum likelihood analysis of each discrete classifier's ROC-statistics. A single continuous classifier can then be selected from the resulting pool of discrete classifier combinations. This combined classifier performs optimally when constructed from two continuous classifiers producing conditionally independent decisions, since this assumption is made during maximum likelihood estimation, which consequently determines the optimal classifier combination rule.

Although the ROC-based strategy is well suited for the pairwise combination of continuous classifiers, the computational cost involved renders it infeasible for the combination of continuous classifier ensembles. In this study we propose the iterative combination of a classifier ensemble, using a similar approach. During the first iteration, the ROC-based strategy is used to combine the first pair of continuous classifiers. Each subsequent iteration, however, combines an additional continuous classifier with the MAROC-based representation of the previous iteration, thereby significantly reducing the computational cost. Using this MAROC-based strategy, the set of continuous classifiers contained in the ensemble are combined from least proficient to most proficient, thereby minimising the amount of potentially valuable discrete classifiers discarded during the combination process. In addition, unlike the SA or MV fusion strategies, the MAROC-based combination strategy is insensitive to the inclusion of a relatively inaccurate base classifier into the ensemble.

1.5.3 A novel off-line signature verification system

By employing the flexible grid-based feature extraction technique proposed in this study, each classifier developed in this study may be considered as novel.

Furthermore, we compare the performance of the optimal combined classifier developed in this study to the systems developed by Dolfing (1998) and Coetzer *et al.* (2004), wherein EERs of 13.3% and 12.2% are reported, respectively. Since both these systems were evaluated using amateur skilled forgeries from Dolfing's data set, they are deemed suitable for comparison. In this study, the MV combined classifier is found to be most proficient, yielding an EER of 10.23%, thereby constituting a credible improvement.

1.6 Thesis outline

Chapter 2: Literature Study discusses selected previous works pertaining to off-line signature verification, thereby providing the reader with a contextual perspective regarding the range of available techniques and corresponding levels of success achieved.

Chapter 3: Image Processing and Feature Extraction discusses how the base classifiers developed in this study convert raw signature images into robust feature vector representations by means of intelligent grid-based segmentation strategies and local feature extraction techniques. Also discussed is the process of vector quantisation.

Chapter 4: Signature Modelling and Verification explains how each of the base classifiers developed in this study constructs a writer-dependent signature model, as well as the methodology considered for subsequent verification of an unknown signature.

Chapter 5: Classifier Combination discusses several strategies, as found in the liter-

ature, for combining the efforts of the previously developed base classifiers. In addition, a novel classifier ensemble strategy, as proposed in this study, is introduced. By utilising any of these combination strategies, a superior combined classifier is obtained.

Chapter 6: Experiments discusses the data and experimental protocol considered during training, optimisation and evaluation of the classifiers developed in this study. Results yielded by the collection of base classifiers and combined classifiers are also presented. In addition, the contributions made by the novel concepts proposed in this study are verified experimentally.

Chapter 7: Conclusion and Future Work presents concluding remarks regarding the complexity and effectiveness of the systems developed in this study. Selected additional topics deemed to be potentially beneficial to the systems developed during this study are also presented as possible future work.

Chapter 2

Literature Study

“If I have seen further it is by standing on the shoulders of giants.”
- Isaac Newton (1642–1727)

2.1 Introduction

The field of off-line signature verification has enjoyed a great deal of attention over the past few decades. In this chapter we present a collection of verification systems proposed over the years. Although some of these systems may seem dated, they represent noteworthy efforts in the field and also provide the reader with a historical perspective regarding advances made in recent years. For a comprehensive discussion regarding the current state of the art, the reader is referred to Impedovo and Pirlo (2008).

The systems presented in this chapter are based on a wide variety of pattern recognition techniques, namely simple distance classifiers (Section 2.2), dynamic time warping (Section 2.3), hidden Markov models (Section 2.4), neural networks (Section 2.5) and support vector machines (Section 2.6). In addition, selected works pertaining to classifier combination are also discussed in Section 2.7.

Unfortunately, there exists at present no standard library of off-line signatures for verification purposes, and therefore no truly objective benchmark regarding results obtained by any of the systems mentioned. Results reported by the various authors do nonetheless provide the reader with a general idea regarding the effectiveness of their proposed feature extraction techniques and verification strategy.

The discussion of each system is chronologically categorised according to the primary method used for verification, and includes the year of publication and author(s) associated. Each of the possibly numerous features considered is also discussed. Where possible, the nature and composition of the data sets used for training and testing are mentioned, as well as the verification results reported.

2.2 Simple distance classifiers

A simple distance classifier (SDC) models each pattern class with a probability density function (PDF), typically Gaussian, and subsequently relies on the distance computed between a test pattern and such a PDF in order to make a classification.

Fang *et al.* (2001) propose a method based on the so-called *smoothness criterion*, since the authors suggest that the cursive segments of forgeries are generally less smooth than those of genuine signatures. Two approaches are proposed for extracting such a smoothness feature. The *crossing method* involves comparing each stroke segment to its smoothed version, obtained by performing a second-order cubic spline smoothing operation. The second method employs the fractal dimension of each stroke segment to evaluate its smoothness. The obtained smoothness feature is then combined with various global shape features. These include the signature aspect ratio, baseline shift of the vertical projection, the percentage of positively slanted border pixels, as well as the percentage of vertically slanted border pixels. Verification is achieved by means of an SDC based on the Mahalanobis distance. The database considered consists of 1320 genuine signatures and 1320 amateur skilled forgeries. An FRR of 18.1% and FAR of 16.4% is reported.

Majhi *et al.* (2006) implement a novel feature extraction method based on geometric centres. Features are obtained by recursively dividing a signature image into sub-images along horizontal and vertical axes located on the geometric centre of the parent image. Geometric centres of the final sub-images subsequently form the feature vector. An Euclidean distance model is used for classification on a database containing 30 genuine signatures, 10 random forgeries, 10 simple forgeries and 10 skilled forgeries per writer. The number of writers considered during testing is not disclosed. The authors reportedly achieve FARs of 2.08% (random forgeries), 9.75% (simple forgeries) and 16.36% (skilled forgeries), associated with an FRR of 14.58%.

2.3 Dynamic time warping

The DTW algorithm is a popular template matching technique based on dynamic programming and is discussed further in Section 4.3 and Appendix A.

Coetzer (2005) utilises DTW in order to construct a verification system aimed at detecting skilled and simple forgeries. Signature representation is achieved by means of the discrete Radon transform (DRT). Experiments are performed on the *Stellenbosch data set*, collected from 22 writers, containing 30 genuine signatures, 6 simple forgeries and 6 skilled forgeries per writer. The author reports EERs of approximately 18% and 4.5% when considering skilled and simple forgeries, respectively.

Shanker and Rajagopalan (2007) use a DTW algorithm which is modified to incorporate a stability factor, in conjunction with a vertical projection feature. Their signature database, collected from 100 individuals, includes 1075 genuine signatures, 300 simple forgeries, as well as 56 skilled forgeries. The authors reportedly achieve an FRR of 25% and FARs close to 0% and 20% when considering simple and skilled forgeries, respectively.

Güler and Meghdadi (2008) also optimise the basic DTW algorithm in order to detect skilled and random forgeries. Each signature image is converted into a numerical sequence, or *gradient stream*, of localised pixel gradients based on a 4-directional axis. The data set considered is a sub-corpus of the MCYT bimodal database and contains 1000 genuine signatures and 500 skilled forgeries from 50 individuals. EERs of 25.1% (skilled forgeries) and 5.5% (random forgeries) are reported.

2.4 Hidden Markov models

An HMM models each pattern class using a sequence of observations, along with the relationship between individual observations within such a sequence, and is discussed further in Section 4.4 and Appendix B.

Coetzer *et al.* (2004) propose the combination of an HMM-based verifier with DRT-based features. By constructing an HMM utilising a ring topology, rotation invariance is achieved. The system is tested on the Stellenbosch data set (as discussed in the previous section), as well as Dolfing's data set (as considered in this study). Experiments performed on the Stellenbosch data set reportedly yield EERs of approximately 18% and 4.5% when skilled and simple forgeries are considered, respectively. An EER of 12.2% is reported when considering only amateur skilled forgeries from Dolfing's data set. Considering only professional skilled forgeries from Dolfing's data set reportedly yields an EER of 15%.

Oliveira *et al.* (2005) investigate the use of grid segmentation in conjunction with the graphological features pixel density, pixel distribution, progression, slant and form. Experiments are performed on a database collected from 60 individuals, including 2400 genuine signatures, 1200 random forgeries, 600 simple forgeries and 600 skilled forgeries. Feature-specific EERs of 7.87% (pixel density), 7.65% (pixel distribution), 7.92% (slant), 9.15% (progression) and 11.30% (form) are reported.

Wen *et al.* (2009) combine the efforts of a ring-structured HMM with a set of ring-peripheral features based on a transformation-ring-projection. The system is evaluated on a data set containing 2640 signatures, of which 1320 are skilled forgeries, representative of 55 individual writers. The authors report an EER of 11.4%. Furthermore, the system is also tested using a sub-corpus of the MCVT bimodal database, containing 2250 signatures from 75 individuals, resulting in an EER of 15.02%.

2.5 Neural networks

Artificial NNs may generally be described as adaptive parallel computing systems, composed of a multitude of interconnected non-linear computing elements. The primary advantage of an NN is that it can be used to model complex input-output relationships, as it is able to adapt its structure based on information encountered during the learning phase. As a result, the use of NNs for the purposes of signature verification has gained considerable popularity during recent decades.

Huang and Yan (1997) propose a verification system based on geometric feature extraction of localised, aligned shape features. A composite neural network classifier, consisting of several independent feature networks, as well as a single decision network, is also investigated. The system is tested on a data set which contains 504 genuine signatures and 3024 forgeries of varying skill levels, representative of 21 individual writers. The authors reportedly achieve an FAR marginally below 0.05% when only random forgeries are considered, whilst an AER of 11.45% is achieved when considering skilled forgeries.

Baltazakis and Papamarkos (2001) propose the use of a two-stage neural network classifier. The first stage, utilising global, grid and texture features, generates decisions which in turn serve as input to the second stage. This second stage, comprising a radial

basis function (RBF) network structure, subsequently makes the final decision. The system is tested on a data set consisting of approximately 2000 signatures representative of 115 writers. The authors report an FAR and FRR of 9.81% and 3%, respectively.

Armand *et al.* (2006) employ a novel combination of the modified direction feature with additional distinguishing features such as the image centroid, surface area, length and skew. Using the aforementioned feature set, they compare the performance of a resilient back-propagation (RBP) NN with that of an RBF-NN. The systems are tested on a subset of the GPDS signature database, containing 936 genuine signatures and 1170 forgeries of unspecified quality, representative of 39 writers. Verification accuracies of 91.21% and 88.0% are reported when using the RBP-NN and RBF-NN, respectively.

2.6 Support vector machines

SVMs represent a special class of linear classifiers. In order to classify a pattern as belonging to one of two classes, an SVM constructs a hyperplane in the feature space, such that it maximally separates the margin between the two classes. For this reason, SVMs are also referred to as maximum margin classifiers.

Lv *et al.* (2005) investigate the feasibility of combining static and pseudo-dynamic features in their novel SVM-based system, developed specifically for the purpose of Chinese signature verification. Static features considered include moment features and a 16-directional pixel distribution feature, whilst gray distribution and stroke width distribution constitute the pseudo-dynamic feature set. Their experiments are performed on a database representative of 20 writers, containing 25 genuine signatures and 30 amateur skilled forgeries per writer. An AER of about 5% is reported.

2.7 Combined classifiers

The verification systems discussed in the previous sections each employ a single classification technique. In recent years, however, the notion of utilising a set of classifiers in a single verification system has gained notable popularity. The use of such combined classifiers of course yields more stable and accurate systems, but may have been considered computationally exhaustive in the past. In the modern computing environment, however, combined classifiers are both computationally viable and economically attractive.

Santos *et al.* (2004) develop a writer-independent combined classifier utilising graphometric features and a set of NN base classifiers. The signature modelling and verification protocols of this system are based on the questioned document expert's approach. Each writer submits a relatively small reference set of genuine signatures for model training and validation. A separate data set is used for model evaluation. During verification of a questioned signature, each signature in the reference set is independently compared to the questioned signature using a NN classifier, consequently yielding a set of partial decisions. The final decision is obtained by combining these partial decisions using the majority vote rule. Model evaluation is performed using a data set containing 600 genuine signatures (of which 300 are used as reference), 300 random forgeries, 300 simple forgeries and 300 skilled forgeries. The authors report an FRR of 10.33%, associated with FARs

of 15.67% (skilled forgeries), 1.67% (simple forgeries) and 4.41% (random forgeries).

Oliveira *et al.* (2008) also propose a writer-independent combined classifier. This system utilises the grid-based predominant slant and pixel distribution features, each associated with an SVM classifier. Separate data sets are used for model optimisation and evaluation. In addition, the experimental protocol employs 10-fold cross-validation. The use of several score fusion techniques, as well as the majority vote rule, is investigated as a potential combination strategy. Optimal classifier combination is achieved, however, when considering the ROC-based combination of their continuous classifier pair. Model evaluation is performed using a data set containing 600 genuine signatures (of which 300 are used as reference), 300 random forgeries, 300 simple forgeries and 300 skilled forgeries. It is suspected, although not confirmed, that this is the same data set used by Santos *et al.* (2004). The ROC-based combined classifier reportedly yields an FRR of 4.7%, associated with FARs of 1.12% (skilled forgeries), 1.33% (simple forgeries) and 2.04% (random forgeries).

2.8 Concluding remarks

As mentioned earlier, no standard library of signature data is currently available for verification purposes. Any results reported in this chapter therefore provide only a relative measure of success achieved by the authors. Nevertheless, the collection of systems discussed does illustrate the availability of a wide range of techniques and models suitable for the successful implementation of a signature verification system.

Several works found in the literature are, however, of particular interest to the concepts investigated during the course of this study. The utilisation of a grid-based segmentation scheme coupled with local features, as discussed in Justino *et al.* (2000), Justino *et al.* (2005) and Oliveira *et al.* (2005) (see Section 2.4), for example, serve as a foundation for the development of the flexible grid-based feature extraction techniques proposed in this study. In addition, the discussion on ROC-based classifier combination by Oliveira *et al.* (2008) led to the development of the MAROC-based classifier ensemble combination strategy proposed in this study.

Also of particular interest is the work of Coetzer *et al.* (2004), as this system utilises an HMM-based classifier evaluated on the same data set considered in this study. The results reported therefore constitute a realistic, although somewhat relative¹, benchmark for the results reported in this study.

In Chapters 3 and 4, we discuss the various stages comprising the design and implementation of the base classifiers developed in this study. Chapter 5 presents the classifier combination strategies implemented on the resulting classifier ensemble.

¹The system developed by Coetzer *et al.* does not adhere to the same experimental protocol considered in this study. Certain discrepancies may therefore be expected if the results are to be compared directly.

Chapter 3

Image Processing and Feature Extraction

“The laws of nature are written in the language of mathematics . . . the symbols are triangles, circles and other geometrical figures, without whose help it is impossible to comprehend a single word.”

- Galileo Galilei (1564–1642)

3.1 Introduction

The process of feature extraction constitutes one of the fundamental components of the pattern recognition process, as it enables a verification system to represent signature patterns in an intelligent and robust manner. The feature vector representation generated by this process may subsequently be used to train a suitable classification model.

In this chapter we explain how the DTW and HMM base classifiers developed in this study convert a raw signature image into a suitable feature vector representation. In Section 3.2, we present an efficient noise removal algorithm and show that it is well equipped to successfully remove both standard impulse noise and high density impulse noise regions. In Section 3.3.1, we discuss the traditional rigid grid-based image segmentation strategy, whilst Section 3.3.2 introduces the flexible grid segmentation strategy - a novel extension of the rigid grid-based approach. We show that each of these segmentation strategies ensures a feature vector representation that is both translation and scale invariant. The issue of rotation invariance is not addressed in this study, as the data set used during experimentation (see Section 6.2) is already normalised in this regard.

The set of grid-based feature extraction techniques considered in this study is discussed in Sections 3.4.1–3.4.4. In Section 3.4.5, we explain how the aforementioned segmentation and feature extraction techniques are combined, in order to construct a set of suitable feature vectors. Finally, Section 3.5 provides a discussion on vector quantisation as essential feature vector post-processing, required specifically by the HMM base classifiers developed in this study.

3.2 Noise removal

As mentioned in Section 1.4.2, the data set considered in this study is completely free of background noise and therefore requires no preprocessing with respect to noise removal. In any practically deployable verification system, however, the process of noise removal constitutes an integral and generally unavoidable preprocessing stage. For this reason, the successful incorporation of an efficient noise removal algorithm is deemed fit for discussion within the context of this study.

During the document digitisation process, it is possible that unwanted foreign objects, such as dust or ink residue, are present in the obtained signature image. These pixel anomalies often adversely affect the successful completion of both the signature segmentation and feature extraction processes, as they contribute shape information not pertaining to the signature being represented.

A popular and effective means of achieving noise removal is by implementation of the *adaptive median filter* (AMF), an extension of the well known order-statistics median filter. As is its predecessor, the AMF is well suited for removing impulse noise. Furthermore, as it makes use of a dynamically adjustable filter window, it is also capable of removing noise regions possessing a higher spatial density. Perhaps the greatest advantage of using the AMF, however, is the fact that it is, unlike the median filter, detail preserving.

As suggested in Swanepoel (2007), however, it is suspected that non-uniform stroke width greatly impedes the level of success achieved by the AMF. Observations indicate that the optimal maximum dimensions of the adjustable filter window are linearly proportional to the stroke width. By letting this user-defined parameter exceed a certain threshold, one risks decimating image segments belonging to the signatures themselves, consequently destroying valuable shape information.

For a comprehensive discussion of the AMF-algorithm, as illustrated in Figure 3.1, the reader is referred to Gonzalez and Woods (2002).

3.3 Signature segmentation

During the course of this study, two signature segmentation strategies are considered. These include the traditional grid segmentation scheme, as well as a novel extension of the grid-based approach, referred to as flexible grid segmentation. In order to avoid confusion, the traditional grid segmentation strategy is henceforth explicitly referred to as *rigid grid segmentation*.

3.3.1 Rigid grid segmentation

The use of grid-based segmentation strategies have gained notable popularity during recent years. At present, there exists two distinct methods regarding segmentation grid construction, each with its own set of benefits and limitations. The first is based on a fixed grid cell size, whilst the second is based on a fixed grid resolution. In general, the feature extraction technique and classification model considered dictates which of these strategies is appropriate. In this study, the latter approach is deemed most suitable, as

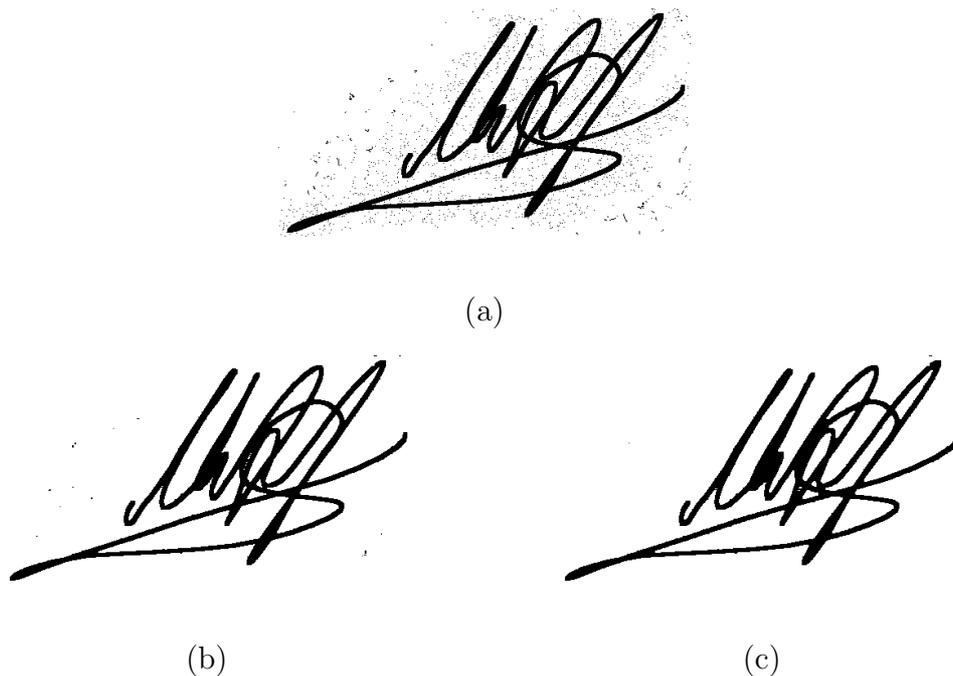


Figure 3.1: Signature image noise removal by means of the AMF, as implemented by Swanepoel (2007). (a) A signature image containing synthetically generated impulse noise, as well as high density noise regions. (b) The corrected image, as obtained by implementing the standard median filter. Although the impulse noise is successfully removed, the median filter is incapable of correcting areas possessing high density noise. (c) The corrected image, as obtained by implementing the AMF. Practically all traces of noise have been removed

it not only enables complete control of both the observation sequence length and feature space dimension (see Section 3.4.5), but also plays an important role in ensuring scale invariant feature vectors.

The first step in constructing such a segmentation grid involves calculating an appropriate grid perimeter. This is achieved by obtaining the signature bounding box, thereby isolating only pixels representative of pen strokes. The importance of an efficient noise removal algorithm should now become apparent, as any noise present in the outer regions of a signature image would greatly impair the construction of a suitable grid perimeter. The successful construction of such a perimeter not only enables a feature extraction algorithm to discard the image regions which contribute no shape information, but also ensures that any resulting feature vectors remain translation invariant.

The signature image region I contained within such a bounding box is subsequently divided into a set of sub-images $\{I_{ij}\}$, where $i = 1, 2, \dots, M$ and $j = 1, 2, \dots, N$, whilst M and N denote the number of rows and columns contained within the segmentation grid, respectively. Since all grid cells are constrained to be equally sized, it is possible that the initial grid perimeter may need to be adjusted accordingly, resulting in a small percentage of pixels located on the extremities of the signature image to fall outside the modified grid perimeter. This percentage of potential data loss, however, is deemed negligible. The rigid grid segmentation process is illustrated in Figure 3.2.

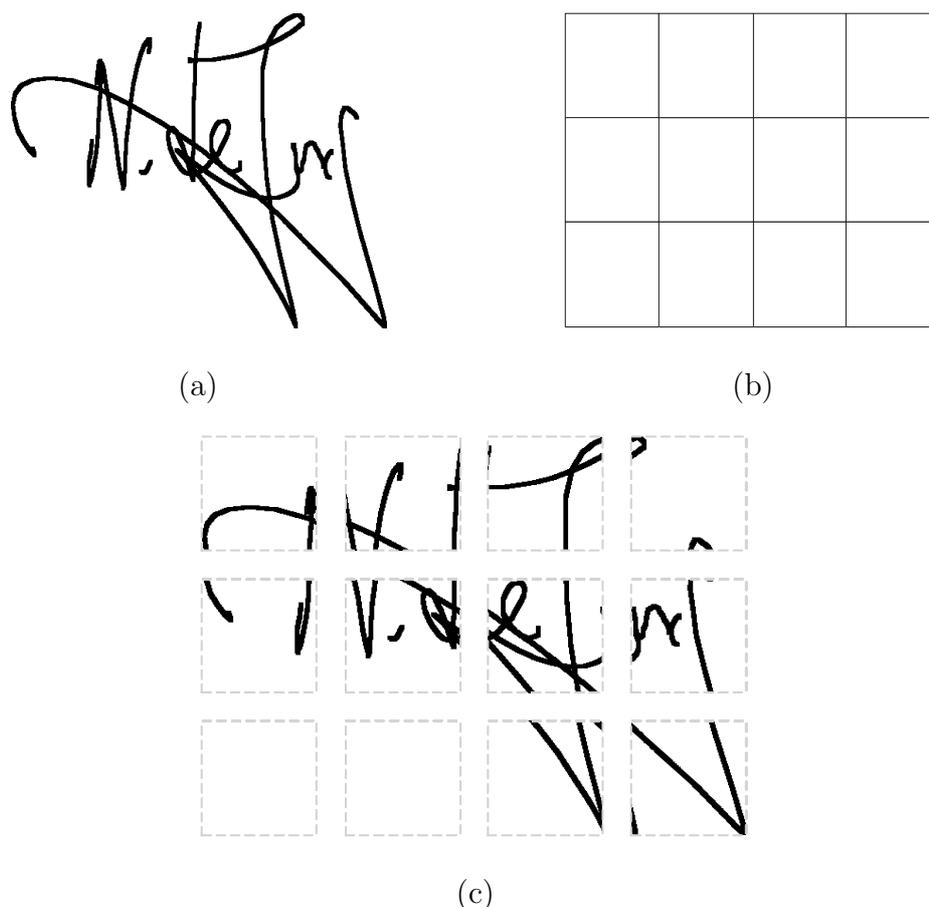


Figure 3.2: The rigid grid segmentation strategy. (a) The original signature image I , (b) a 3×4 rigid segmentation grid and (c) the resulting image segmentation $\{I_{ij}\}$.

3.3.2 Flexible grid segmentation

In this study, we propose a novel extension to the rigid grid segmentation strategy, referred to as *flexible grid segmentation*, by introducing the concept of *flexible grid cell boundaries*. After completing the rigid grid segmentation, as described in the previous section, each grid cell boundary is dilated horizontally by a factor F_x and vertically by a factor F_y of the original cell width or height, respectively. The dilation of any given grid cell boundary is immediately halted, however, upon reaching the grid perimeter. These so-called *flexibility parameters* may be interpreted as the proportion of adjacent rigid grid cells incorporated into a flexible grid cell. Both F_x and F_y may be assigned arbitrary values, although the constraints

$$0 \leq F_x \leq N - 1, \quad (3.1)$$

$$0 \leq F_y \leq M - 1 \quad (3.2)$$

are imposed to ensure a legitimate $M_{F_y} \times N_{F_x}$ *flexible segmentation grid*.

Constraint 3.1 enforces two equally important flexibility limits. Firstly, $F_x \geq 0$ ensures that no flexible grid cell may contract, which would result in a loss of signature

shape information. Secondly, for all values of $N > 1$, the requirement $F_x \leq N - 1$ ensures that at least one flexible grid cell is strictly contained within the grid perimeter. If this were not the case, each cell boundary could potentially coincide with the grid perimeter, resulting in needless data repetition. A similar argument follows for Constraint 3.2. Furthermore, it should be clear that $F_x = F_y = 0$ represents the trivial case, as it is equivalent to rigid grid segmentation.

Although imposing Constraints 3.1 and 3.2 produces a valid segmentation grid, additional restrictions are required to ensure efficient feature extraction. Consider, for example, the construction of a flexible segmentation grid for which $F_x \rightarrow N - 1$ and $F_y \rightarrow M - 1$. Although technically valid, the resulting signature image segmentation would produce exceedingly similar sub-images, consequently leading to *over-smoothing* of the feature set. For this reason, it is recommended that Constraints 3.1 and 3.2 be modified to include

$$0 \leq F_x \leq l_x \ll N - 1, \quad (3.3)$$

$$0 \leq F_y \leq l_y \ll M - 1, \quad (3.4)$$

where l_x and l_y denote the maximum number of horizontally and vertically adjacent rigid grid cells, respectively, reachable from any given flexible grid cell.

By allowing adjacent grid cells to overlap, the flexible grid segmentation scheme therefore not only possesses all the capabilities of rigid grid segmentation regarding signature analysis, but also inherently provides additional information regarding signature progression on a global scale. The flexible grid segmentation strategy is illustrated in Figure 3.3.

3.4 Feature extraction

The utilisation of grid-based segmentation schemes enables the base classifiers developed in this study to consider various *local features*. These features allow the analysis of signature images on both stroke and sub-stroke level, thereby providing a robust platform for signature representation.

A collection of diverse local features are considered in this study, namely pixel density, gravity centre distance, orientation and predominant slant. The reader is reminded that each of the MN signature sub-images, as generated during the segmentation process, is represented by a binary $m \times n$ image matrix I_{ij} . For the purposes of the discussions presented in Sections 3.4.1–3.4.4, however, the simplified notation J is used to represent an arbitrary signature sub-image. Furthermore, a pixel value of 1 is considered indicative of a pen stroke coordinate, whilst zero-valued pixels denote the document background.

As shown in Sections 3.4.1–3.4.3, the discussion of these features is greatly simplified by the incorporation of *image moments*, as proposed by Hu (1962). Image moments generally represent a weighted average of image pixel intensities and have proven useful for object description throughout the literature. The image moment M_{pq} , said to be of *order* $p + q$, is defined for an $m \times n$ image J as

$$M_{pq}(J) = \sum_{i=1}^m \sum_{j=1}^n i^p j^q J(i, j). \quad (3.5)$$

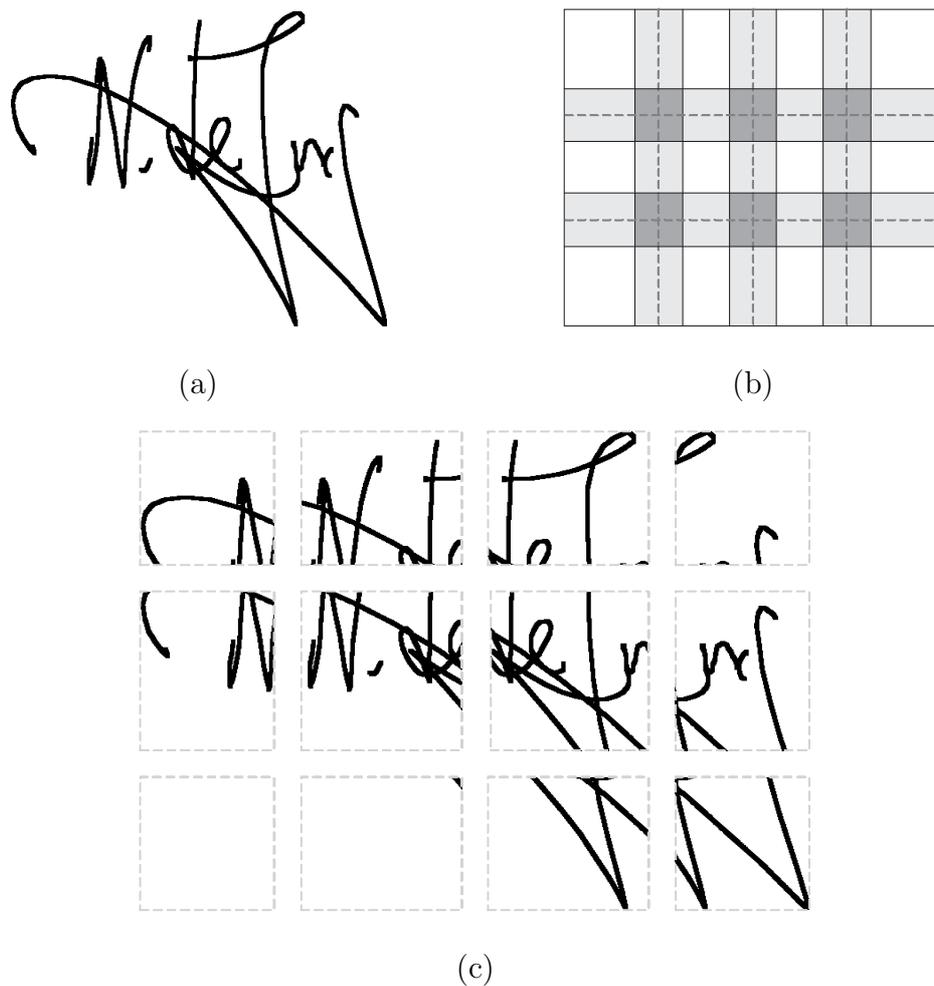


Figure 3.3: The flexible grid segmentation strategy. (a) The original signature image I . (b) A 30.25×40.25 flexible grid, where the dotted lines indicate a 3×4 rigid grid, whilst the shaded areas indicate the degree of overlap between adjacent grid cells. (c) The resulting image segmentation $\{I_{ij}\}$. Note that a portion of I is shared between each pair of adjacent grid cells. Also note that each flexible grid cell is dynamically sized according to its proximity to the grid perimeter.

3.4.1 Pixel density

The pixel density feature $x_{\text{PD}} \in [0, 1]$ has been used to great effect in such works as Justino *et al.* (2001) and Oliveira *et al.* (2005). Since the pixel density of a signature segment is directly linked to stroke width, it is also commonly referred to as *apparent pen pressure*. For this reason, the pixel density feature is said to contain pseudo-dynamic signature information. The pixel density of an image J is obtained by computing the ratio of pen stroke pixels to total image pixels, or

$$x_{\text{PD}}(J) = \frac{M_{00}(J)}{mn}. \quad (3.6)$$

3.4.2 Gravity centre distance

The gravity centre distance feature $x_{\text{GCD}} \in [0, 1]$, as described in Justino *et al.* (2005), is obtained in two phases. Firstly, the coordinates of the image centroid (\bar{x}, \bar{y}) are computed using

$$\bar{x} = \frac{M_{10}(J)}{M_{00}(J)}, \bar{y} = \frac{M_{01}(J)}{M_{00}(J)}. \quad (3.7)$$

Thereafter, the gravity centre distance associated with J , as illustrated in Figure 3.4, can be computed as follows,

$$x_{\text{GCD}}(J) = \frac{\sqrt{\bar{x}^2 + \bar{y}^2}}{\sqrt{m^2 + n^2}}. \quad (3.8)$$

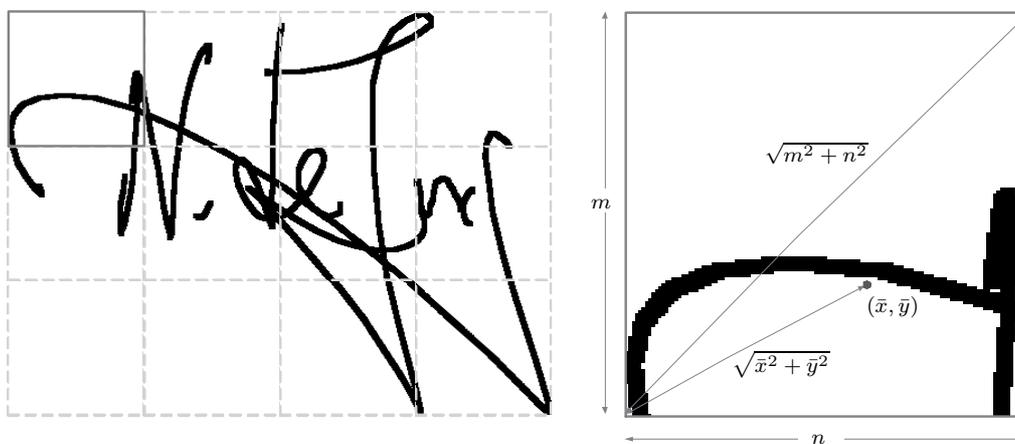


Figure 3.4: Computation of the gravity centre distance feature.

3.4.3 Orientation

The orientation feature $x_{\text{ORT}} \in [-90^\circ, 90^\circ)$ is defined as the angle ϕ between the horizontal coordinate system axis and the major axis of the ellipse possessing the same second moment as the image J .

Similar to pixel density, this feature is also said to contain pseudo-dynamic signature information, as the orientation of a signature stroke segment may be linked to *estimated pen velocity*¹. This is of course greatly dependent on the grid resolution considered during segmentation. For relatively low grid resolutions, each signature region considered may contain multiple stroke segments, thereby nullifying the potential for pseudo-dynamic signature analysis.

¹Only the directional component of pen velocity may be inferred from the orientation of a pen stroke. In order to estimate stroke speed, such features as *progression*, for example, may be used. The progression feature is discussed further in Oliveira *et al.* (2005).

Computation of the orientation feature is based on the method proposed by Haralick and Shapiro (1992). Firstly, we compute the second moments M_{xx} , M_{yy} and M_{xy} as

$$M_{xx} = \frac{M_{20}(J)}{M_{00}(J)}, M_{yy} = \frac{M_{02}(J)}{M_{00}(J)}, M_{xy} = M_{11}(J). \quad (3.9)$$

The orientation feature, as illustrated in Figure 3.5, is subsequently computed as

$$x_{\text{ORT}}(J) = \arctan \left(\frac{M_{yy} - M_{xx} + \sqrt{(M_{yy} - M_{xx})^2 + 4M_{xy}^2}}{2M_{xy}} \right). \quad (3.10)$$

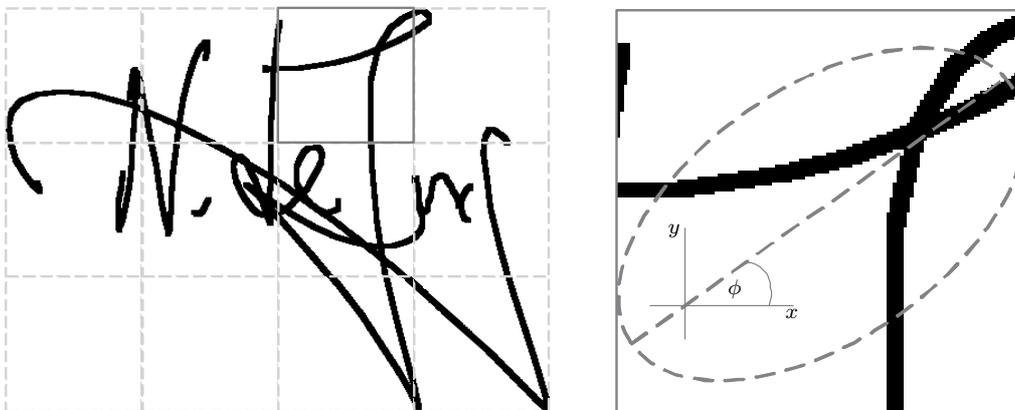


Figure 3.5: Computation of the orientation feature. Indicated with dotted lines alongside the segmented image region is the ellipse possessing the same second moments, as well as its major axis. The angle ϕ denotes the resulting orientation feature value.

3.4.4 Predominant slant

In order to compute the predominant slant feature $x_{\text{PS}} \in \{1, 2, \dots, S\}$, where S denotes the number of user-defined slant elements considered, the two-stage method proposed by Justino *et al.* (2000) is implemented. Firstly, the image skeleton J_s is extracted from J , as illustrated in Figure 3.6, using the skeletonisation algorithm discussed in Gonzalez and Woods (2002). Subsequently, J_s is traversed with each of the S individual 8-neighbourhood slant elements. The base classifiers developed in this study consider 4 such elements, defined as

$$1: \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}, 2: \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, 3: \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, 4: \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

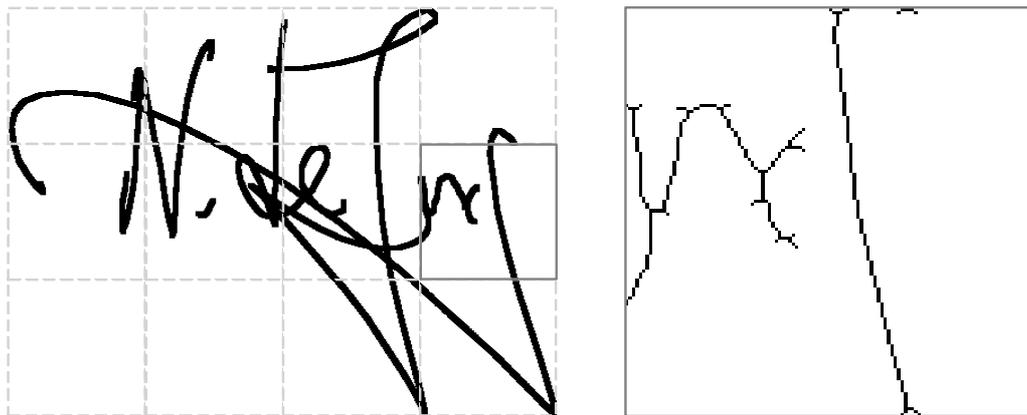


Figure 3.6: Computation of the predominant slant feature. The image skeleton clearly exposes numerous straight line segments to be identified by the set of slant elements, consequently producing the predominant slant feature value.

An occurrence record of each slant element is kept by means of a symbol counter n_i , where $i = 1, 2, \dots, S$ denotes the symbol number. Subsequently, the symbol that occurs most frequently is selected to represent predominant slant, or

$$x_{\text{PS}}(J) = \underset{i=1,2,\dots,S}{\operatorname{argmax}}[n_i]. \quad (3.11)$$

3.4.5 Feature vector construction

As mentioned in Section 1.2.1, each signature pattern \mathbf{X} is represented by a set of T , d -dimensional feature vectors $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$. Sections 3.3.1 and 3.3.2 propose two segmentation strategies for dividing a signature image into a set of sub-images, whilst Sections 3.4.1–3.4.4 discuss several methods for obtaining a scalar-valued feature from any given sub-image.

In this section, these concepts are combined in order to construct a set of intelligent, robust feature vectors, thereby completing the feature extraction process. Given a signature image I , we first perform grid-based segmentation, yielding a set of MN signature sub-images $\{I_{ij}\}$. By extracting a single feature $x_f(I_{ij})$, where $f \in \{\text{PD}, \text{GCD}, \text{ORT}, \text{PS}\}$, from each of these sub-images, I is subsequently encoded as

$$\mathbf{X} = \begin{bmatrix} x_f(I_{11}) & x_f(I_{12}) & \dots & x_f(I_{1N}) \\ x_f(I_{21}) & x_f(I_{22}) & \dots & x_f(I_{2N}) \\ \vdots & \vdots & \ddots & \vdots \\ x_f(I_{M1}) & x_f(I_{M2}) & \dots & x_f(I_{MN}) \end{bmatrix}. \quad (3.12)$$

As illustrated in Figures 3.2 and 3.3, it is quite possible that a grid-based segmentation scheme may occasionally result in signature sub-images that contain no pen stroke pixels at all. In such cases, we let $x_f(I_{ij}) = 0$, regardless of the feature being represented.

Subsequently, each column² of \mathbf{X} is chosen to represent a feature vector \mathbf{x}_i , thereby yielding a set of N , M -dimensional feature vectors

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}. \quad (3.13)$$

As mentioned in Section 3.3.1, the use of a grid-based segmentation strategy ensures translation invariant feature vectors. Furthermore, any fixed-resolution segmentation grid automatically adapts its grid cell size according to the signature image dimensions. This property, used in conjunction with the scale invariant local features discussed in Sections 3.4.1–3.4.4, therefore also ensures a scale invariant feature vector representation.

3.5 Vector quantisation

As discussed in the previous section, each signature is represented by a set of T feature vectors in \mathfrak{R}^d . The HMM base classifiers developed in this study, however, are designed specifically for discrete observation sequences. For this reason, the efforts of a vector quantiser are required.

3.5.1 Overview

A vector quantiser Q is formally defined as a mapping from a d -dimensional vector space \mathfrak{R}^d onto a finite set of K distinct *codewords* or *symbols* $\mathbf{V} = \{v_1, v_2, \dots, v_K\}$, or

$$Q : \mathfrak{R}^d \mapsto \mathbf{V}. \quad (3.14)$$

The set \mathbf{V} is generally referred to as a K -level *codebook* or *symbol alphabet*.

The VQ process is associated with a partitioning of the vector space \mathfrak{R}^d into K regions $\mathbf{R} = \{R_1, R_2, \dots, R_K\}$, such that each region R_k contains those vectors $\mathbf{x} \in \mathfrak{R}^d$ which are mapped by Q to the codeword v_k , or

$$R_k = Q^{-1}(v_k), \quad (3.15)$$

where

$$\bigcup_{k=1}^K R_k = \mathfrak{R}^d. \quad (3.16)$$

In practice, the VQ process is analogous to pattern classification by means of an SDC, where each codeword region R_k denotes a pattern class. When using SDCs, each pattern class is typically represented by a d -dimensional Gaussian PDF

$$f(\mathbf{x}|R_k) = \frac{1}{(2\pi)^{\frac{d}{2}} \sqrt{|\boldsymbol{\Sigma}_k|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)}. \quad (3.17)$$

Each region R_k is therefore completely specified by its associated *mean vector* $\boldsymbol{\mu}_k$ and *covariance matrix* $\boldsymbol{\Sigma}_k$, which may be estimated from a set of sample patterns as

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \mathbf{x}_i, \quad (3.18)$$

²Feature vectors are constructed from the feature matrix *columns* specifically, due to the natural progression of Western handwriting from left to right.

$$\Sigma_k = \frac{1}{N_k - 1} \sum_{i=1}^{N_k} (\mu_k - \mathbf{x}_i)(\mu_k - \mathbf{x}_i)^T, \quad (3.19)$$

where $\mathbf{x}_i \in R_k$ and N_k denotes the number of sample patterns assigned to R_k .

Furthermore, as each codeword is a mapping of vectors in the feature space to a compacted representation, each encoding operation results in a quantisation error

$$\epsilon_k(\mathbf{x}|v_k) = D(\mathbf{x}, f(\mathbf{x}|R_k)). \quad (3.20)$$

This error, commonly referred to as the *distortion*, may be calculated using any suitable distance measure, and is used to gauge the representative ability of codeword v_k . It is therefore the primary objective of any quantiser to minimise the *overall distortion*, defined as $\sum_{k=1}^K \epsilon_k(\mathbf{x}|v_k)$. This is achieved by the iterative re-assignment of sample patterns and subsequent recalculation of the regions R_k , until the overall distortion converges to a local minimum.

3.5.2 Implementation

Each HMM base classifier C_i developed in this study is associated with a unique quantiser Q . Each corresponding codebook \mathbf{V} is generated by means of the well-known *K-means clustering* algorithm, as discussed in Bishop (2006). Although trials conducted by Alpaydin (1998) suggest that the *EM inference* algorithm for Gaussian mixtures outperforms any of its known peers, *K-means clustering* remains, in the context of signature verification, the most popular VQ algorithm found in the literature. This is most likely due to two main performance criteria. Firstly, the *K-means* algorithm generates sufficiently accurate³ codebooks, without being computationally exhaustive. Secondly, this method circumvents the infamous *curse of dimensionality*, where either the size or nature of the codebook training data may lead to unreliable, or even singular, codeword covariance matrices.

Since each codeword v_k in this study is specified using its mean vector only, Q is referred to as a *hard* quantiser. Any unknown vector \mathbf{x}_q submitted for encoding is subsequently associated with the codeword region having the nearest centroid. Furthermore, the distortion measure associated with v_k is consequently based on a Euclidean distance measure

$$\begin{aligned} \epsilon_k(\mathbf{x}|v_k) &= D_{\text{Eucl}}(\mathbf{x}_q, f(\mathbf{x}|R_k)) \\ &= \sqrt{(\mathbf{x}_q - \mu_k)^T (\mathbf{x}_q - \mu_k)}. \end{aligned} \quad (3.21)$$

Even though the *K-means* algorithm is both simple and reliable, certain measures are taken in this study, in order to ensure maximal efficiency. As mentioned in the previous section, a codebook \mathbf{V} contains exactly K codewords. It is possible, however, for sufficiently large values of K , that certain regions are not assigned any sample patterns during codeword reestimation. An empty region R_n can therefore also have no associated mean μ_n , resulting in a *null codeword*.

³The accuracy of a VQ codebook \mathbf{V} is measured on the basis of overall distortion, as discussed in the previous section.

In order to address this issue, the method proposed by Tanguay (1993) is employed. Consequently, K does not necessarily denote the size of a K -level codebook, but rather its upper bound. If, during training, a null codeword \mathbf{v}_n is encountered, it is simply removed from the codebook. Furthermore, n is added to the set K_\emptyset , containing all null codeword indices. The partition

$$\bigcup_{\substack{k=1 \\ k \notin K_\emptyset}}^K R_k = \mathfrak{R}^d \quad (3.22)$$

remains complete, as the subspace of \mathfrak{R}^d previously occupied by $R_n = \emptyset$ is simply assimilated by its neighbouring regions.

3.6 Concluding remarks

In this chapter we explained how the set of base classifiers developed in this study make use of several image processing and data processing techniques, in order to convert a raw signature image into a suitable feature vector representation.

Traditionally, when developing a single signature verification system, one would construct feature vectors comprised of several independent feature types, thereby ensuring greater separation of patterns in the feature space. As one of the primary objectives of this study is the combination of a classifier ensemble, however, each base classifier is associated with a single feature type.

In the next chapter, we introduce the classification models considered in this study, namely DTW and HMMs, and explain how each is used to model a writer's signature based on the feature vector representation yielded by the algorithms discussed in this chapter.

Chapter 4

Signature Modelling and Verification

“The oldest, shortest words - ‘yes’ and ‘no’ - are those which require the most thought.”
- Pythagoras (582BC–497BC)

4.1 Introduction

In the previous chapter we discussed how a raw signature image is converted into a robust feature vector representation, considering various grid-based techniques, thereby completing the feature extraction phase. In this chapter we explain how these feature-based signature representations may be used to construct writer-dependent models used for signature verification, thereby completing the classification phase, as well as the underlying pattern recognition process, as discussed in Section 1.2.1.

Section 4.2 introduces several key concepts regarding the writer-dependent approach to signature verification. Sections 4.3 and 4.4 focus on the base classifiers developed in this study and show that they employ two fundamentally different classification techniques. The first classifier, a template matching technique based on DTW, is presented in Section 4.3. We explain the advantages of using the DTW-based approach to vector matching as an alternative to other popular distance measures. The second classifier, a discrete observation left-right HMM, is presented in Section 4.4. We explain the general advantages of utilising an HMM-based approach to signature modelling, as well as the reasoning behind the specific HMM design considered in this study.

Note that Sections 4.3 and 4.4 provide condensed discussions of the classification techniques used for signature modelling and verification. Detailed discussions of the theoretical background required for DTW and HMM development are reserved for Appendices A and B, respectively. Although the topics discussed in these appendices are of vital importance to the successful implementation of the DTW and HMM base classifiers, they are not deemed central within the context of this chapter.

4.2 Overview

In this section we introduce several key aspects generally associated with signature modelling and verification. For details regarding the specific protocols considered by the DTW and HMM base classifiers developed in this study, see Sections 4.3 and 4.4, respectively.

4.2.1 Modelling

The act of signature creation is a dynamic, time-varying process, since it is physically impossible for any writer to exactly duplicate a signature in successive attempts. Certain variations therefore become evident in the feature sets of different genuine signatures belonging to a single writer. This phenomenon is commonly referred to as *intra-class variability*. Consequently, it becomes essential for any efficient verification system to sufficiently understand and compensate for such variations.

In general, given a set of training signatures, the construction of a writer-dependent signature model is based on a certain predefined reference entity. In this study, for example, the reference entities used by the DTW and HMM base classifiers are represented by a signature pattern \mathbf{X}_k (see Section 4.3.2) and hidden Markov model λ (see Section 4.4.4), respectively. Following possible model optimisation, any signature pattern subsequently submitted for verification is matched against such a reference entity, thereby yielding an appropriate classifier score.

In order to compensate for the intra-class variability associated with writer ω , certain writer-specific statistics are also included in the signature model. The most popular statistics are undoubtedly μ_ω and σ_ω , denoting the mean and standard deviation, respectively, of the classifier scores obtained from the training set of writer ω . Whilst μ_ω represents a benchmark for the classifier score obtained from a typical genuine signature belonging to writer ω , the measure of tolerable score variability obtained from the training set is quantified by σ_ω .

These statistics are therefore also said to estimate the confidence distribution¹ of genuine signatures belonging to writer ω , as illustrated in Figure 4.1. In an ideal scenario, this estimated distribution would include any and all genuine signatures produced by writer ω , whilst excluding all forgeries. As shown in the next section, however, this is generally not a valid assumption.

4.2.2 Verification

The presence of intra-class variability, as discussed in the previous section, generally prohibits a verification system from producing a definitive decision as to the origin of a questioned signature. A more realistic approach involves associating a probabilistic measure of confidence to the validity of a questioned signature's claim of ownership. A confidence of 1 is associated with a perfect match, whilst a confidence of 0 is associated with a complete mismatch.

During the verification process, given a questioned signature pattern \mathbf{X} and claim of ownership ω , each base classifier first computes an appropriate measure of dissimilarity, denoted by $\delta \in [0, \infty)$, between \mathbf{X} and the trained model for writer ω . Subsequently, in order to employ a global decision threshold τ , it is required that δ undergoes *score normalisation*. This process compensates for writer-specific variability, consequently yielding the confidence score $s_c \in [0, 1]$.

¹Although the confidence distributions of genuine and forged signatures do not typically fall into a well-identifiable class, the idealisation of representing each as a Gaussian distribution is commonly used throughout the literature.

Although the process of score normalisation may seem relatively simple, the selection of an appropriate normalisation method is definitely not a trivial endeavour, as shown in Jain *et al.* (2005). The base classifiers developed in this study perform score normalisation by means of the monotonically decreasing function

$$\begin{aligned} s_c &= N(\delta, \mu_\omega, \sigma_\omega) \\ &= \frac{1}{1 + e^{6(\frac{\delta}{\mu_\omega + \sigma_\omega} - 1)}}, \end{aligned} \quad (4.1)$$

where μ_ω and σ_ω denote the writer-specific statistics obtained during model training. Equation 4.1, an adaptation of the well-known *logistic function*, is in many aspects comparable to the *double-sigmoid*² normalisation function proposed by Cappelli *et al.* (2000). Note that Equation 4.1 is specifically designed to convert a dissimilarity measure into a confidence score, although it may easily be adapted to perform normalisation of a similarity measure.

Subsequently, if $s_c \geq \tau$, \mathbf{X} is classified as belonging to G (the positive class). Otherwise, \mathbf{X} is classified as belonging to F (the negative class). Based on the probabilistic definition of s_c , it could logically follow that the discrete decision threshold $\tau = 0.5$ would generally provide the most accurate classification of a questioned signature. This is, however, hardly ever the case, since G and F rarely occupy clearly separable confidence distributions, as illustrated in Figure 4.1.

The probable overlapping of G and F gives rise to two important subsets. Firstly, we define $G^F \subset G$ as the set of genuine signatures misclassified as forgeries. Similarly, $F^G \subset F$ is defined as the set of forgeries misclassified as genuine. Ultimately, we are interested in the *optimal verification threshold* τ^* , which minimises the area occupied by $G^F \cup F^G$.

4.3 Dynamic time warping

4.3.1 Overview

When utilising a template matching technique, a questioned feature set is matched to a feature set representative of the claimed pattern class, in order to obtain a measure of dissimilarity. The feature vectors contained within these sets are commonly referred to as *test vectors* \mathbf{x}_q and *reference vectors* \mathbf{x}_k , respectively.

The simplest method for matching two vectors is to calculate the Euclidean distance between them, that is

$$D_{\text{Eucl}}(\mathbf{x}_q, \mathbf{x}_k) = \sqrt{(\mathbf{x}_k - \mathbf{x}_q)^T (\mathbf{x}_k - \mathbf{x}_q)}. \quad (4.2)$$

From Equation 4.2 it is clear that a Euclidean distance measure is based on the differences between the corresponding elements of the test and reference vectors.

²In order to implement the piecewise-defined double-sigmoid normalisation function, statistical knowledge of both the genuine signature and forgery distributions is required. This approach is therefore not suitable for the writer-dependant models considered in this study.

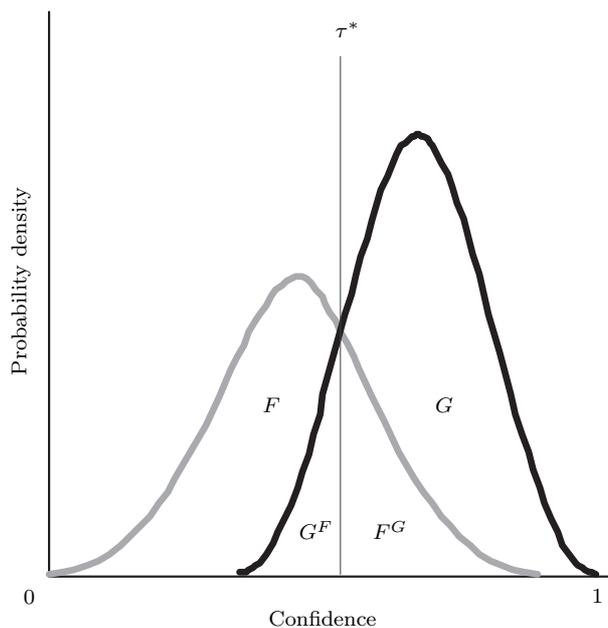


Figure 4.1: Hypothetical Gaussian confidence distributions for genuine signatures (G) and forgeries (F), as well as their respective misclassification subsets G^F and F^G . Also indicated is the optimal verification threshold τ^* such that $G^F \cup F^G$ is minimised.

This approach, however, is not always desirable, since subtle variations in the composition of the signatures may result in certain features not representing the same signature segments. This likelihood increases dramatically as higher grid resolutions are employed during the feature extraction phase.

For this reason, the DTW-algorithm has long been the preferred candidate for use in pattern recognition systems based on template matching. This is due to the fact that the DTW-algorithm, as illustrated in Figure 4.2, is able to non-linearly align two feature vectors, prior to calculating the Euclidean distance between these aligned vectors. This is achieved by employing techniques based on dynamic programming. The obtained DTW-based vector distance, whilst still based on the Euclidean distance measure, consequently matches the vector elements based on similarity rather than location. This approach to vector matching therefore greatly improves the overall stability of the associated verification system.

The DTW-algorithm, as implemented in this study, is discussed in more detail in Appendix A. For a comprehensive discussion on the theory of DTW and the applications thereof, the reader is referred to Coetzer (2005).

4.3.2 Model training

In order to construct an adequately representative writer-dependent model, each writer ω submits a set of K_ω genuine signatures, represented by

$$\{\mathbf{X}_1^{(\omega)}, \mathbf{X}_2^{(\omega)}, \dots, \mathbf{X}_{K_\omega}^{(\omega)}\}, \quad (4.3)$$

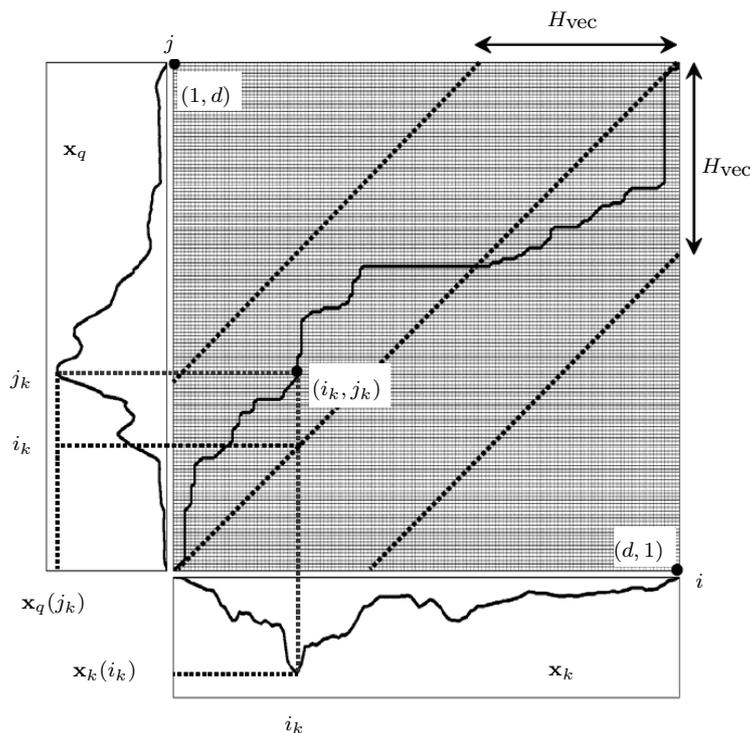


Figure 4.2: Illustration of the feature vector alignment process utilised by DTW. The algorithm identifies similar features contained within the test and reference vectors. The resulting dissimilarity measure is based on the optimal path obtained between these vectors, as opposed to simply matching corresponding components. For the DTW base classifiers developed in this study, feature vectors extracted from the test and reference patterns have the same dimension d . The internal parameter H_{vec} , referred to as the *bandwidth*, is used to regulate the algorithm’s *flexibility*, and is discussed in Section A.1.

for training. The first step entails selecting the reference signature $\mathbf{X}_k^{(\omega)}$, which subsequently acts as a template for signatures belonging to writer ω . In this study, we define $\mathbf{X}_k^{(\omega)}$ as the *most representative* signature found in the training set. In order to determine this most representative signature, each training pattern $\mathbf{X}_i^{(\omega)}$ is matched to every other pattern $\mathbf{X}_{j \neq i}^{(\omega)}$ in the training set, after which the signature that yields the smallest average dissimilarity to the other training signatures is deemed the most representative. Signatures are matched on the basis of the DTW-based distance between them, denoted by $D(\mathbf{X}_i^{(\omega)}, \mathbf{X}_j^{(\omega)})$, as discussed in Section A.1.

In addition, the writer-specific statistics μ_ω and σ_ω are also determined by matching the identified reference signature to every other training signature. These statistics are computed as

$$\mu_\omega = \frac{1}{K_\omega - 1} \sum_{\substack{i=1 \\ i \neq k}}^{K_\omega} D(\mathbf{X}_i^{(\omega)}, \mathbf{X}_k^{(\omega)}), \quad (4.4)$$

$$\sigma_\omega = \frac{1}{K_\omega - 2} \sum_{\substack{i=1 \\ i \neq k}}^{K_\omega} (D(\mathbf{X}_i^{(\omega)}, \mathbf{X}_k^{(\omega)}) - \mu_\omega)^2. \quad (4.5)$$

The writer-specific statistics, in conjunction with the reference signature, comprise the writer model, denoted by

$$M_\omega = (\mathbf{X}_k^{(\omega)}, \mu_\omega, \sigma_\omega). \quad (4.6)$$

This training process is performed for each of the Ω writers enrolled into the system database, yielding

$$\{M_1, M_2, \dots, M_\Omega\}, \quad (4.7)$$

thereby completing the DTW-based model training phase.

4.3.3 Verification

When a DTW base classifier receives a questioned signature pattern \mathbf{X}_q and claim of ownership ω , it first needs to obtain the dissimilarity measure δ by matching \mathbf{X}_q to the trained model M_ω . As discussed in the previous section, this is achieved by computing

$$\delta = D(\mathbf{X}_q, \mathbf{X}_k^{(\omega)}). \quad (4.8)$$

In order to map $\delta \in [0, \infty)$ to the confidence interval $[0, 1]$, Equation 4.1 is applied, yielding

$$s_c = N(\delta, \mu_\omega, \sigma_\omega). \quad (4.9)$$

Finally, a sliding global threshold $\tau \in [0, 1]$ is used to obtain a verification decision D , such that

$$D = \begin{cases} 1, & \text{if } s_c \geq \tau \\ 0, & \text{if } s_c < \tau \end{cases}. \quad (4.10)$$

The signature pattern \mathbf{X}_q is accepted as genuine if and only if $D = 1$.

4.4 Hidden Markov models

4.4.1 Overview

In the previous section we discussed how a DTW base classifier models a writer's signature by computing the average distance between feature sets representative of a typical genuine signature and those of a questioned signature. In such a model, the order in which these feature vectors appear within the feature set is not considered.

HMMs adopt an alternative, stochastic approach by considering not only the feature vectors, but also the relationships between consecutive feature vectors in the feature set. In order to sensibly employ HMMs, it is therefore required that time-evolution exists between each pair of consecutive feature vectors. This concept of signature evolution is simulated by the grid-based feature vector construction process, as discussed in Section 3.4.5.

This section provides concise discussions regarding the HMM base classifiers developed in this study. An elaboration of key HMM concepts relevant to this study is provided in

Appendix B. Furthermore, for comprehensive discussions on the theory of HMMs and the applications thereof, the reader is referred to such works as Deller *et al.* (1987) and Rabiner (1989).

4.4.2 Notation

Given a sequence of discrete observations $\mathbf{O} = \{o_1, o_2, \dots, o_T\}$, the discrete observation HMM λ is characterised by the following key elements:

- N , the number of states in the model. The individual states are denoted by

$$S = \{s_1, s_2, \dots, s_N\}, \quad (4.11)$$

whilst the state at time t is denoted by q_t .

- M , the number of distinct observation symbols. The individual symbols are denoted by

$$V = \{v_1, v_2, \dots, v_M\}. \quad (4.12)$$

As the set of observation symbols corresponds to the output of the system being modelled, it is strictly required that $o_i \in V$ for $i = 1, 2, \dots, T$.

- The state transition probability distribution $\mathbf{A} = \{a_{ij}\}$, where

$$a_{ij} = P(q_{t+1} = s_j | q_t = s_i), \quad 1 \leq i, j \leq N. \quad (4.13)$$

As each element of \mathbf{A} represents a probability, the additional constraint

$$\sum_{j=1}^N a_{ij} = 1, \quad 1 \leq i \leq N \quad (4.14)$$

is imposed.

- The observation symbol probability distribution for state j , $\mathbf{B} = \{b_j(k)\}$, where

$$b_j(k) = P(o_t = v_k | q_t = s_j), \quad \begin{matrix} 1 \leq j \leq N \\ 1 \leq k \leq M \end{matrix} \quad (4.15)$$

- The initial state distribution $\pi = \{\pi_i\}$, where

$$\pi_i = P(q_1 = s_i), \quad 1 \leq i \leq N. \quad (4.16)$$

Similar to the state transition probability distribution, it is required that

$$\sum_{i=1}^N \pi_i = 1. \quad (4.17)$$

Since the model parameters N and M can be inferred from \mathbf{B} , the compact notation

$$\lambda = (\mathbf{A}, \mathbf{B}, \pi) \quad (4.18)$$

is commonly used to specify the complete parameter set for a discrete observation HMM.

4.4.3 HMM topology

If, apart from having to adhere to the rules of probability, no additional constraints are imposed on the model parameters π and \mathbf{A} , the HMM is referred to as *fully connected* or *ergodic*. Such an HMM has the property that every state can at any time be reached from every other state.

By imposing additional constraints on π and \mathbf{A} , however, it is possible to customise the HMM in order to better suit the nature of the data considered. Such a customised HMM is said to have a specific *topology*, as illustrated in Figure 4.3.

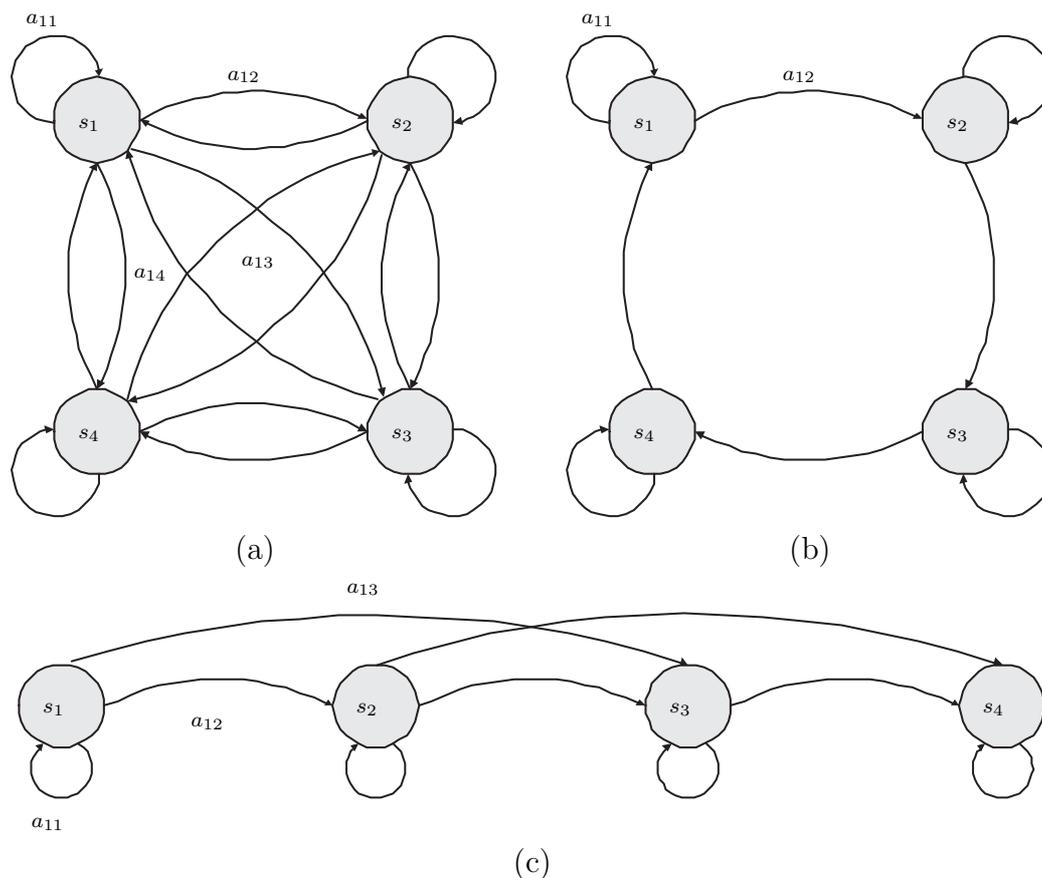


Figure 4.3: Examples of popular HMM topologies for $N = 4$. (a)-(b) Ergodic and ring-structured HMMs, respectively. (c) A left-right HMM with $l = 2$ forward links per state. Note that only the left-right model has a designated initial state.

One such topology, referred to as the *left-right* model (see Figure 4.3 (c)), has proved highly effective in applications where the modelling of handwriting or speech is involved. Since its underlying state sequence has the property that the state index remains non-decreasing, the left-right HMM model is especially well suited to model signals that evolve with time in one direction only. For this reason, left-right HMMs are commonly used to great effect for signature verification and recognition. The HMM base classifiers developed in this study each utilises a left-right HMM, mainly due to the fact that the writer-specific

signatures contained in Dolfing's data set (see Section 6.2) possess a similar baseline alignment. As a result, signature alignment in actual fact becomes a distinguishing feature. We therefore explicitly avoid rotation invariance, since its incorporation would only impede system performance.

From a mathematical perspective, a left-right HMM can be constructed by imposing two additional constraints on π and \mathbf{A} . Firstly, it is required that

$$\pi_i = \begin{cases} 1, & \text{if } i = 1 \\ 0, & \text{if } i \neq 1 \end{cases}, \quad (4.19)$$

since the state sequence must begin in state s_1 (and end in state s_N). Furthermore, the constraint

$$a_{ij} = 0, \quad j < i \quad (4.20)$$

ensures that state indices remain non-decreasing. Often, in order to prevent radical changes in the state index, an additional constraint is imposed on \mathbf{A} , that is

$$a_{ij} = 0, \quad j > i + l, \quad (4.21)$$

where l denotes the maximum number of (consecutive) states reachable from state s_i . This implies that, during any given state transition, up to $l - 1$ states may be skipped.

4.4.4 Model training

As was the case for the DTW-based model, each writer submits a set of K_ω genuine signatures for training, represented by

$$\{\mathbf{X}_1^{(\omega)}, \mathbf{X}_2^{(\omega)}, \dots, \mathbf{X}_{K_\omega}^{(\omega)}\}. \quad (4.22)$$

Before these patterns can be submitted to a discrete HMM, however, each set of feature vectors needs to be converted into a set of discrete observations. This is achieved by performing an appropriate VQ technique, as discussed in Section 3.5, yielding

$$\{\mathbf{O}_1^{(\omega)}, \mathbf{O}_2^{(\omega)}, \dots, \mathbf{O}_{K_\omega}^{(\omega)}\}. \quad (4.23)$$

The HMM parameter set is subsequently optimised using Viterbi re-estimation (see Section B.3), where the dissimilarity between an observation \mathbf{O} and model λ is given by

$$D(\mathbf{O}, \lambda) = -\ln P(\mathbf{O}|\lambda). \quad (4.24)$$

In addition, the writer-specific statistics μ_ω and σ_ω are determined from the training set by computing

$$\mu_\omega = \frac{1}{K_\omega} \sum_{i=1}^{K_\omega} D(\mathbf{O}_i^{(\omega)}, \lambda_\omega), \quad (4.25)$$

$$\sigma_\omega = \frac{1}{K_\omega - 1} \sum_{i=1}^{K_\omega} (D(\mathbf{O}_i^{(\omega)}, \lambda_\omega) - \mu_\omega)^2. \quad (4.26)$$

The writer-specific statistics, in conjunction with the trained HMM, represent the writer-dependent model, denoted by

$$M_\omega = (\lambda_\omega, \mu_\omega, \sigma_\omega). \quad (4.27)$$

This training process is performed for each of the Ω writers enrolled into the system database, yielding

$$\{M_1, M_2, \dots, M_\Omega\}, \quad (4.28)$$

thereby completing the HMM-based training phase.

4.4.5 Verification

When an HMM base classifier receives a questioned signature pattern \mathbf{X}_q and claim of ownership ω , it first needs to convert each of the T , d -dimensional feature vectors into a discrete symbol represented in the VQ codebook (see Section 3.5.1). This VQ process consequently produces the discrete observation sequence \mathbf{O}_q , which can be matched to the trained model M_ω by means of the Viterbi algorithm (see Section B.2). The resulting dissimilarity measure

$$\delta = D(\mathbf{O}_q, \lambda_\omega) \quad (4.29)$$

is subsequently mapped to the confidence interval $[0, 1]$ by applying Equation 4.1, yielding

$$s_c = N(\delta, \mu_\omega, \sigma_\omega). \quad (4.30)$$

Finally, a sliding global threshold $\tau \in [0, 1]$ is used to obtain a verification decision D , such that

$$D = \begin{cases} 1, & \text{if } s_c \geq \tau \\ 0, & \text{if } s_c < \tau \end{cases}. \quad (4.31)$$

The signature pattern \mathbf{X}_q is accepted as genuine if and only if $D = 1$.

4.5 Concluding remarks

In this chapter we introduced the DTW and HMM classification techniques utilised by the base classifiers developed in this study. We showed that each technique employs a fundamentally different approach to signature modelling. We also showed that, conceptually, the verification protocols associated with these models are identical. Although not proven in this study, it is known that an HMM-based approach to signature verification is generally superior to a DTW-based approach in terms of computational efficiency.

By employing different feature extraction techniques, as discussed in the previous chapter, in conjunction with different classification techniques, as discussed in this chapter, we obtain a set of fundamentally different base classifiers. Each of these base classifiers can constitute a signature verification system in its own right, as shown in Chapter 6 - where Dolfing's data set will be used to gauge the performance of each of the base classifiers separately.

The combined classifiers developed in this study, however, harness the efforts of several base classifiers, by employing a suitable classifier fusion strategy. A collection of such classifier fusion strategies is presented in the next chapter.

Chapter 5

Classifier Combination

“The whole is more than the sum of its parts.”
- Aristotle (384BC–322BC)

5.1 Introduction

The base classifiers, as discussed in Chapters 3 and 4, each combine the efforts of a single feature extraction technique with a single classification technique, in order to construct a signature model. As these classifiers are fundamentally different from each other, either on the basis of feature type or classification technique, one would expect to observe a gain in verification proficiency when combining the efforts of the resulting classifier ensemble.

In this chapter we discuss several techniques available for base classifier combination, based on either their associated confidence scores or verification decisions. Score-based fusion is briefly discussed in Section 5.2, whilst a discussion on decision-based fusion strategies is presented in Section 5.3. Most importantly, in Section 5.3.2 we discuss a classifier combination strategy based on ROC-curve analysis, wherein we also present the novel elitist MAROC-based classifier ensemble combination strategy proposed in this study.

5.2 Score-based fusion

Given a set of N confidence scores $\{s_1, s_2, \dots, s_N\}$ from N base classifiers, the *combined score* s^* can be calculated by utilising any number of score fusion techniques - the most popular of which include score *maximisation*, *minimisation* and *averaging*. The final decision D is subsequently obtained by imposing a decision threshold on s^* .

By employing score maximisation, for example, the combined classifier minimises only the FRR. Score minimisation, on the other hand, minimises only the FAR. As mentioned in Section 1.2.8, though, any decrease in the FRR inevitably results in an increased FAR, and vice versa. When employing either of these score fusion strategies, the appropriate strategy is dictated by the penalty¹ associated with a false acceptance or false rejection.

¹In the context of cheque processing, for example, a much higher penalty is generally associated with a false acceptance than with a false rejection. When considering especially large cheques, the clearance of a forged cheque is deemed unacceptable.

In order to achieve an overall improvement in classifier performance, the score averaging fusion strategy focusses on the minimisation of the AER. When employing SA, the combined score is calculated as

$$s^* = \frac{1}{N} \sum_{i=1}^N s_i. \quad (5.1)$$

Although the impact on system performance is generally less dramatic when employing SA, a consistent improvement in verification accuracy can be expected. Since each base classifier influences the final decision, though, one should be mindful not to include any relatively inaccurate base classifiers, thereby diminishing the overall improvement in performance.

5.3 Decision-based fusion

Another popular approach to classifier ensemble combination involves obtaining a verification decision from each base classifier individually, after which a suitable decision fusion strategy is applied. In such cases, the decision made by each base classifier involved in the combination process is referred to as a *partial decision*, denoted by $P \in \{0, 1\}$, where $P = 1$ is associated with an acceptance, whilst $P = 0$ indicates a rejection.

5.3.1 Voting

Given a set of N partial decisions $\{P_1, P_2, \dots, P_N\}$ from N base classifiers, a voting strategy may be applied in order to obtain the final decision D , by computing

$$D = \begin{cases} 1, & \text{if } \sum_{i=1}^N P_i \geq k \\ 0, & \text{if } \sum_{i=1}^N P_i < k \end{cases}. \quad (5.2)$$

From Equation 5.2 it is clear that k represents the number of votes required to obtain a positive result. Although $k \in [1, N]$ can be chosen arbitrarily, one should bear in mind that the combined classifier's tendency to accept a questioned signature, and therefore also the corresponding FAR, is inversely proportional to k .

In general, the *majority vote* rule is employed, where the optimal number of votes required is defined as

$$k = \left\lceil \frac{N+1}{2} \right\rceil, \quad (5.3)$$

where $\lceil \cdot \rceil$ denotes the ceiling operator. As is the case with SA, the inclusion of a relatively inaccurate base classifier may also impede the overall performance of the combined classifier when the MV rule is employed, but to a much lesser extent.

5.3.2 ROC-based combination

In this section we present the classifier combination strategy, originally proposed by Haker *et al.* (2005), which is based on the maximum likelihood analysis of ROC-curves. We explain how this method, henceforth referred to as *Haker's algorithm*, can be used

to combine any pair of continuous classifiers. We also discuss how Haker's algorithm can be incorporated into the design of the novel classifier ensemble combination strategy proposed in this study.

Haker's algorithm is geared towards the optimal combination of decisions made by two discrete classifiers C_A and C_B . The algorithm assumes that the individual performances of C_A and C_B have already been experimentally determined and recorded as points in ROC-space. In essence, Haker's algorithm combines the decisions made by C_A and C_B using one of four different combination rules:

- S_A , where the combined decision is represented by the decision made by C_A .
- S_B , where the combined decision is represented by the decision made by C_B .
- $S_{A\&B}$, where the popular AND rule is employed to combine the decisions made by C_A and C_B . Under the AND rule, a questioned signature is accepted if and only if both classifiers accept it individually.
- $S_{A|B}$, where the popular OR rule is employed to combine the decisions made by C_A and C_B . Under the OR rule, a questioned signature is accepted if at least one of the two classifiers accepts it individually.

Given a classifier pair, the appropriate combination rule is determined using maximum likelihood estimation (MLE), based on the performance statistics of said classifiers in ROC-space.

It is important to note that Haker's algorithm has been shown to outperform strategies which employ only one of the combination rules presented above, regardless of the ROC-statistics of the individual classifiers. In the next section, we discuss Haker's algorithm in more detail.

Haker's algorithm

Recall from Section 1.2.8 that, for a continuous classifier C_i , each point $(f_i^+(\tau), t_i^+(\tau))$ in ROC-space denotes the FPR-TPR pair associated with a specific decision threshold τ , thereby representing the discrete classifier $C_i(\tau)$.

The ROC-space combination strategy utilised by Haker's algorithm involves using maximum likelihood analysis to obtain a combination rule for two ROC operating points $(f_A^+(\tau_A), t_A^+(\tau_A))$ and $(f_B^+(\tau_B), t_B^+(\tau_B))$. Given a questioned pattern, each of the corresponding classifiers produces either a positive or negative output, resulting in four possible cases. For each of these cases, Table 5.1 expresses the MLE of the unknown truth $T \in \{0, 1\}$, where $T = 1$ is associated with a genuine signature, whilst $T = 0$ indicates a forgery.

Each inequality in the rightmost column of Table 5.1 represents a logical expression and therefore evaluates to a single binary output - the MLE for T . For example, assuming conditional independence, it follows that

$$\begin{aligned} P(C_A = 1, C_B = 1|T = 1) &= P(C_A = 1|T = 1)P(C_B = 1|T = 1) \\ &= t_A^+ t_B^+. \end{aligned} \tag{5.4}$$

C_A	C_B	Combination MLE for truth T
1	1	$P(C_A = 1, C_B = 1 T = 1) \geq P(C_A = 1, C_B = 1 T = 0)$
1	0	$P(C_A = 1, C_B = 0 T = 1) \geq P(C_A = 1, C_B = 0 T = 0)$
0	1	$P(C_A = 0, C_B = 1 T = 1) \geq P(C_A = 0, C_B = 1 T = 0)$
0	0	$P(C_A = 0, C_B = 0 T = 1) \geq P(C_A = 0, C_B = 0 T = 0)$

Table 5.1: Maximum likelihood combination of the binary output of classifiers C_A and C_B .

C_A	C_B	Combination MLE for truth T
1	1	$t_A^+ t_B^+ \geq f_A^+ f_B^+$
1	0	$t_A^+(1 - t_B^+) \geq f_A^+(1 - f_B^+)$
0	1	$(1 - t_A^+)t_B^+ \geq (1 - f_A^+)f_B^+$
0	0	$(1 - t_A^+)(1 - t_B^+) \geq (1 - f_A^+)(1 - f_B^+)$

Table 5.2: Maximum likelihood combination, in terms of the associated TPR and FPR, of the binary output of classifiers C_A and C_B .

Following a similar argument for the remaining inequalities, the entries in Table 5.2 are obtained.

From the definition of the TPR and FPR, it should be clear that, for any classifier C_i , whenever the TPR t_i^+ is less than the corresponding FPR f_i^+ , a simple negation of the classifier decision will result in (f_i^+, t_i^+) being reflected about the line TPR=FPR in ROC-space. The assumption $t_i^+ \geq f_i^+$ is therefore reasonable, from which it follows that $t_A^+ t_B^+ \geq f_A^+ f_B^+$ and $(1 - t_A^+)(1 - t_B^+) \leq (1 - f_A^+)(1 - f_B^+)$. Consequently, whenever C_A and C_B are in agreement, their common output denotes the MLE estimate of T . Therefore, only the two middle rows of Table 5.2 need to be determined, which results in four possible MLE combination schemes. These combination schemes are provided in Table 5.3.

Furthermore, Table 5.4 shows how the *predicted* TPR and FPR associated with each combination scheme is obtained. These predicted combined performances are based on the assumption that the base classifiers make *independent* decisions. In practise, however, this is rarely the case. The signature modelling strategies employed by the DTW and HMM base classifiers developed in this study, for example, although considering completely different approaches, are both derived from dynamic programming techniques. It is therefore suspected that the combination of these base classifiers may not facilitate the optimal performance of a ROC-based combined classifier.

In order to address this issue, the base classifiers developed in this study, when determining the optimal combination of each discrete classifier pair, do not rely on a predicted performance. Instead, the appropriate combination rule, as determined from the discrete classifiers' ROC-statistics, is employed on actual data, thereby producing the realistically attainable performance of such a discrete classifier combination. Although the generation of this *estimated* performance is computationally far more expensive than that of the predicted performance, the potentially significant impact of the independence criterion is minimised. Nevertheless, the predicted performance may still be used in order to determine an upper bound for the expected performance. Should the base classifiers prove to be sufficiently independent, this upper bound remains realistically achievable.

The pseudo-code for Haker's algorithm is provided in Appendix C. When utilising this

C_A	C_B	$S_{A\&B}$	S_A	S_B	$S_{A B}$
1	1	1	1	1	1
1	0	0	1	0	1
0	1	0	0	1	1
0	0	0	0	0	0

Table 5.3: Combination rules considered for the output of classifiers C_A and C_B .

Scheme	TPR	FPR
$S_{A\&B}$	$t_A^+ t_B^+$	$f_A^+ f_B^+$
S_A	t_A^+	f_A^+
S_B	t_B^+	f_B^+
$S_{A B}$	$t_A^+ + t_B^+ - t_A^+ t_B^+$	$f_A^+ + f_B^+ - f_A^+ f_B^+$

Table 5.4: Calculation of the predicted combined TPR and FPR associated with the set of MLE combination rules.

algorithm to combine a pair of continuous classifiers C_A and C_B , each of the J threshold-specific classifiers contained in C_A is combined with each of the K threshold-specific classifiers contained in C_B , using combination rule S_A , S_B , $S_{A\&B}$, or $S_{A|B}$. This process consequently yields JK additional points in ROC-space. These additional ROC-points, however, each represents two decision thresholds τ_A and τ_B coupled with the appropriate MLE combination rule. In order to reproduce the predicted/estimated performance of such a discrete classifier combination in practise, the combined classifier employs a set of modified decision rules, presented in Table 5.5. The decision rules associated with schemes S_A and S_B are identical to those employed by a single classifier (see Sections 4.3.3 and 4.4.5). The decision rule associated with scheme $S_{A\&B}$ evaluates to $D = 1$ if and only if $s_A \geq \tau_A$ AND $s_B \geq \tau_B$, whilst the decision rule associated with scheme $S_{A|B}$ evaluates to $D = 1$ if $s_A \geq \tau_A$ OR $s_B \geq \tau_B$.

Scheme	Decision rule
$S_{A\&B}$	$\min[s_A - \tau_A, s_B - \tau_B] \geq 0$
S_A	$s_A \geq \tau_A$
S_B	$s_B \geq \tau_B$
$S_{A B}$	$\max[s_A - \tau_A, s_B - \tau_B] \geq 0$

Table 5.5: Decision rules considered by the combined classifier, given the base classifier scores s_A and s_B .

Furthermore, since the goal of classifier combination is the construction of a superior hybrid classifier, a single ROC-curve is *selected* from the pool of JK discrete classifier combinations. From the definition of the TPR and FPR, it should be clear that the optimal ROC-curve obtainable from such a pool of classifiers is represented by the upper-left boundary of the convex hull of these JK points in ROC-space. This MAROC-curve, as illustrated in Figure 5.1, therefore contains no sub-optimal discrete base classifiers

or sub-optimal combined classifiers, and is representative of the combined classifier's maximum attainable performance.

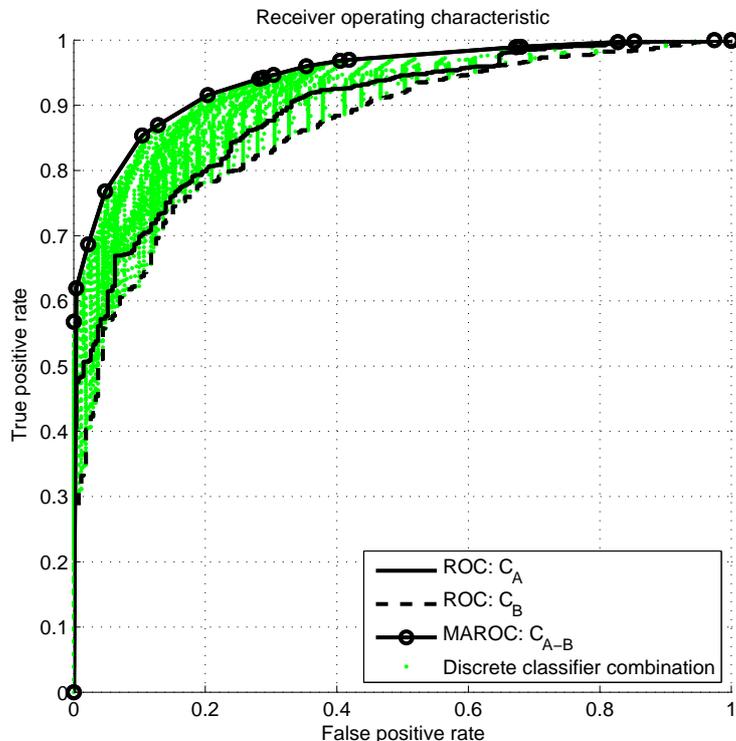


Figure 5.1: Example of a typical MAROC-curve associated with continuous classifiers C_A and C_B , comprised of $J \approx 1200$ and $K \approx 1200$ discrete classifiers, respectively. Note that, for illustrative purposes, only 0.25% of the discrete classifier combinations obtained from Haker's algorithm are presented in the figure. Also note that the resulting MAROC-curve, obtained by considering all JK possible discrete classifier combinations, is completely specified by approximately 20 ROC-points.

Elitist MAROC-based classifier ensemble combination

In theory, Haker's algorithm can be used to combine any number of continuous classifiers. This is achieved by simply including an additional nested level of combinations for each additional continuous classifier. In practise, however, the resulting computational cost will render this combination strategy, even when a relatively small number of continuous classifiers is considered, virtually intractable.

Consider, for example, the set of eight base classifiers developed in this study, each occupying in excess of 1000 points in ROC-space. Of course, as the number of discrete classifiers comprising any continuous classifier equals the number of distinct decision thresholds utilised, this number may be predefined arbitrarily. In order to ensure sufficient

continuity in an ROC-curve, however, it is recommended that a large number of decision thresholds be employed. The base classifiers developed in this study each typically utilises 1100–1300 decision thresholds. For the purposes of this complexity analysis, however, the assumption is made that each utilises exactly 1200 thresholds.

Let $K = 1200$ and $N = 8$ denote the length of each ROC-curve and the number of ROC-curves considered for combination, respectively. The utilisation of Haker’s algorithm to combine these N ROC-curves would therefore result in the computation of

$$K^N \approx 4.3 \times 10^{24} \quad (5.5)$$

discrete classifier combinations. Even in a computing environment capable of performing one million combinations per second, this calculation will take 136,253,000,000 years to execute!

Since the calculation of all K^N possible discrete classifier combinations is clearly not a computationally viable option, given currently available computing systems, one is forced to consider a diminished solution space, resulting in a potentially sub-optimal solution. The primary objective therefore becomes the selection of a suitable subset of discrete classifiers for inclusion into the classifier ensemble considered for combination. Recent developments include the *dynamic overproduce-and-choose* strategy proposed by Dos Santos *et al.* (2008), the *hybrid genetic algorithm* proposed by Kim and Oh (2008) and the *adaptive splitting and selection method* proposed by Jackowski and Wozniak (2009). These efforts represent only a small percentage of the strategies proposed in the literature, but illustrate that the problem of classifier ensemble selection remains a significant area of research.

In this study we propose a novel classifier ensemble combination strategy. Whilst utilising Haker’s original algorithm, we propose the iterative combination of ROC-curves with MAROC-curves, as an alternative to the nested combination of ROC-curves, in order to achieve classifier ensemble combination. By considering MAROC-curves as suitable candidates for classifier combination, this strategy greatly reduces the overall computational cost.

In this strategy, the process of combining N ROC-curves is achieved in $N - 1$ separate iterations. The first iteration involves combining two K -point ROC-curves using Haker’s algorithm, consequently producing K^2 discrete classifier combinations. In each of the subsequent $N - 2$ iterations, however, one of the remaining K -point ROC-curves is combined, again using Haker’s algorithm, with the L -point MAROC-curve obtained during the previous iteration, consequently producing KL discrete classifier combinations. Letting $L = 20$ denote the length of a typical MAROC-curve (see Figure 5.1), the combination of N ROC-curves consequently results in the computation of

$$K^2 + (N - 2)(KL) = 1.58 \times 10^6 \quad (5.6)$$

discrete classifier combinations. Using the same computing environment described earlier, the execution time is reduced to 1.58 seconds!

As clearly illustrated by the previous example, this strategy definitely satisfies the requirement of computational efficiency. The only remaining question is therefore whether the resulting solution adequately satisfies the requirement regarding optimality, as only

$$\frac{K^2 + (N - 2)(KL)}{K^N} \times 100 = 3.68 \times 10^{-19}\% \quad (5.7)$$

of all possible combinations are considered.

The issue of optimality is addressed in a surprisingly simple and elegant manner. As mentioned in the previous section, a MAROC-curve represents the convex hull of all discrete classifiers generated during a single iteration. We therefore know that all discrete classifier combinations not represented in the MAROC-curve are, within the context of a single iteration, sub-optimal. Furthermore, during combination, the set of N ROC-curves are presented to the algorithm in a specific order, namely from lowest to highest corresponding AUC-measure. In this manner, it is ensured that the majority of discarded classifier combinations are associated with the relatively inaccurate classifiers.

This elitist approach therefore maximises the likelihood that the sub-optimal solution obtained is not significantly inferior to the optimal solution. In addition, the inclusion of a relatively inaccurate base classifier would not directly reduce overall system performance, as its contribution is simply discarded during combination.

5.4 Concluding remarks

In Chapters 3 and 4 we discussed the design and implementation of a typical base classifier developed in this study. In this chapter we discussed how the verification performance achieved by such a base classifier can be improved upon, by constructing a superior combined classifier which utilises the efforts of several base classifiers. Both score-based and decision-based fusion strategies are considered.

In the next chapter, the verification performance achieved by the individual base classifiers, as well as the combined classifiers, is experimentally evaluated.

Chapter 6

Experiments

“It doesn’t matter how beautiful your theory is. If it doesn’t agree with experiment, it’s wrong.”

- Richard Feynman (1918–1988)

6.1 Introduction

In Chapters 3–5 we established the theoretical background required to develop several candidates for an efficient off-line signature verification system. Each such candidate may utilise either a single DTW or HMM base classifier, or combine the efforts of an ensemble constructed from several base classifiers.

In this chapter, we optimise and evaluate each of these verification system candidates experimentally. In Section 6.2 we present the data set used for experimentation, whilst Section 6.3 discusses the experimental protocol. Results obtained for each of the evaluated systems are reported in Section 6.4. In Section 6.5 we experimentally confirm the contributions made in this study. In Section 6.6 the results are compared to the results reported for previous systems proposed in the literature.

6.2 Data

The signature database considered in this study contains 4837 signatures obtained from 51 different writers. This data set, originally captured on-line, is referred to as *Dolfing’s data set*, as it was originally utilised by Dolfing (1998) for the purpose of developing an on-line signature verification system. Dolfing’s original data set has since been converted into an off-line data set, thereby rendering it suitable for the evaluation of the classifiers developed in this study. For a detailed discussion regarding the conversion algorithm, the reader is referred to Coetzer (2005).

Since Dolfing’s data set was originally captured on-line, the resulting off-line representation can be viewed as an *ideal* data set, in the sense that none of the signature images suffer from the presence of background noise. In addition, each signature image contained in the data set has a uniform stroke width of approximately 5 pixels.

During the acquisition of Dolfing’s data set, each of the 51 writers submitted 30 signature samples, contributing a total of 1530 genuine signatures. A total of 3000 amateur

skilled forgeries and 307 professional skilled forgeries were subsequently collected. Only the set of amateur skilled forgeries are considered in this study.

In order to gain further insight into the nature of Dolfing’s data set, we present a collection of samples, one genuine signature for each of the 51 writers, in Figure 6.1. From these samples, it is clear that the signatures comprising Dolfing’s data set represent a wide variety of handwriting styles, as well as greatly varying levels of intricacy regarding signature design.



Figure 6.1: Typical examples of signature images contained in Dolfing’s data set. Each image represents a genuine signature belonging to one of the 51 enrolled writers. Note that all the signatures presented have the same uniform stroke width of 5 pixels. This property is not clearly illustrated in the figure, however, since the scale of the images differ.

6.3 Experimental protocol

In order to design the most sensible experimental protocol, it is necessary to create a realistic analogy between the model evaluation process and the practical deployment of a typical signature verification system.

Consider, for example, the processing of cheques within the banking environment. In practise, prior to system deployment, signatures are gathered from a designated group of individuals, commonly referred to as *guinea pigs*. These guinea pigs are considered

representative of a potentially unlimited population. As a result, only guinea pig signatures are used to optimise the model configuration, within a controlled environment. Once deployed, the system constructs a writer-dependent signature model for each enrolled *client*, using the optimal model configuration as obtained from the set of guinea pig writers. Each new client is required to supply a small set of genuine signature samples for model training.

In order to simulate this scenario in our experimental setup, as illustrated in Figure 6.2, Dolfing’s data set is partitioned into two disjoint data sets. This partitioning is discussed in the next section. The use of separate data sets for model optimisation (see Section 6.3.2) and evaluation (see Section 6.3.3) therefore lends a much greater degree of credibility to the results reported in this chapter, since the concepts of guinea pigs and clients are sufficiently incorporated.

6.3.1 Data set partitioning

Prior to experimentation, it is required that Dolfing’s data set be partitioned into two disjoint data sets. This partitioning ensures that different data is used for model optimisation and model evaluation, thereby avoiding potential *over-fitting* of the model parameters. This requirement is of paramount importance, as the evaluation of an over-fitted model is not only associated with a misrepresentation in verification proficiency, but also a diminished generalisation potential. These topics are discussed further in Section 6.3.3.

The composition of these data set partitions, referred to as the *optimisation set* (OS) and *evaluation set* (ES), is summarised in Table 6.1. It should be clear that the optimisation set is analogous to a set of guinea pig signatures, whilst the evaluation set is analogous to a set of client signatures, as discussed in the previous section.

Dolfing’s data set		Genuine	Forgery
Optimisation set (34 writers)	T_O	510	-
	O_B	272	1005
	O_C	238	1005
Evaluation set (17 writers)	T_E	255	-
	E	255	990

Table 6.1: The number of signatures used in the partitioning of Dolfing’s data set into a separate optimisation set and evaluation set. The specific tasks associated with the subsets T_O , O_B , O_C , T_E and E are discussed in Sections 6.3.2 and 6.3.3.

Note that the signatures allocated to each partition is further subdivided into the disjoint sets T_O , O_B , O_C , T_E and E . Each of these sets facilitates a specific task during model optimisation or evaluation, and is discussed further in Sections 6.3.2 and 6.3.3.

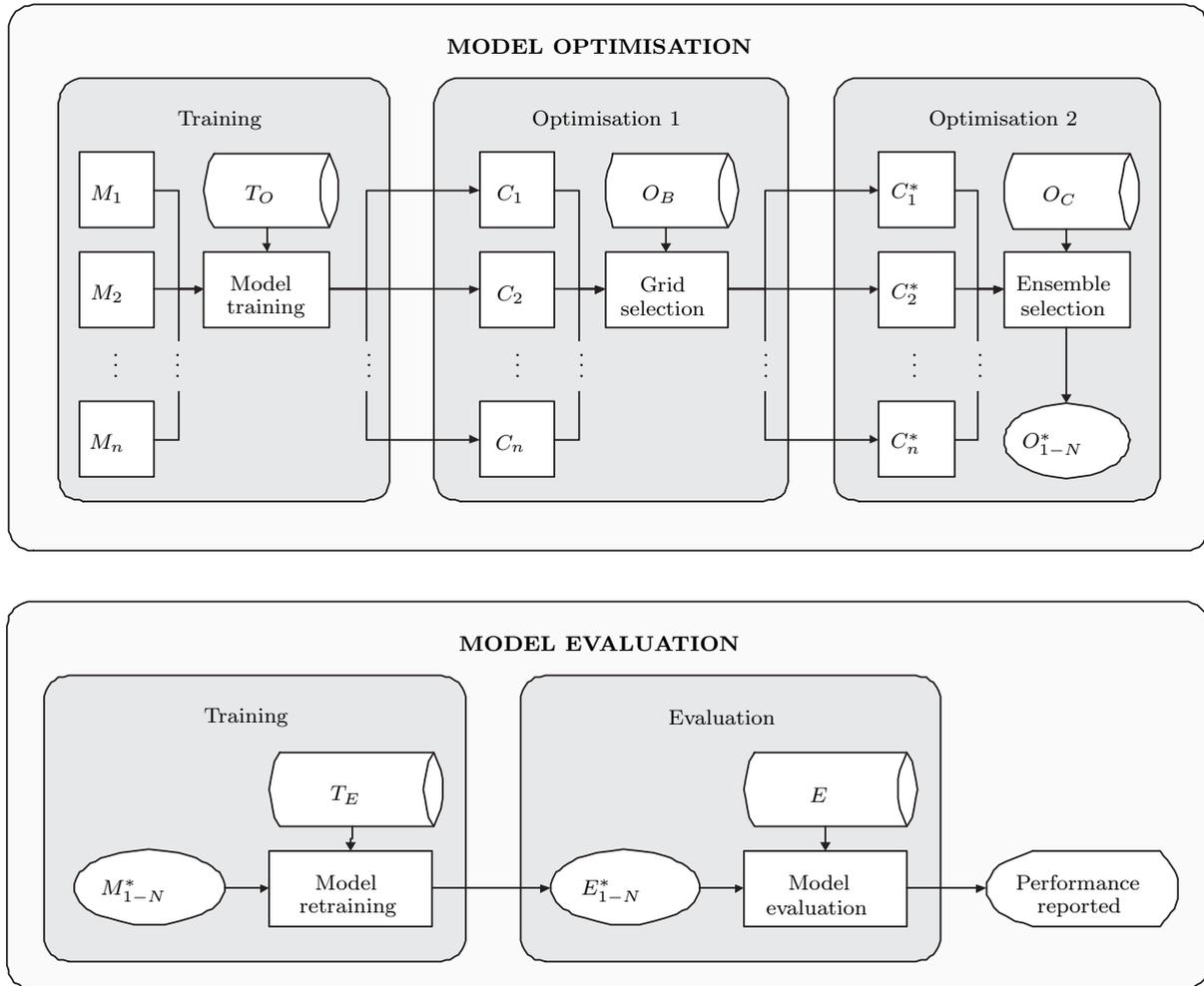


Figure 6.2: Schematic representation of the experimental protocol considered in this study. Note that each process utilises a disjoint subset of signature data. Each entity M_i , C_i and C_i^* denotes an untrained, trained and optimised base classifier, respectively. The entities O_{1-N}^* and E_{1-N}^* denote the optimal combined classifier, as obtained from the optimisation set and evaluation set, respectively, whilst M_{1-N}^* denotes the collection of untrained base classifiers associated with the optimal classifier ensemble. Note that n denotes the number of available base classifiers, whilst N denotes the number of base classifiers utilised in the optimal classifier ensemble. Detailed discussions on model optimisation and model evaluation are presented in Sections 6.3.2 and 6.3.3, respectively.

6.3.2 Model optimisation

Base classifiers

The aim of model optimisation is to determine the optimal configuration for each base classifier considered. In general, this configuration refers to the set of internal model parameters, or *hyper-parameters*. In the case of discrete HMM base classifiers, these parameters include the number of states, the number of allotted forward links per state, as well as the VQ codebook size. For DTW base classifiers, only the bandwidth is considered to be adjustable. For any classifier utilising a grid-based feature extraction technique, the segmentation grid resolution is also considered an adjustable parameter.

In this study, however, since we are more interested in the significance of the flexible grid segmentation strategy, only the grid configuration is optimised, whilst all other hyper-parameters are held fixed. These parameter values, presented in Table 6.2, are determined experimentally prior to model optimisation. A relatively small set of values, similar to values that proved to be optimal in previous studies reported in the literature, are considered.

Configuration	Parameter	Section	Value
Adaptable grid	l_x	3.3.2	2
	l_y	3.3.2	2
VQ codebook	K (PD,PS)	3.5.1	130
	K (GCD,ORT)	3.5.1	125
DTW	H_{vec}	4.3.1	$0.25d$
HMM	N	4.4.2	$0.25T$
	l	4.4.3	2
	p_{min}	B.3.3	10^{-8}

Table 6.2: The set of fixed model hyper-parameter values. These values are fixed prior to model optimisation, thereby greatly decreasing the number of model configurations to consider. Note that reference is made to the section in which each parameter is introduced. Also note that d and T refer to the feature vector dimension and observation sequence length, respectively, as introduced in Section 1.2.1.

The set of flexible grid configurations considered during optimisation are presented in Table 6.3. Note that the special case $F_x = F_y = 0$ is included, thereby ensuring that rigid segmentation grids are also considered during optimisation.

Parameter	Value set
M	{10,20}
N	{20,40}
F_x	{0,0.25, ..., 2}
F_y	{0,0.25, ..., 2}

Table 6.3: The set of flexible grid parameter values considered for model optimisation. Since $F_x = F_y = 0$ is included, the combination of these parameter values results in a total of 4 rigid segmentation grids and 320 flexible segmentation grids considered per base classifier.

During base classifier optimisation, only the optimisation set is used. Each model M_i is trained using signature patterns contained in T_O , consequently yielding the trained base classifier C_i . Model performance is determined by testing C_i on signature patterns contained in O_B . After testing each grid configuration considered, only the optimal base classifier C_i^* is retained.

The performance measures yielded during the optimisation process may only be used to assess the model configuration. Using these measures to denote overall verification proficiency constitutes an over-fitted depiction of model performance.

Combined classifiers

When constructing a combined classifier, each base classifier C_i^* has already been optimised with respect to its grid configuration. No further alterations are therefore required with regard to base classifier model configurations. As mentioned in Sections 5.2 and 5.3, however, the inclusion of a relatively inaccurate base classifier into the classifier ensemble can negatively impact the performance of the combined classifier.

The process of finding the most proficient combined classifier therefore only entails the task of selecting the most proficient ensemble of base classifiers, given a certain fusion strategy, where each ensemble is constructed from models already trained on T_O and optimised on O_B . The performance of each combined classifier C_{1-N}^* is then determined using signatures contained in O_C . After assessing each ensemble selection considered, only the optimal combined classifier O_{1-N}^* is retained for evaluation.

6.3.3 Model evaluation

During model evaluation, only the evaluation set is used. The models comprising O_{1-N}^* , however, belong to the set of writers contained in the optimisation set. As a result, the corresponding models M_{1-N}^* are retrained using signature patterns contained in T_E , yielding the optimal combined classifier E_{1-N}^* . Verification proficiency is subsequently evaluated using signature patterns contained in E .

It should be clear that, for the purposes of base classifier evaluation, an identical protocol can be used. In such cases, a *trivial ensemble*, composed of a single base classifier, is presented for evaluation.

System performance is gauged using the AUC and EER measures. In addition, the generalisation potential of each classifier is measured by introducing the *generalisation error* ϵ , defined as

$$\epsilon = \text{AUC}_O - \text{AUC}_E, \quad (6.1)$$

where AUC_O and AUC_E denote the AUC achieved using the optimisation set and evaluation set, respectively. By using an evaluation set completely disjoint from the optimisation set, an objective depiction of performance regarding both model accuracy and generalisation potential is obtained, since the impact of model over-fitting is completely avoided.

6.3.4 Cross-validation

The model optimisation and evaluation processes, as discussed in the previous sections, convincingly address the issue of model over-fitting. The use of separate optimisation and evaluation data sets, however, gives rise to another important issue regarding the credibility of results reported.

Consider, for example, the varied levels of intricacy displayed by the writers in Dolfig’s data set (see Figure 6.1). It should logically be expected that a relatively simple signature design is easier to reproduce than a considerably more complex one. Throughout the literature, writers possessing easily reproducible signatures are often referred to as *sheep*. In contrast, a writer whose signature is relatively difficult to reproduce is referred to as a *wolf*.

The data set partitioning method discussed in Section 6.3.1, however, only takes into account the number of writers contained in each partition - not the nature of signatures allocated. The optimisation set may, for example, contain mostly wolves, whilst the evaluation set contains mostly sheep. This would lead to unrealistically high system performance. Conversely, a system may be optimised using mostly sheep and subsequently evaluated using mostly wolves. Although relatively poor results are expected, this will not be a true reflection of the system performance.

In order to address this issue, experiments conducted in this study perform model optimisation and evaluation based on the *stratified*¹ *k-fold cross-validation*² principle, as illustrated in Figure 6.3. Since Dolfig’s data set contains 51 writers, the classifiers developed in this study are evaluated using $k = 3$ folds. Also, since each fold constitutes an independent experiment, the performance of each classifier is represented by μ_{AUC} , μ_{EER} and μ_ϵ , denoting the average performance achieved over all the folds. Although it is common practise to also report the standard deviation of the performance achieved, this measure is deemed redundant within the context of a 3-fold protocol.

By employing the cross-validation strategy, the results reported in this study are therefore obtained using all the writers contained in Dolfig’s data set, thereby avoiding the experimental influence of sheep and wolves. As a result, the performance reported in the next section is not only objective, but also comprehensive.

6.4 Results

In this section we present the results obtained by optimising and evaluating the set of base classifiers and combined classifiers experimentally, using the protocol discussed in Section 6.3.

Note that all the tabulated results indicate the performance achieved during each separate fold, in addition to the corresponding average performance. All graphical results, however, denote only the average ROC-based performance. The ROC-curves presented are obtained by taking the average TPR and FPR achieved across all the folds, for each distinct decision threshold τ .

¹Stratified cross-validation requires that a similar distribution of positive and negative test patterns is shared between the original data set and each of its partitions.

²The subsets O_B and O_C , as contained in the optimisation set, can also constitute an independent data set, referred to as the *validation set* throughout the literature.

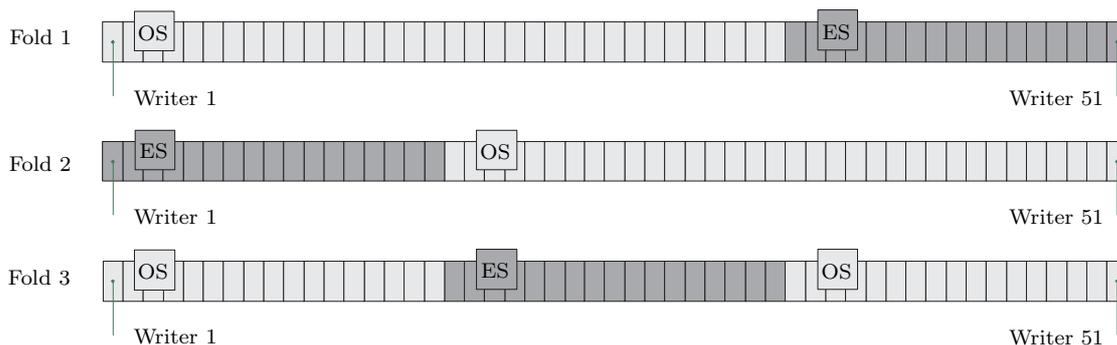


Figure 6.3: The 3-fold cross-validation procedure considered in this study. Each rectangular segment, within the context of a single fold, represents the signature set belonging to one of the 51 writers contained in Dolfig’s data set. As the fold index increases, the optimisation set and evaluation set are shifted 17 writers to the right, thereby considering all 51 writers for evaluation over the 3 folds.

6.4.1 Base classifiers

DTW base classifiers

The optimisation of the grid configurations used by the DTW base classifiers, as discussed in Section 6.3.2, yields the set of optimal segmentation grids presented in Table 6.4.

Optimisation set				
DTW	Feature			
	PD	GCD	ORT	PS
Fold 1	$20_0 \times 20_{1.25}$	$10_1 \times 20_{0.5}$	$10_1 \times 20_{0.5}$	$20_{0.25} \times 20_{0.75}$
Fold 2	$20_{1.75} \times 20_{0.75}$	$10_{0.75} \times 40_1$	$20_{0.75} \times 20_1$	$10_{0.25} \times 20_{0.75}$
Fold 3	$20_{1.5} \times 20_{0.75}$	$10_{0.75} \times 40_1$	$20_{0.75} \times 20_1$	$10_{0.25} \times 20_{0.75}$

Table 6.4: Optimal grid configurations obtained for the set of DTW base classifiers.

It is clear from Table 6.4 that there is no one specific optimal grid configuration for any of the features considered, thereby reiterating the importance of model optimisation within the context of the deployment environment. It is also important to note that, for all features considered and across all folds, none of the optimal segmentation grids are rigid, thereby emphasising the significance of the flexible grid-based feature extraction technique proposed in this study.

The AUC and EER measures achieved during model optimisation and model evaluation are presented in Table 6.5. The performance achieved is also illustrated in Figure 6.4.

Similar results are obtained when utilising the pixel density, gravity centre distance and orientation features. It is clear that the classifier utilising the predominant slant feature significantly outperforms its peers, in terms of both verification accuracy and generalisation potential, as presented in Table 6.6.

Optimisation set					Evaluation set				
DTW	Feature				DTW	Feature			
	PD	GCD	ORT	PS		PD	GCD	ORT	PS
AUC (%)					AUC (%)				
Fold 1	90.74	90.87	91.30	93.34	Fold 1	90.68	90.15	87.21	91.55
Fold 2	92.54	91.48	90.14	92.41	Fold 2	86.48	88.41	92.78	93.08
Fold 3	90.08	90.70	92.73	94.17	Fold 3	92.55	91.39	88.88	92.43
μ_{AUC}	91.12	91.02	91.39	93.31	μ_{AUC}	89.90	89.98	89.62	92.35
EER (%)					EER (%)				
Fold 1	17.27	16.99	16.99	13.28	Fold 1	16.87	18.41	20.75	15.67
Fold 2	15.51	16.56	18.41	14.71	Fold 2	22.35	19.95	14.90	13.73
Fold 3	18.03	17.22	15.47	12.81	Fold 3	15.67	16.07	20.00	13.33
μ_{EER}	16.94	16.92	16.96	13.60	μ_{EER}	18.30	18.14	18.55	14.24

Table 6.5: Results obtained for the set of DTW base classifiers.

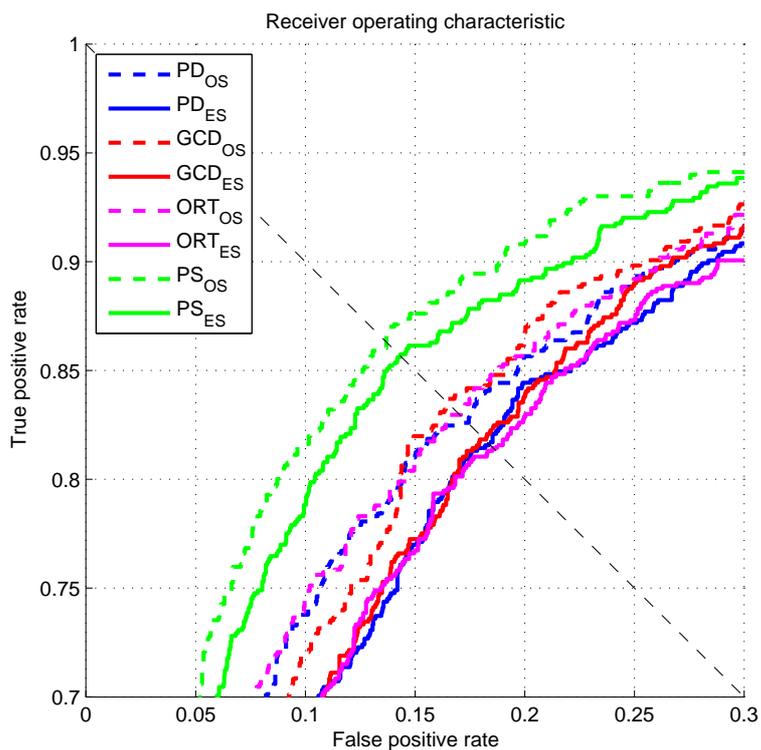


Figure 6.4: ROC-based performance achieved by the set of DTW base classifiers, using both the optimisation set and the evaluation set.

Dolfing's data set				
Feature	PD	GCD	ORT	PS
μ_ϵ (%)	1.22	1.04	1.77	0.96

Table 6.6: Generalisation errors obtained for the set of DTW base classifiers, using both the optimisation set and the evaluation set.

From Table 6.6 we deduce that each DTW base classifier is able to sufficiently reproduce its verification performance, given a set of signatures outside the optimisation set. The generalisation potential of each classifier is also clearly illustrated by the ROC-curves (see Figure 6.4) obtained during optimisation and evaluation. For each base classifier, the generalisation error is equivalent to the area between said ROC-curves.

HMM base classifiers

The optimisation of the grid configuration used by the HMM base classifiers, as discussed in Section 6.3.2, yields the set of optimal segmentation grids presented in Table 6.7.

Optimisation set				
HMM	Feature			
	PD	GCD	ORT	PS
Fold 1	$10_{0.75} \times 40_{0.75}$	$20_0 \times 40_1$	$10_{0.25} \times 40_1$	$20_0 \times 40_{1.25}$
Fold 2	$10_{0.75} \times 40_1$	$10_{0.25} \times 40_{0.25}$	$10_{0.5} \times 40_{0.5}$	$20_{0.5} \times 40_0$
Fold 3	$10_{0.5} \times 40_{0.5}$	$10_{0.25} \times 40_{0.75}$	$10_{1.25} \times 40_1$	$20_1 \times 40_1$

Table 6.7: Optimal grid configurations obtained for the set of HMM base classifiers.

As is the case with the set of DTW base classifiers, no single optimal grid configuration is identified for any of the features considered. Also, no optimal grid configuration utilises a rigid segmentation grid. It is worth noting that each optimal grid configuration, in the context of an HMM base classifier, contains 40 columns - which is the maximum number considered. This was expected, as it is associated with the maximum number of HMM states considered. It has been confirmed experimentally, however, using the feature extraction techniques proposed in this study, that any further increase in the number of grid columns does not result in a further improvement in classifier performance.

The AUC and EER measures achieved during model optimisation and model evaluation are presented in Table 6.8. The performance is also illustrated in Figure 6.5.

The set of HMM base classifiers achieves comparable results, except for the classifier utilising the pixel density feature, which significantly outperforms those utilising the gravity centre distance, orientation and predominant slant features. On average, the HMM base classifiers outperform their DTW-based counterparts. This was expected, since these classifiers incorporate a greater level of complexity into their writer-dependent signature models, as discussed in Section 4.4.1. The generalisation error of each HMM base classifier is presented in Table 6.9.

When compared to the results presented in Table 6.6, a slight improvement in the generalisation potential of the classifiers is evident. From Table 6.9 we deduce that each

Optimisation set					Evaluation set				
HMM	Feature				HMM	Feature			
	PD	GCD	ORT	PS		PD	GCD	ORT	PS
AUC (%)					AUC (%)				
Fold 1	93.90	90.00	92.18	93.43	Fold 1	92.39	91.70	90.23	91.54
Fold 2	93.27	90.17	90.24	90.92	Fold 2	92.97	88.86	92.00	90.85
Fold 3	93.47	89.79	91.97	92.71	Fold 3	93.07	89.66	88.96	90.17
μ_{AUC}	93.55	89.99	91.46	92.35	μ_{AUC}	92.81	90.07	90.40	90.85
EER (%)					EER (%)				
Fold 1	14.33	18.41	15.13	14.28	Fold 1	14.18	16.57	15.62	15.67
Fold 2	14.38	17.27	18.08	16.94	Fold 2	14.22	18.48	15.00	16.47
Fold 3	14.33	18.03	15.09	14.75	Fold 3	14.88	18.41	18.01	18.06
μ_{EER}	14.35	17.90	16.10	15.32	μ_{EER}	14.43	17.82	16.21	16.73

Table 6.8: Results obtained for the set of HMM base classifiers.

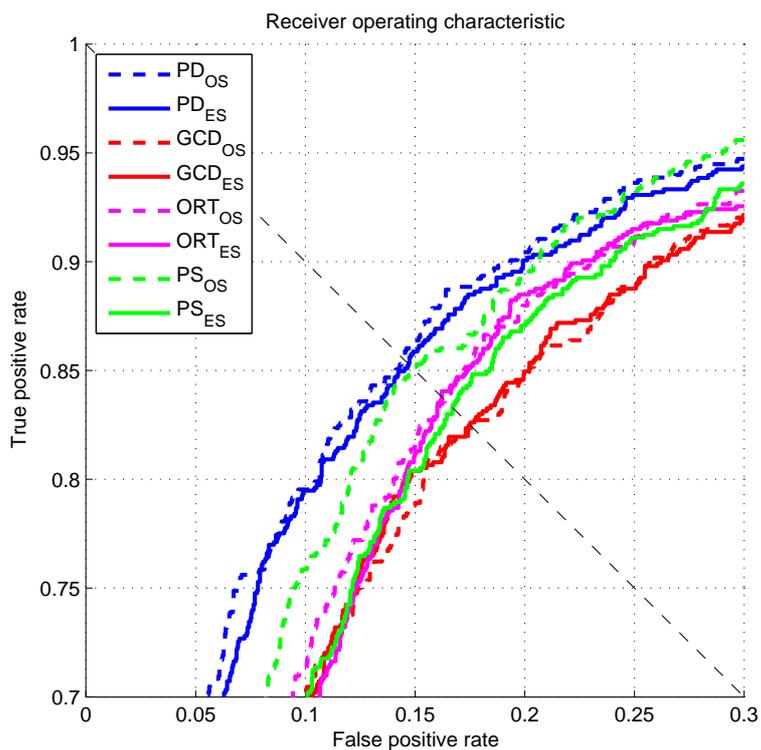


Figure 6.5: ROC-based performance achieved by the set of HMM base classifiers, using both the optimisation set and the evaluation set.

Dolfing's data set				
Feature	PD	GCD	ORT	PS
μ_ϵ (%)	0.74	-0.08	1.06	1.50

Table 6.9: Generalisation errors obtained for the set of HMM base classifiers, using both the optimisation set and the evaluation set.

HMM base classifier is also able to sufficiently reproduce its verification performance, given a set of signatures outside the optimisation set.

In particular, the negative generalisation error achieved, when utilising the gravity centre distance feature, indicates better performance for the evaluation set than the optimisation set. This classifier is therefore said to possess perfect generalisation potential, at least when using Dolfing's data set.

Base classifier ranking

In the next section, we optimise and evaluate the set of combined classifiers discussed in Chapter 5. As mentioned in Section 6.3.2, the optimisation of a combined classifier consists of identifying the optimal number of base classifiers to be included into the ensemble considered for combination. When constructing such a combined classifier, the most important pitfall to avoid is the inclusion of a relatively inaccurate base classifier.

For this reason, each of the DTW and HMM base classifiers, as optimised and evaluated in the previous section, is assigned a specific ranking, based on its performance achieved using Dolfing's optimisation set.

During the remainder of this chapter, when referring to an N -level combined classifier C_{1-N} , we therefore refer to a combined classifier utilising only the set of N top ranked base classifiers, as presented in Table 6.10.

Optimisation set									
Rank	Fold 1			Fold 2			Fold 3		
	C	F	P	C	F	P	C	F	P
1	HMM	PD	93.90	HMM	PD	93.27	DTW	PS	94.17
2	HMM	PS	93.43	DTW	PD	92.54	HMM	PD	93.47
3	DTW	PS	93.34	DTW	PS	92.41	DTW	ORT	92.73
4	HMM	ORT	92.18	DTW	GCD	91.48	HMM	PS	92.71
5	DTW	ORT	91.30	HMM	PS	90.92	HMM	ORT	91.97
6	DTW	GCD	90.87	HMM	ORT	90.24	DTW	GCD	90.70
7	DTW	PD	90.74	HMM	GCD	90.17	DTW	PD	90.08
8	HMM	GCD	90.00	DTW	ORT	90.14	HMM	GCD	89.79

Table 6.10: Performance-based ranking of the DTW and HMM base classifiers, as obtained during model optimisation. For each base classifier, the classification technique (C), feature extraction technique (F) and resulting AUC (%) performance measure (P) is specified.

6.4.2 Combined classifiers

The optimisation of the classifier ensemble selection used by the each of the combined classifiers, as discussed in Section 6.3.2, yields the set of optimal combination levels presented in Table 6.11.

Optimisation set			
	Strategy		
	SA	MV	MAROC
Fold 1	4	4	8
Fold 2	8	7	8
Fold 3	5	5	8

Table 6.11: Optimal combination levels for the set of combined classifiers.

Both the SA and MV combined classifiers reveal a sensitivity regarding the inclusion of relatively inaccurate base classifiers into the classifier ensemble. This is expected, since all the base classifiers in the ensemble have an equal contribution towards the final decision, as discussed in Sections 5.2 and 5.3.1. The efforts of relatively inaccurate base classifiers therefore impede the potential gain in proficiency provided by combining only top ranked base classifiers.

The MAROC combined classifier, in contrast, is consistently improved upon by including the entire set of available base classifiers into the classifier ensemble. As discussed in Section 5.3.2, this is due to the fact that any discrete classifier combinations, shown to perform relatively poorly, can simply be discarded during the combination process. This superior robustness of the MAROC combined classifier enables it to isolate the efforts of any sufficiently accurate discrete classifiers contained in one of the relatively inaccurate base classifiers.

The AUC and EER measures achieved during model optimisation and model evaluation are presented in Table 6.12. The performance is also illustrated in Figure 6.6.

Optimisation set				Evaluation set			
	Strategy				Strategy		
	SA	MV	MAROC		SA	MV	MAROC
AUC (%)				AUC (%)			
Fold 1	96.18	96.69	97.82	Fold 1	94.94	96.34	94.00
Fold 2	96.74	97.06	97.28	Fold 2	95.04	95.22	94.24
Fold 3	95.70	96.80	97.96	Fold 3	95.21	95.83	93.87
μ_{AUC}	96.21	96.85	97.69	μ_{AUC}	95.06	95.80	94.04
EER (%)				EER (%)			
Fold 1	10.09	8.82	7.65	Fold 1	12.54	10.20	11.56
Fold 2	9.64	8.37	9.22	Fold 2	10.98	10.64	11.96
Fold 3	10.09	8.42	8.59	Fold 3	10.10	9.85	14.11
μ_{EER}	9.94	8.54	8.49	μ_{EER}	11.21	10.23	12.54

Table 6.12: Results obtained by the set of combined classifiers.

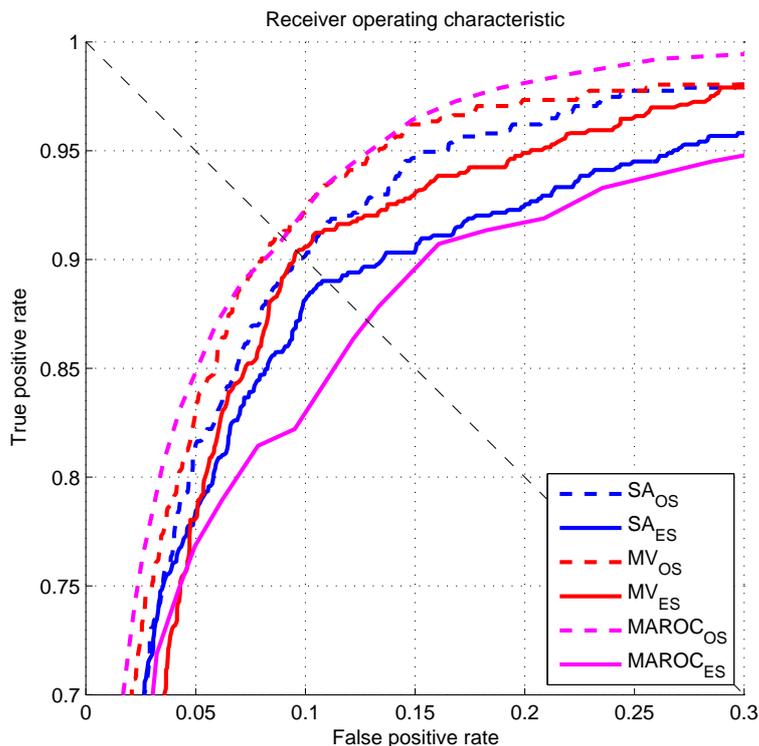


Figure 6.6: ROC-based performance achieved by the set of combined classifiers, using both the optimisation set and the evaluation set.

When compared to the results presented in Tables 6.5 and 6.8, it is clear that a definite gain in verification proficiency is achieved by employing any suitable classifier ensemble combination strategy. Each of the combined classifiers yields significantly better performance measures than the top ranked base classifier, especially regarding the AUC-measure, indicating a convincing improvement in overall system performance. The generalisation error of each combined classifier is presented in Table 6.13.

Dolfing's data set			
Strategy	SA	MV	MAROC
μ_{ϵ} (%)	1.15	1.05	3.65

Table 6.13: Generalisation errors obtained for the set of combined classifiers, using both the optimisation set and the evaluation set.

From Table 6.13, it is clear that both the SA and MV combined classifiers are able to sufficiently reproduce their verification performance, given a set of signatures outside the optimisation set.

The MAROC combined classifier, however, experiences a relatively significant decrease in performance when gauged on the evaluation set. In fact, whilst clearly being the most

proficient classifier when considering only the optimisation set, the MAROC combined classifier is subsequently outperformed on the evaluation set by both the SA and MV combined classifiers. This is most likely due to the fact that the set of DTW and HMM base classifiers, when used to construct a classifier ensemble, do not sufficiently satisfy the requirement of conditional independence. This requirement plays an important role in selecting the appropriate combination rule for each set of discrete classifiers contained in these base classifiers. The issue of conditional independence, and specifically its influence on the performance achieved by a MAROC combined classifier, is discussed further in Section 6.5.2.

In conclusion, when considering Dolfing’s data set, the MV combined classifier outperforms each of the other combined classifiers, in addition to each of the base classifiers. The MV combined classifier is therefore nominated as the optimal off-line signature verification system developed in this study.

6.5 Contributions

During the course of this study, we claim to make two noteworthy contributions to the field of off-line signature verification, namely a flexible grid-based feature extraction technique and an elitist MAROC-based classifier ensemble combination strategy. In this section, these claims are verified experimentally.

6.5.1 Flexible grid-based feature extraction

In order to investigate the advantage gained by employing a flexible grid-based feature extraction technique, we repeat the base classifier optimisation and evaluation processes, under the assumption that only rigid segmentation grids are available for feature extraction. Except for the strict requirement that $F_x = F_y = 0$ during model optimisation, the experiments presented in this section are identical to the experiments presented in Section 6.4.1, thereby nullifying the influence on classifier performance resulting from a flexible grid-based approach.

DTW base classifiers

The optimisation of the rigid grid configurations, used by the DTW base classifiers, yields the set of optimal segmentation grids presented in Table 6.14.

Optimisation set				
DTW	Feature			
	PD	GCD	ORT	PS
Fold 1	20×20	10×40	20×20	20×20
Fold 2	10×40	10×40	20×20	10×20
Fold 3	20×20	10×20	20×20	10×20

Table 6.14: Optimal grid configurations obtained for the set of DTW base classifiers, when only rigid grid-based feature extraction is used.

Except when utilising the orientation feature, there still exists variability regarding an optimal feature-specific segmentation grid. The use of rigid grid-based feature extraction therefore does not overcome the need for context-specific model optimisation.

The AUC and EER measures achieved during model optimisation and model evaluation are presented in Table 6.15. In addition, the generalisation error for each DTW base classifier is presented in Table 6.16.

Optimisation set					Evaluation set				
DTW	Feature				DTW	Feature			
	PD	GCD	ORT	PS		PD	GCD	ORT	PS
AUC (%)					AUC (%)				
Fold 1	84.98	88.52	88.47	91.29	Fold 1	80.81	86.25	85.52	88.17
Fold 2	85.59	88.69	86.91	90.27	Fold 2	79.37	85.54	87.19	90.64
Fold 3	82.16	87.22	87.03	91.20	Fold 3	87.62	88.03	87.38	90.74
μ_{AUC}	84.24	88.14	87.47	90.92	μ_{AUC}	82.60	86.61	86.70	89.85
EER (%)					EER (%)				
Fold 1	23.15	19.88	19.78	16.89	Fold 1	26.27	23.54	22.35	19.60
Fold 2	23.53	21.30	21.68	17.65	Fold 2	27.45	23.53	20.83	16.96
Fold 3	26.09	22.44	21.40	17.70	Fold 3	20.35	19.30	21.19	16.47
μ_{EER}	24.26	21.21	20.95	17.41	μ_{EER}	24.69	22.12	21.46	17.68

Table 6.15: Results obtained for the set of DTW base classifiers, when considering only rigid grid-based feature extraction.

Dolfing's data set				
Feature	PD	GCD	ORT	PS
μ_{ϵ} (%)	1.64	1.53	0.77	1.07

Table 6.16: Generalisation errors obtained for the set of DTW base classifiers, when considering only rigid grid-based feature extraction, using both the optimisation set and the evaluation set.

The results presented in Tables 6.15 and 6.16 are compared to the results achieved by the set of DTW base classifiers reported in Section 6.4.1. These comparisons, presented in Table 6.17 and Figure 6.7, indicate the contribution made by adopting a flexible grid-based approach to feature extraction, within the context of DTW-based signature modelling.

From Table 6.17 it is clear that, for each feature considered, the flexible grid-based feature extraction technique significantly outperforms its rigid grid-based counterpart, both in terms of the AUC and EER measures obtained.

The influence on generalisation potential associated with either a rigid grid-based or flexible grid-based feature extraction process is not as clear. Slight improvements in generalisation potential are observed when utilising the flexible grid-based pixel density, gravity centre distance and predominant slant features. In contrast, utilising a rigid grid-based orientation feature results in improved classifier generalisation. The improvement

Dolfing's data set					
Performance	Grid	Feature			
		PD	GCD	ORT	PS
μ_{AUC} (%)	RG	82.60	86.61	86.70	89.85
	FG	89.90	89.98	89.62	92.35
	+/-	+7.3	+3.37	+2.92	+2.50
μ_{EER} (%)	RG	24.69	22.12	21.46	17.68
	FG	18.30	18.14	18.55	14.24
	+/-	-6.39	-3.98	-2.91	-3.44
μ_{ϵ} (%)	RG	1.64	1.53	0.77	1.07
	FG	1.22	1.04	1.77	0.96
	+/-	-0.42	-0.49	+1.00	-0.11

Table 6.17: Comparison of the results obtained for the set of DTW base classifiers, when considering either a rigid grid (RG) or a flexible grid (FG) for feature extraction. The μ_{AUC} and μ_{EER} measures are obtained using the evaluation set, whilst μ_{ϵ} is obtained using both the optimisation set and the evaluation set.

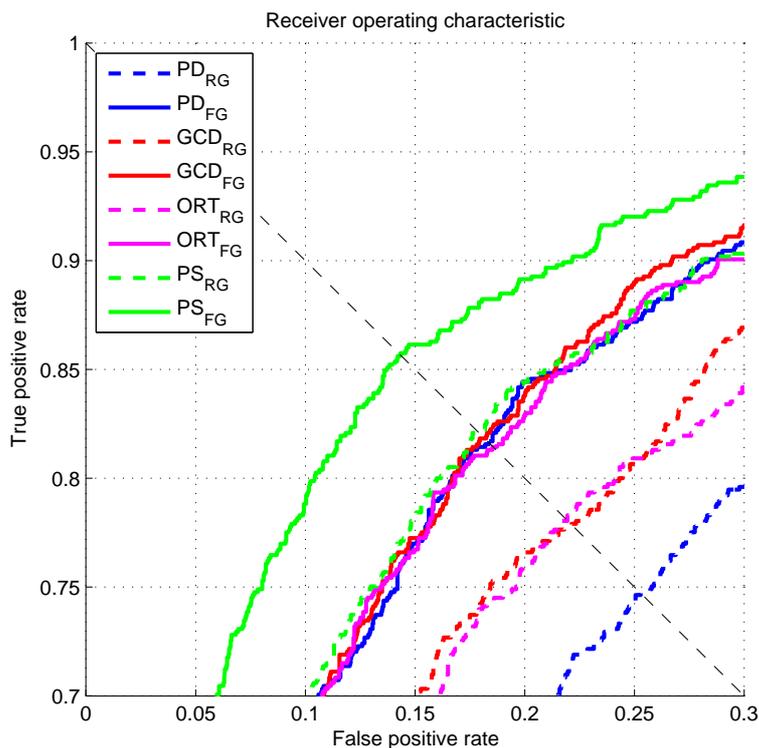


Figure 6.7: ROC-based comparison between results achieved by the set of DTW base classifiers, when considering a rigid grid (RG) and a flexible grid (FG) for feature extraction. Results are obtained using the evaluation set.

achieved in terms of the corresponding AUC measure, however, certainly outweighs this loss in generalisation potential.

We can therefore deduce with certainty that, within the context of DTW-based signature modelling, the use of flexible grid-based feature extraction techniques result in a definite improvement in classifier performance. This gain in classifier performance was expected, since feature vectors constructed by the flexible grid-based approach inherently contain information regarding the succession of feature vectors within the feature set. This additional information (which is not provided by rigid grid-based feature vectors) thereby adds a degree of robustness to the standard DTW-based approach to signature modelling.

HMM base classifiers

The optimisation of the rigid grid configurations, used by the HMM base classifiers, yields the set of optimal segmentation grids presented in Table 6.18.

Optimisation set				
HMM	Feature			
	PD	GCD	ORT	PS
Fold 1	10×40	10×40	10×40	20×40
Fold 2	10×40	10×40	10×40	10×40
Fold 3	10×40	10×40	10×40	20×40

Table 6.18: Optimal grid configurations obtained for the set of HMM base classifiers, when considering only rigid grid-based feature extraction.

When employing rigid grid segmentation, most of the HMM base classifiers do in fact appear to share a common optimal grid configuration. This is, however, not the case when utilising the predominant slant feature, indicating that variability regarding grid configuration may possibly be revealed when increasing the number of folds considered. As was observed from the HMM base classifiers utilising flexible grid-based feature extraction (see Section 6.4.1), each optimal rigid grid contains the maximum number of grid columns considered during optimisation.

The AUC and EER measures achieved during model optimisation and model evaluation are presented in Table 6.19. In addition, the generalisation error for each HMM base classifier is presented in Table 6.20.

The results presented in Tables 6.19 and 6.20 are compared to the results achieved by the set of HMM base classifiers reported in Section 6.4.1. These comparisons, presented in Table 6.21 and Figure 6.8, indicate the contribution made by adopting a flexible grid-based approach to feature extraction, within the context of HMM-based signature modelling.

A consistent improvement in classifier performance is observed when employing flexible grid-based feature extraction, regardless of the feature considered. The improvements in the AUC and EER measures are, however, less significant than the improvements witnessed in the case of the DTW base classifiers. This was expected, as the issue of feature vector succession, one of the main advantages of adopting a flexible grid-based approach,

Optimisation set					Evaluation set				
HMM	Feature				HMM	Feature			
	PD	GCD	ORT	PS		PD	GCD	ORT	PS
AUC (%)					AUC (%)				
Fold 1	90.30	88.61	88.95	91.36	Fold 1	91.04	89.73	89.70	91.04
Fold 2	86.28	86.80	88.00	88.51	Fold 2	90.09	88.02	90.63	90.44
Fold 3	89.96	87.27	87.04	89.96	Fold 3	91.54	89.79	88.78	88.27
μ_{AUC}	88.85	87.56	88.00	89.94	μ_{AUC}	90.89	89.18	89.70	89.92
EER (%)					EER (%)				
Fold 1	16.18	20.21	19.12	16.23	Fold 1	17.16	18.82	17.26	16.82
Fold 2	20.87	20.54	19.55	17.70	Fold 2	17.75	21.18	16.86	18.04
Fold 3	16.18	19.88	21.30	16.94	Fold 3	16.92	18.41	18.81	20.80
μ_{EER}	17.74	20.21	19.99	16.96	μ_{EER}	17.28	19.47	17.64	18.55

Table 6.19: Results obtained for the set of HMM base classifiers, when considering only rigid grid-based feature extraction.

Dolfing's data set				
Feature	PD	GCD	ORT	PS
μ_{ϵ} (%)	2.04	-1.62	-1.70	0.02

Table 6.20: Generalisation errors obtained for the set of HMM base classifiers, when considering only rigid grid-based feature extraction, using both the optimisation set and the evaluation set.

Dolfing's data set					
Performance	Grid	Feature			
		PD	GCD	ORT	PS
μ_{AUC} (%)	RG	90.89	89.18	89.70	89.92
	FG	92.81	90.07	90.40	90.85
	+/-	+1.92	+0.89	+0.70	+0.93
μ_{EER} (%)	RG	17.28	19.47	17.64	18.55
	FG	14.43	17.82	16.21	16.73
	+/-	-2.85	-1.65	-1.43	-1.82
μ_{ϵ} (%)	RG	2.04	-1.62	-1.70	0.02
	FG	0.74	-0.08	1.06	1.5
	+/-	-1.30	+1.54	+2.76	+1.48

Table 6.21: Comparison of the results obtained for the set of HMM base classifiers, when considering either a rigid grid (RG) or a flexible grid (FG) for feature extraction. The μ_{AUC} and μ_{EER} measures are obtained using the evaluation set, whilst μ_{ϵ} is obtained using both the optimisation set and the evaluation set.

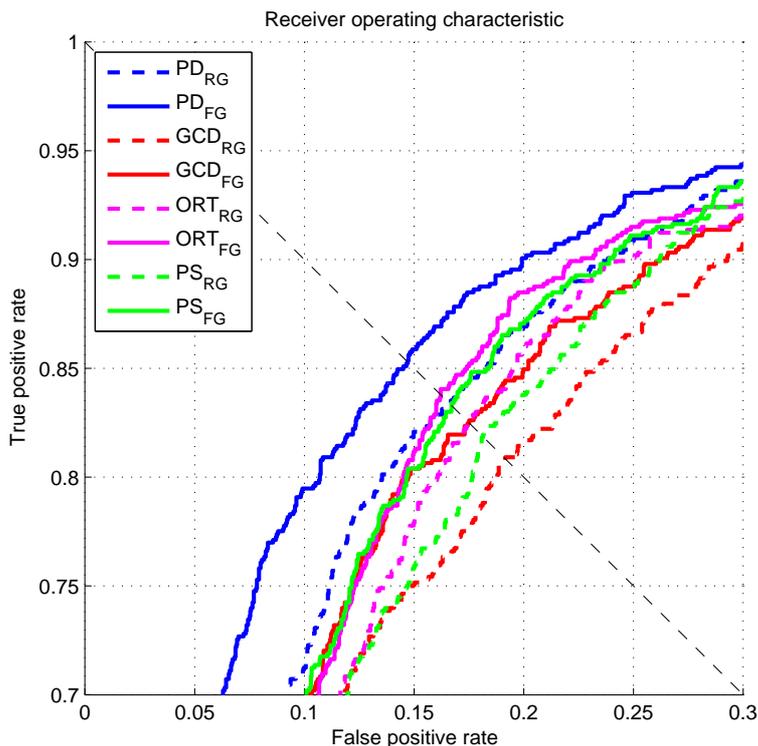


Figure 6.8: ROC-based comparison between results obtained for the set of HMM base classifiers, when considering either a rigid grid (RG) or a flexible grid (FG) for feature extraction. Results are obtained using the evaluation set.

is already addressed by the left-right HMM-based signature modelling strategy. Nevertheless, the additional information regarding signature time-evolution does assist in the construction of more accurate HMM-based signature models.

It is interesting to note that, apart from when utilising the pixel density feature, a consistent improvement in generalisation potential is observed when employing rigid grid-based feature extraction. This is most likely due to context-specific optimisation of the flexibility parameters, thereby creating a highly specialised verification system. Since the improvement in classifier performance is undeniable, though, the prioritisation of either observed verification results or expected generalisation potential is open for debate.

Arguing the greater importance of verification results, we can therefore deduce that, within the context of HMM-based signature modelling, the use of flexible grid-based feature extraction techniques result in a definite improvement in classifier performance.

6.5.2 Elitist MAROC-based classifier ensemble combination

Since the elitist MAROC-based classifier ensemble combination strategy is, in essence, based solely on Haker’s algorithm, it is sensible to investigate whether the results yielded by a traditional implementation of said algorithm is comparable to the results yielded by

the MAROC-based approach. In order to ensure clear and sensible graphical representations, only fold 1 is considered for this comparison.

As mentioned in Section 5.3.2, Haker’s algorithm was specifically designed for the combination of a pair of continuous classifiers. We therefore utilise only the two top ranked base classifiers. The subsequent construction of a combined classifier, based on the traditional implementation of Haker’s algorithm, is illustrated in Figure 6.9.

When using the optimisation set, this combined classifier yields AUC and EER measures of 95.88% and 10.74%, respectively. When using the evaluation set, these measures deteriorate to 94.34% and 12.88%, respectively. A generalisation error of 1.54% is therefore obtained.

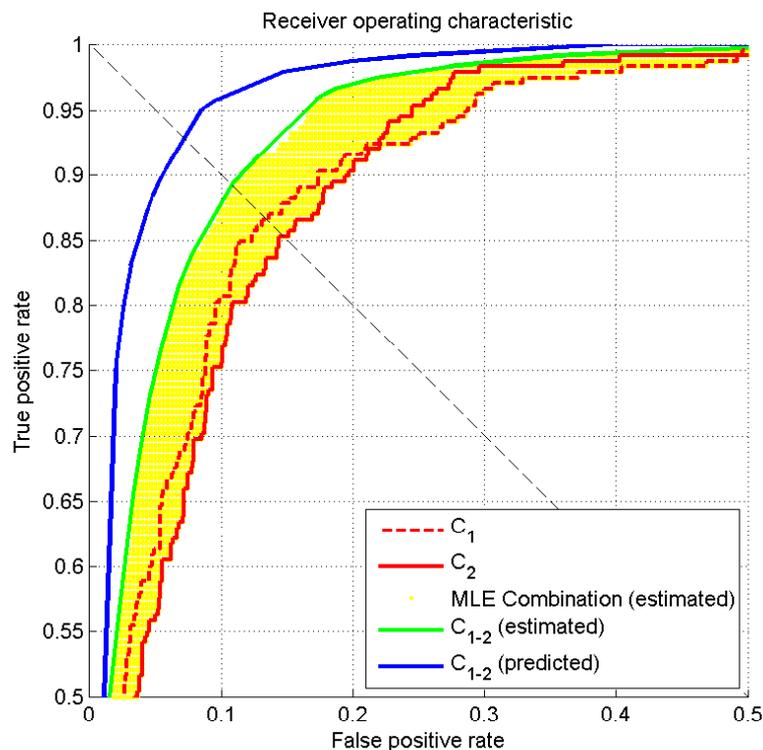


Figure 6.9: ROC-based performance of the combined classifier, using only the two most proficient base classifiers, constructed using Haker’s algorithm. Only the optimisation set associated with fold 1 is used. Note the significant difference between predicted and estimated performance - indicating an insufficient degree of independence between the two classifiers submitted for combination.

As shown earlier, the optimal MAROC combined classifier utilises the entire set of available base classifiers. The combination process is therefore performed in seven iterations, as illustrated in Figures 6.10 and 6.11. The reader is reminded that this approach combines the set of base classifiers in a specific order, namely least proficient to most proficient, as indicated by the corresponding AUC-measure. In addition, since the set of

base classifiers utilised are insufficiently independent, each intermediate combined classifier is constructed using the estimated performance of its predecessor, as opposed to the predicted performance.

It is deemed redundant to include full-scale graphical representations of each iteration, since each corresponding figure illustrates the same general concept. For this reason, only the final iteration is presented in full scale, wherein we indicate each of the relevant entities associated with the combination process.

When using the optimisation set, the iteration-specific performance measures presented in Table 6.22 are obtained.

Optimisation set							
Estimated performance	Iteration						
	1	2	3	4	5	6	7
AUC (%)	93.58	94.73	95.71	96.59	97.31	97.54	97.82
EER (%)	14.37	12.30	10.89	9.75	9.22	8.85	7.65

Table 6.22: Results obtained for the set of intermediate MAROC combined classifiers constructed during iterations 1–7. Only the optimisation set associated with fold 1 is used.

During iterations 1–3, we observe results inferior to those achieved when applying Haker’s algorithm to the two most proficient continuous base classifiers. This is reasonable, as iterations 1–3 represent a combination of the set of four least proficient base classifiers. During each subsequent iteration, however, the MAROC combined classifier consistently outperforms the combined classifier yielded by Haker’s algorithm.

When using the evaluation set, the optimal MAROC combined classifier yields AUC and EER measures of 94.00% and 11.56%, respectively. This combined classifier therefore yields a generalisation error of 3.91%.

From this example, the MAROC-based combination strategy improves system performance to a greater extent than Haker’s algorithm, albeit not significantly, when tested on the evaluation set. This comparison is presented graphically in Figure 6.12. As mentioned earlier, however, the DTW and HMM base classifiers developed in this study do not adequately satisfy the independence criterion, resulting in sub-optimal performance of the corresponding MAROC combined classifier.

We are therefore equally interested in the predicted performance of the MAROC combined classifier, as these measures can be used to infer optimal classifier performance, given independent base classifiers. This optimal performance can be expected when combining sufficiently independent base classifiers, each with an individual performance comparable to the base classifiers developed in this study. When using the optimisation set, the iteration-specific predicted performance measures presented in Table 6.23 are obtained.

From Table 6.23 it is abundantly clear that the requirement of independent classifier decisions plays a critical role in the success of a MAROC combined classifier. The predicted results achievable not only indicate the degree of dependence between the set of base classifiers developed in this study, but also indicate the significant improvement in system performance expected when a classifier ensemble that *does* satisfy the independence criterion is combined. The successful design and implementation of such a

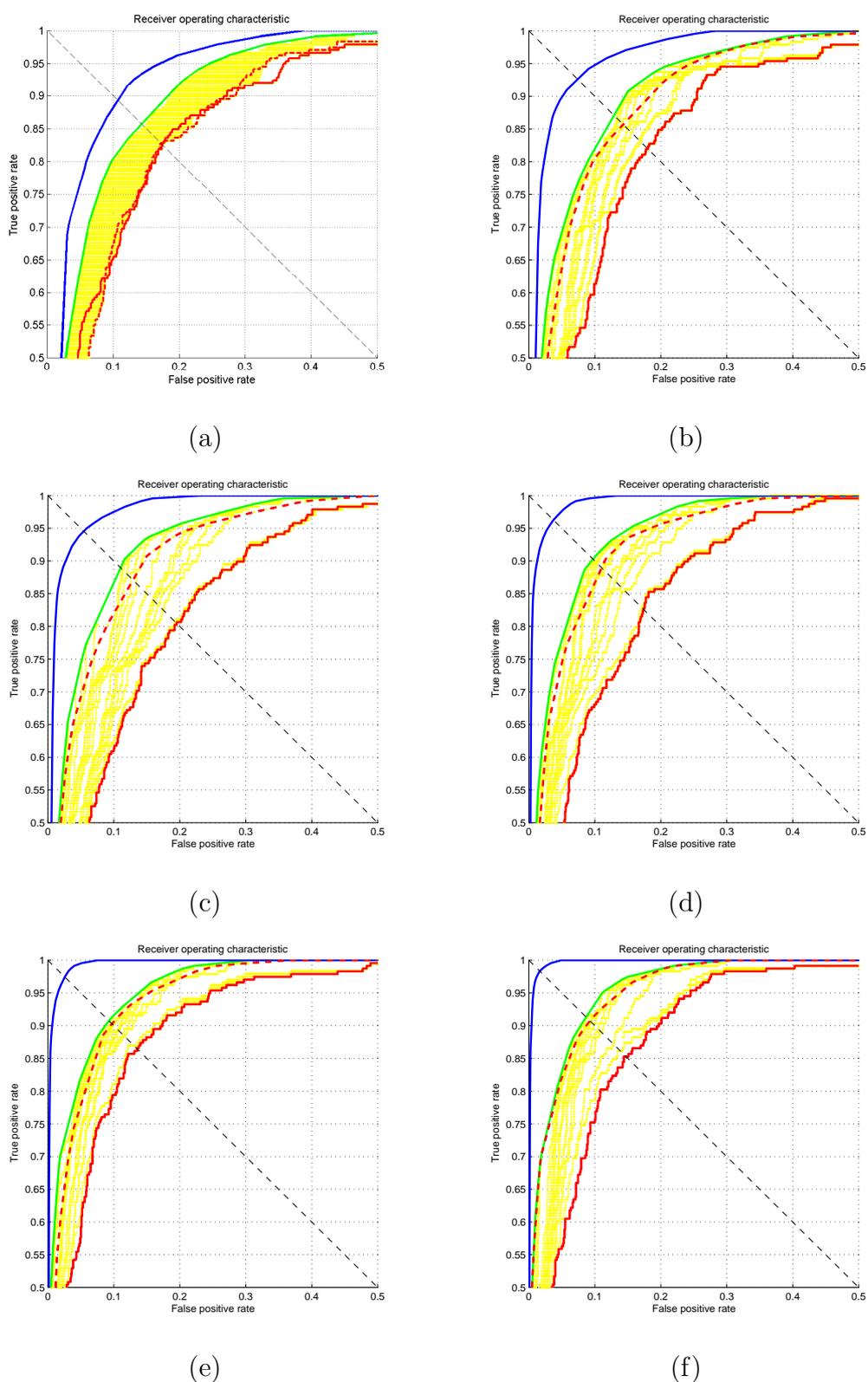


Figure 6.10: ROC-based performance of the set of intermediate MAROC combined classifiers constructed during iterations 1–6. Only the optimisation set associated with fold 1 is used. The final iteration is presented in more detail in Figure 6.11.

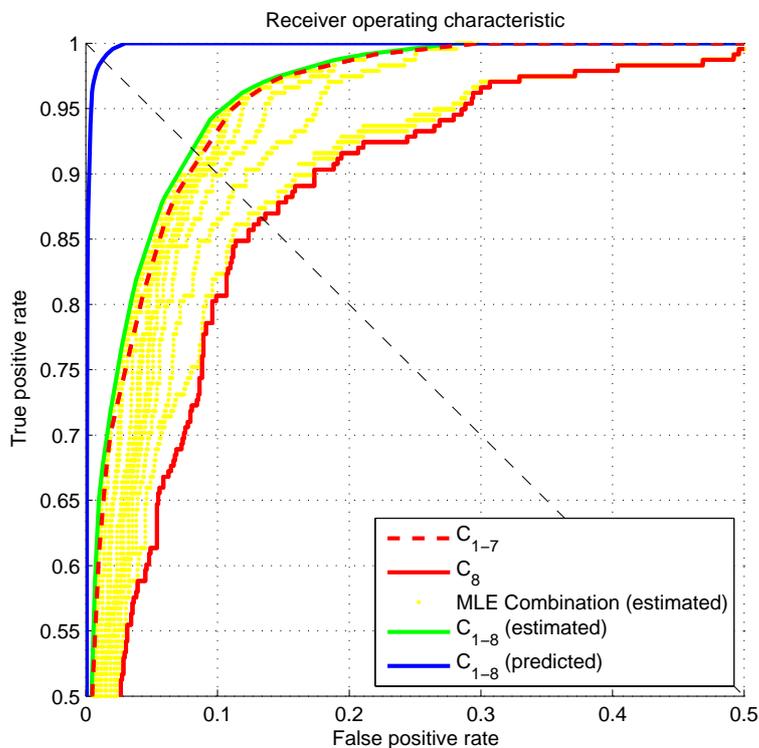


Figure 6.11: ROC-based performance of the MAROC combined classifier constructed during the seventh and final iteration. Only the optimisation set associated with fold 1 is used. Note that, similar to iterations 2–6, the combination of a ROC-curve with a MAROC-curve is associated with significantly fewer discrete classifier combinations than Haker’s original algorithm. It is this property that renders the MAROC-based combination strategy computationally feasible for a much larger number of continuous classifiers, as discussed in Section 5.3.2.

Optimisation set							
Predicted performance	Iteration						
	1	2	3	4	5	6	7
AUC (%)	94.21	98.07	98.78	99.43	99.68	99.82	99.89
EER (%)	9.75	6.44	5.00	3.10	2.27	1.64	1.26

Table 6.23: Predicted performance of the set of intermediate MAROC combined classifiers constructed during iterations 1–7. Only the optimisation set associated with fold 1 is used. These predictions assume conditionally independent classifier decisions.

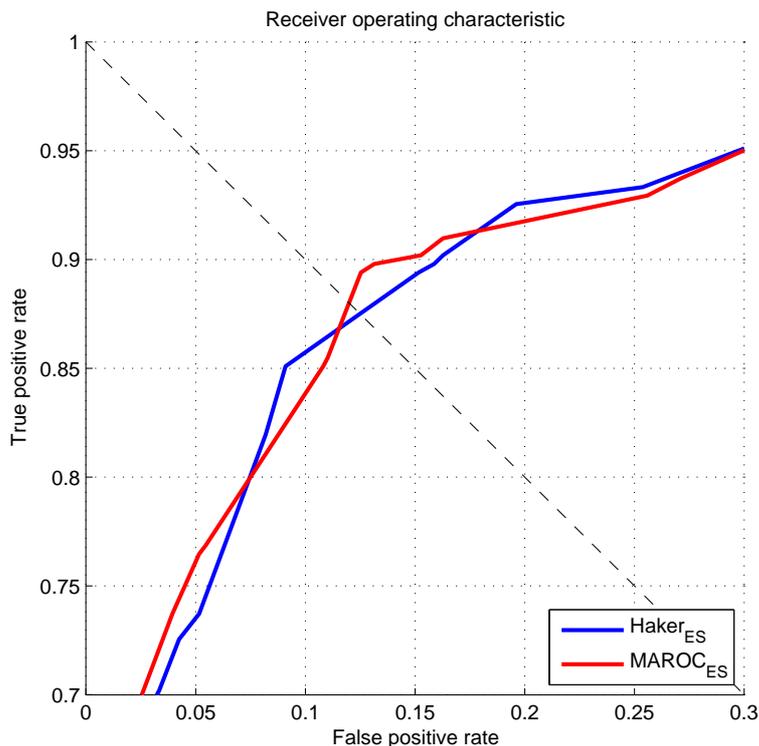


Figure 6.12: ROC-based comparison between results obtained for the combined classifier using Haker’s algorithm with the MAROC combined classifier developed in this study. Results are obtained using only the evaluation set associated with fold 1.

conditionally independent classifier ensemble therefore definitely warrants further investigation. Such an investigation, however, is considered outside the scope of this study.

6.6 Comparison with previous work

Since each of the verification system candidates developed in this study utilises a flexible grid-based feature extraction technique, these systems may be considered novel. In this section, we are more interested in whether any of these novel systems contribute to the current state of the art in terms of verification proficiency.

As mentioned in Section 2.1, there currently exists no standard international signature database to facilitate a direct comparison between results reported in this study with results reported in the literature. We are therefore limited in the sense that only verification systems evaluated using Dolfing’s data set may be used for such a comparison. Two such systems are found in the literature, namely the on-line verification system proposed by Dolfing (1998) and the off-line verification system proposed by Coetzer *et al.* (2004).

Since Dolfing’s system is aimed at the verification of dynamic signatures, it is not entirely sensible to directly compare all his reported results to the results reported in this

study. We may, however, use the algorithm developed by Dolfing which considers only the spatial coordinates of dynamic signatures. Even so, this algorithm uses both pen stroke coordinates and pen trajectory information, thereby giving it a distinct advantage over systems developed in this study.

Since the system proposed by Coetzer *et al.* is aimed at the verification of static signatures, no such discrepancies are present in the model training process. This system is therefore considered much more suitable for comparison to the results reported in this study. It is worth mentioning, however, that the HMM-based system developed by Coetzer *et al.* uses the same data set for both model optimisation and evaluation. This system therefore also has an advantage over systems developed in this study, although to a much lesser extent.

The results reported by Dolfing and Coetzer *et al.* are presented in Table 6.24, along with the results reported for the SA and MV combined classifiers developed in this study. The MAROC combined classifier is not included in this comparison, since it was shown in Section 6.5.2 that this combination strategy, when employed on the DTW and HMM base classifiers developed in this study, performs sub-optimally. We therefore reserve the comparison of a MAROC combined classifier, to suitable systems found in the literature, for a future study.

Author/System	EER (%)
Dolfing (1998)	13.3
Coetzer <i>et al.</i> (2004)	12.2
SA (This study)	11.21
MV (This study)	10.23

Table 6.24: Comparison of the results obtained for the SA and MV combined classifiers developed in this study, after considering amateur forgeries only, with previous systems evaluated using Dolfing’s data set.

The results reported for both the SA and MV combined classifiers compare well with the results reported by Dolfing, as well as those reported by Coetzer *et al.*. Since the experimental protocol used in this study ensures an objective, comprehensive measure of performance, the results reported in this study are considered a credible improvement.

6.7 Concluding remarks

In this chapter we evaluated the performance achieved by the set of DTW and HMM base classifiers, as well as the set of score-based and decision-based combined classifiers, developed in this study. Each verification system candidate was optimised and evaluated using separate data sets, in conjunction with a 3-fold cross-validation protocol, thereby ensuring the credibility of results reported in this chapter.

The model evaluation process revealed that the set of HMM base classifiers generally outperforms the set of DTW base classifiers. The optimal DTW base classifier, utilising a predominant slant feature, and the optimal HMM base classifier, utilising a pixel density feature, achieve comparable performance measures. Furthermore, it was shown that the

set of combined classifiers consistently outperform both the optimal DTW and HMM base classifiers. The MV combined classifier proved most proficient.

In addition, the significance of the flexible grid-based feature extraction technique, as well as the potential significance of the elitist MAROC-based classifier ensemble strategy, was confirmed experimentally. The results reported in this chapter compare well with those of existing systems, previously proposed in the literature, which also consider the amateur skilled forgeries within Dolfing's data set.

In the next chapter, we conclude this study by briefly discussing selected key issues. Also presented is a selection of topics deemed both relevant and potentially beneficial to the systems developed in this study, but considered as future work.

Chapter 7

Conclusion and Future Work

“The important thing is not to stop questioning.”
- Albert Einstein (1879–1955)

7.1 Conclusion

During the course of this study, we developed a set of eight continuous base classifiers. Each of these classifiers was constructed by combining the efforts of a specific local feature with a specific signature modelling technique. In addition, we developed a set of three combined classifiers, each constructed by combining the efforts of an ensemble of base classifiers by means of either score-based or decision-based fusion strategies.

As mentioned in Section 1.3, this study had two primary objectives. The first objective, namely the design and implementation of a novel feature extraction technique, was completed successfully. The feature extraction technique proposed in this study utilises existing local features in conjunction with a novel flexible grid-based signature segmentation strategy. This flexible grid-based feature extraction technique was shown to consistently outperform the traditional rigid grid-based feature extraction technique found in the literature. The second objective, namely the design and implementation of a robust combined classifier, was also completed successfully. We considered two existing classifier fusion strategies, namely score averaging and the majority vote rule, as well as a novel elitist MAROC-based classifier ensemble combination strategy. The MAROC-based approach is an extension of the existing ROC-based combination strategy for classifier pairs. Each of the combined classifiers was shown to outperform the top ranked base classifier. Furthermore, we showed that a significant gain in proficiency is expected for the MAROC combined classifier, subject to the construction of a classifier ensemble from sufficiently independent base classifiers.

The set of novel combined classifiers developed in this study compare well with previous systems reported in the literature. This is an exciting prospect, since the MAROC combined classifier was shown to perform sub-optimally. We may therefore realistically expect to significantly improve upon the results reported in this study.

7.2 Future work

During the course of this study, certain concepts were encountered that warrant further investigation, as they are deemed potentially beneficial to the systems developed in this study. These concepts are not included in this work, either due to being considered outside the scope of this study, or simply due to time constraints. In this section, some of these concepts are briefly discussed as a possible continuation of this study.

7.2.1 Adaptive grid segmentation

In Section 6.5.1 we verified experimentally the improvement in verification proficiency gained by adopting the flexible grid-based feature extraction technique. The flexible segmentation grids developed in this study, however, are subject to the requirement that each grid cell boundary is dilated, if possible, to exactly the same extent. The application-specific optimal flexible segmentation grid therefore needs to be obtained experimentally, as shown in Section 6.3.2. The optimal implementation of this concept of overlapping segmentation grid cells is therefore definitely worth investigating as part of further research.

In particular, we are interested in the potential advantages obtainable by the development of an intelligent, non-uniform adaptive segmentation grid. The property of non-uniformity simply implies that the flexibility parameters associated with each individual grid cell need not necessarily be comparable. The property of adaptivity implies that the system will automatically determine each grid cell's optimal flexibility, during run-time, upon receiving an arbitrary signature image. This process of self-optimisation should be based on the identification of a certain region of interest, as described by a specific property, perhaps analogous to the self-optimisation of the AMF-window discussed in Section 3.2.

7.2.2 Conditionally independent classifier ensembles

In Section 6.5.2 we showed experimentally that a MAROC combined classifier, constructed from an ensemble of eight conditionally independent base classifiers, is predicted to achieve AUC and EER measures on Dolfig's optimisation set of 99.89% and 1.26%, respectively. Such levels of verification proficiency are usually reserved for on-line systems. No predictions can sensibly be made on the evaluation set. As discussed in Section 5.3.2, this prediction remains valid only if the set of independent base classifiers each has an individual performance comparable to the DTW and HMM base classifiers developed in this study. Considering the base classifier performance reported in this study, this is deemed a reasonably attainable goal.

It therefore definitely warrants further investigation to attempt the design and implementation of such a conditionally independent classifier ensemble. The requirement of independent classifier decisions may most likely be achieved by either considering a fundamentally different set of features and/or signature modelling techniques. Potential features to investigate include the DRT, a multi-angle global projection feature, or perhaps the Fourier boundary descriptor feature. Definite candidates for investigation include the NN-based and SVM-based signature modelling techniques. In fact, the study

by Justino *et al.* (2005) suggests that an SVM base classifier is likely to outperform an otherwise similar HMM base classifier.

It is therefore considered quite possible to achieve the predicted results reported in Section 6.5.2, which would constitute a most significant contribution to the current state of the art.

7.2.3 The writer-independent approach

As mentioned in the previous section, we are very interested in investigating the potential benefit, especially to the MAROC combined classifier developed in this study, by including base classifiers utilising NNs and SVMs.

The inclusion of such discriminative classifiers, however, requires the availability of skilled forgeries for model training. As this requirement is satisfied only by the set of guinea pig writers, a writer-dependent verification strategy is considered infeasible for the purposes of sensible model evaluation. By employing the writer-independent approach, as discussed in Section 1.2.7, this impediment is circumvented, since no model retraining is required prior to model evaluation.

Furthermore, since a single model is employed for all new clients, the use of writer-independent base classifiers significantly reduces both the computational cost and the memory requirements of the resulting classifier ensemble. This property therefore greatly improves the economic viability of the resulting off-line signature verification system.

Bibliography

- Alpaydin, E. (1998). Soft vector quantization and the EM algorithm. *Neural Networks*, vol. 11, pp. 467–477.
- American Bankers Association (2007). Attempted check fraud doubles to \$12.2 billion according to ABA survey.
- Armand, S., Blumenstein, M. and Muthukkumarasamy, V. (2006). Off-line signature verification using the enhanced modified direction feature and neural-based classification. *International Joint Conference on Neural Networks*, pp. 684–691.
- Association for Payment Clearing Services (2008). APACS announces latest fraud figures.
- Baltazakis, H. and Papamarkos, N. (2001). A new signature verification technique based on a two-stage neural network classifier. *Engineering Applications of Artificial Intelligence*, vol. 14, pp. 95–103.
- Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Cappelli, R., Maio, D. and Maltoni, D. (2000). Combining fingerprint classifiers. *International Workshop on Multiple Classifier Systems*, vol. 1, pp. 351–361.
- 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., Herbst, B. and du Preez, J. (2006). Off-line signature verification: A comparison between human and machine performance. *International Workshop on Frontiers in Handwriting Recognition*, vol. 10, pp. 481–485.
- Deller, J., Proakis, J. and Hansen, J. (1987). *Discrete-Time Processing of Speech Signals*. Prentice-Hall Inc.
- Dolfing, J. (1998). *Handwriting Recognition and Verification - A Hidden Markov Approach*. Ph.D. thesis, Eindhoven University of Technology.
- Dos Santos, E., Sabourin, R. and Maupin, P. (2008). A dynamic overproduce-and-choose strategy for the selection of classifier ensembles. *Pattern Recognition*, vol. 41, pp. 2993–3009.

- Fang, B., Wang, Y., Leung, C., Tse, K., Tang, Y., Kwok, P. and Wong, Y. (2001). Offline signature verification by the analysis of cursive strokes. *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 4, pp. 659–673.
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, vol. 27, pp. 861–874.
- Fink, G. (2008). *Markov Models for Pattern Recognition: From Theory to Applications*. Springer.
- Gonzalez, R. and Woods, R. (2002). *Digital Image Processing*. 2nd edn. Prentice-Hall Inc.
- Güler, I. and Meghdadi, M. (2008). A different approach to off-line handwritten signature verification using the optimal dynamic time warping algorithm. *Digital Signal Processing*, vol. 18, pp. 940–950.
- Haker, S., Wells, W., Warfield, S., Talos, I., Bhagwat, J., Goldberg-Zimring, D., Mian, A., Ohno-Machado, L. and Zou, K. (2005). Combining classifiers using their receiver operating characteristics and maximum likelihood estimation. *International Conference on Medical Image Computing and Computer Assisted Intervention*, vol. 8, no. 1, pp. 506–514.
- Haralick, R. and Shapiro, L. (1992). *Computer and Robot Vision*, vol. 1. Addison-Wesley.
- Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *Institute of Radio Engineers Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187.
- Huang, K. and Yan, H. (1997). Off-line signature verification based on geometric feature extraction and neural network classification. *Pattern Recognition*, vol. 30, no. 1, pp. 9–17.
- Impedovo, D. and Pirlo, G. (2008). Automatic signature verification: The state of the art. *Institute of Electrical and Electronics Engineers Transactions on Systems, Man and Cybernetics*, vol. 38, no. 5, pp. 609–635.
- Jackowski, K. and Wozniak, M. (2009). *Adaptive Splitting and Selection Method of Classifier Ensemble Building*, pp. 525–532. Lecture Notes in Computer Science. Springer.
- Jain, A., Nandakumar, K. and Ross, A. (2005). Score normalization in multimodal biometric systems. *Pattern Recognition*, vol. 18, pp. 2270–2285.
- Justino, E., Bortolozzi, F. and Sabourin, R. (2001). Off-line signature verification using HMM for random, simple and skilled forgeries. *International Conference on Document Analysis and Recognition*, vol. 1, pp. 105–110.
- Justino, E., Bortolozzi, F. and Sabourin, R. (2005). A comparison of SVM and HMM classifiers in the off-line signature verification. *Pattern Recognition Letters*, vol. 26, pp. 1377–1385.

- Justino, E., El Yacoubi, A., Bortolozzi, F. and Sabourin, R. (2000). An off-line signature verification system using HMM and graphometric features. *International Conference on Pattern Recognition*, vol. 2, pp. 11–17.
- Kim, Y.-W. and Oh, I.-S. (2008). Classifier ensemble selection using hybrid genetic algorithms. *Pattern Recognition Letters*, vol. 29, pp. 796–802.
- Lv, H., Wang, W., Wang, C. and Zhuo, Q. (2005). Off-line Chinese signature verification based on support vector machines. *Pattern Recognition Letters*, vol. 26, no. 15, pp. 2390–2399.
- Majhi, B., Reddy, Y. and Babu, D. (2006). Novel features for off-line signature verification. *International Journal of Computers, Communication & Control*, vol. 1, no. 1, pp. 17–24.
- Oliveira, L., Justino, E., Freitas, C. and Sabourin, R. (2005). The graphology applied to signature verification. *Conference of the International Graphonomics Society*, vol. 12, pp. 286–290.
- Oliveira, L., Justino, E., Sabourin, R. and Bortolozzi, F. (2008). Combining classifiers in the ROC-space for off-line signature verification. *Journal of Universal Computer Science*, vol. 14, no. 2, pp. 237–251.
- Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Institute of Electrical and Electronics Engineers*, vol. 77, no. 2, pp. 257–286.
- Santos, C., Justino, E., Bortolozzi, F. and Sabourin, R. (2004). An off-line signature verification method based on the questioned document experts approach and a neural network classifier. *International Workshop on Frontiers in Handwriting Recognition*, vol. 9, pp. 498–502.
- Schmidt, C. (1994). Signature verification using time-delay neural networks. *Midwest Symposium on Circuits and Systems*, vol. 37, no. 2, pp. 1395–1398.
- Shanker, A. and Rajagopalan, A. (2007). Off-line signature verification using DTW. *Pattern Recognition Letters*, vol. 28, no. 12, pp. 1407–1414.
- Swanepoel, J. (2007). *Off-line Handwritten Character Recognition*. Honour’s seminar, Stellenbosch University.
- Tanguay, D. (1993). *Hidden Markov Models for Gesture Recognition*. Master’s thesis, Massachusetts Institute of Technology.
- United States Census Bureau (2008). International database, population division.
- Wen, J., Fang, B. and Zhang, Y.T.T. (2009). Model-based signature verification with rotation invariant features. *Pattern Recognition*, vol. 42, pp. 1458–1466.

Appendix A

Dynamic Time Warping: Key Concepts

This appendix provides a detailed discussion on the DTW-algorithm, as used by the DTW base classifiers developed in this study.

A.1 Algorithm

In order to obtain the DTW-based distance between a questioned vector \mathbf{x}_q and a reference vector \mathbf{x}_k , a grid is constructed such that each node (i, j) relates component i of \mathbf{x}_q to component j of \mathbf{x}_k . For each node, the distance

$$D_{\text{node}}(i, j) = (\mathbf{x}_q(i) - \mathbf{x}_k(j))^2, \quad 1 \leq i, j \leq d \quad (\text{A.1})$$

is computed, which is said to reflect the *node-based cost* associated with $\mathbf{x}_q(i)$ and $\mathbf{x}_k(j)$. The optimal path is subsequently defined as the *complete* path

$$(i_0, j_0)(i_1, j_1) \dots (i_K, j_K) \quad (\text{A.2})$$

through the grid for which the total node-based cost

$$D_{\text{node}}^{(\text{compl})}(i, j) = \sum_{k=0}^K D_{\text{node}}(i_k, j_k) \quad (\text{A.3})$$

is minimised. Several constraints are imposed on the solution space, in order to ensure that the resulting optimal path is in fact a valid path.

Firstly, it is required that the optimal path is complete. In other words, for two d -dimensional feature vectors \mathbf{x}_q and \mathbf{x}_k , $\mathbf{x}_q(1)$ must always be related to $\mathbf{x}_k(1)$, whilst $\mathbf{x}_q(d)$ must always be related to $\mathbf{x}_k(d)$. This is achieved by the requirement

$$(i_0, j_0) = (1, 1), \quad (\text{A.4})$$

$$(i_K, j_K) = (d, d). \quad (\text{A.5})$$

It is not strictly required that \mathbf{x}_q and \mathbf{x}_k share the same dimension, as indicated by Constraint A.5, although this is always the case for base classifiers developed in this study.

Furthermore, it is required that the optimal path be *monotonically increasing*, as illustrated in Figure 4.2. A node (i, j) may therefore only be considered for inclusion into the optimal path if it is preceded by the node $(i - 1, j - 1)$, $(i, j - 1)$ or $(i - 1, j)$. Consequently, it follows that

$$i_k \geq i_{k-1}, \quad k = 1, 2, \dots, K, \quad (\text{A.6})$$

$$j_k \geq j_{k-1}, \quad k = 1, 2, \dots, K. \quad (\text{A.7})$$

As a result, any node that fails to satisfy Constraints A.6 and A.7 does not form part of a legitimate path and can therefore not form part of the optimal path. Finally, when constructing the optimal path, it is required that

$$|j_k - i_k| \leq H_{\text{vec}}, \quad k = 0, 1, \dots, K. \quad (\text{A.8})$$

Since the *bandwidth* $H_{\text{vec}} \in [0, d]$ may be chosen arbitrarily, Constraint A.8 is not considered rigid, although it does address two key aspects regarding the practicality of the obtained solution. Firstly, it ensures that components with exceedingly different indices are not related, which would not be sensible. Secondly, it limits the computational cost associated with the DTW-algorithm. It should be clear that for feature vectors of a very high dimension, constructing a complete cost grid could become computationally exhaustive. Note that when $H_{\text{vec}} = 0$, the optimal path is restricted to the diagonal $i = j$, thereby producing the Euclidean distance.

Before we discuss the DTW-based algorithm for obtaining the *complete optimal path* and its associated *complete optimal cost*, it is necessary to define two key concepts. Let $D_{\text{node}}^{(\text{part})}(i, j)$ denote the *partial optimal path* that terminates at node (i, j) . Also, let $\leftarrow (i, j)$ denote the *optimal preceding node* for node (i, j) . A preceding node is deemed optimal if the partial optimal path that passes through it and terminates at (i, j) minimises the *partial optimal cost*.

The complete procedure for finding the optimal path and corresponding DTW-based distance by means of the DTW-algorithm can now be stated as follows:

- Initialisation:

$$D_{\text{node}}^{(\text{part})}(1, 1) = D_{\text{node}}(1, 1). \quad (\text{A.9})$$

- Recursion:

All nodes within the allotted bandwidth are considered in a left-to-right, bottom-to-top fashion. For each node considered, $\leftarrow (i, j)$ is computed as

$$C_1 = D_{\text{node}}^{(\text{part})}(i - 1, j - 1), \quad (\text{A.10})$$

$$C_2 = D_{\text{node}}^{(\text{part})}(i, j - 1), \quad (\text{A.11})$$

$$C_3 = D_{\text{node}}^{(\text{part})}(i - 1, j), \quad (\text{A.12})$$

$$\leftarrow (i, j) = \text{argmin}\{C_1, C_2, C_3\}. \quad (\text{A.13})$$

If the minimum value for C_i is shared by more than one of the preceding nodes, $\leftarrow (i, j)$ is selected in the following order of preference: $(i - 1, j - 1)$, then $(i, j - 1)$, then $(i - 1, j)$. Subsequently, $D_{\text{node}}^{(\text{part})}(i, j)$ is computed as

$$D_{\text{node}}^{(\text{part})}(i, j) = D_{\text{node}}(i, j) + D_{\text{node}}^{(\text{part})}(\leftarrow (i, j)). \quad (\text{A.14})$$

- Path backtracking:

As ensured by Constraints A.4 and A.5, nodes (i_0, j_0) and (i_K, j_K) of the optimal path are reserved for $(1, 1)$ and (d, d) , respectively. The remainder of the optimal path may be obtained through backtracking from node (i_K, j_K) by iteratively letting

$$(i_k, j_k) = \leftarrow (i_{k+1}, j_{k+1}). \quad (\text{A.15})$$

- Termination:

$$D_{\text{node}}^{(\text{compl})}(\mathbf{x}_q, \mathbf{x}_k) = D_{\text{node}}^{(\text{part})}(d, d). \quad (\text{A.16})$$

Appendix B

Hidden Markov Models: Key Concepts

This appendix provides detailed discussions on fundamental theory surrounding the discrete observation HMMs developed in this study. Throughout this appendix, the notation established in Section 4.4.2 is used.

B.1 The three basic problems of HMMs

In order to apply a discrete observation HMM to any real-world application, three basic problems need to be solved:

HMM Problem I: Model evaluation. Given a discrete observation sequence $\mathbf{O} = \{o_1, o_2, \dots, o_T\}$ and a model $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$, how do we efficiently compute $P(\mathbf{O}|\lambda)$, the probability that \mathbf{O} was generated by λ ?

HMM Problem II: Finding the optimal state sequence. Given a discrete observation sequence $\mathbf{O} = \{o_1, o_2, \dots, o_T\}$ and a model $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$, how do we determine the state sequence $\mathbf{Q}^* = \{q_1^*, q_2^*, \dots, q_T^*\}$ that most probably generated \mathbf{O} ?

HMM Problem III: Model optimisation. Given a discrete observation sequence $\mathbf{O} = \{o_1, o_2, \dots, o_T\}$ and a model $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$, how do we adjust the model parameters, thereby obtaining $\bar{\lambda} = (\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\pi})$, such that $P(\mathbf{O}|\bar{\lambda}) \geq P(\mathbf{O}|\lambda)$?

The primary objective of HMM design is therefore to provide a set of robust solutions to these three problems, which constitutes the foundation of any efficient HMM implementation. In Sections B.2 and B.3 we show how the Viterbi algorithm may be used to provide efficient solutions to all three basic problems.

B.2 The Viterbi algorithm

The Viterbi algorithm attempts to solve HMM Problems I and II by utilising dynamic programming techniques to maximise $P(\mathbf{Q}|\mathbf{O}, \lambda)$.

In order to determine the optimal state sequence \mathbf{Q}^* associated with the observation sequence $\mathbf{O} = \{o_1, o_2, \dots, o_T\}$, the quantity

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} [P(q_t = s_i, \mathbf{O}|\lambda)] \quad (\text{B.1})$$

is defined as the highest probability along a single path which accounts for the first t observations and ends in state s_i . It follows by induction that

$$\delta_{t+1}(j) = \max_{1 \leq i \leq N} [\delta_t(i) a_{ij}] b_j(o_{t+1}). \quad (\text{B.2})$$

Furthermore, the array $\psi_t(j)$ is used to keep record of the argument which maximises $\delta_{t+1}(j)$, for each t and j , thereby ensuring that the state sequence may be retrieved.

The complete procedure for finding the best state sequence by means of the Viterbi algorithm can now be stated as follows:

- Initialisation:

$$\delta_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N \quad (\text{B.3})$$

$$\psi_1(i) = 0 \quad (\text{B.4})$$

- Recursion:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(o_t), \quad \begin{array}{l} 2 \leq t \leq T \\ 1 \leq j \leq N \end{array} \quad (\text{B.5})$$

$$\psi_t(j) = \operatorname{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \quad \begin{array}{l} 2 \leq t \leq T \\ 1 \leq j \leq N \end{array} \quad (\text{B.6})$$

- Termination:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (\text{B.7})$$

$$q_T^* = \operatorname{argmax}_{1 \leq i \leq N} [\delta_T(i)] \quad (\text{B.8})$$

- Path backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \quad (\text{B.9})$$

Given the discrete observation sequence $\mathbf{O} = \{o_1, o_2, \dots, o_T\}$, the Viterbi algorithm consequently produces its optimal state sequence $\mathbf{Q}^* = \{q_1^*, q_2^*, \dots, q_T^*\}$, as well its probability $P^* = P(\mathbf{Q}^*|\lambda)$, thereby providing solutions to HMM Problems II and I, respectively.

B.3 Training

Each HMM constructed in this study represents a writer-dependent signature model. It is therefore of critical importance that each such HMM is as representative as possible of the sample signatures available for training. In this section, we discuss how the Viterbi algorithm may be used to obtain an optimal configuration for each writer's associated HMM, thereby providing a solution to HMM Problem III.

B.3.1 Parameter optimisation

Apart from the initial state distribution π , which is permanently bound by the left-right topology considered in this study, the set of HMM parameters may theoretically be initialised arbitrarily, as they are to be reestimated during model training. It has been shown, however, that the initial parameter estimates have a definite influence on the accuracy of the optimised model obtained after reestimation. The importance of an effective initialisation for λ is illustrated conceptually in Figure B.1.

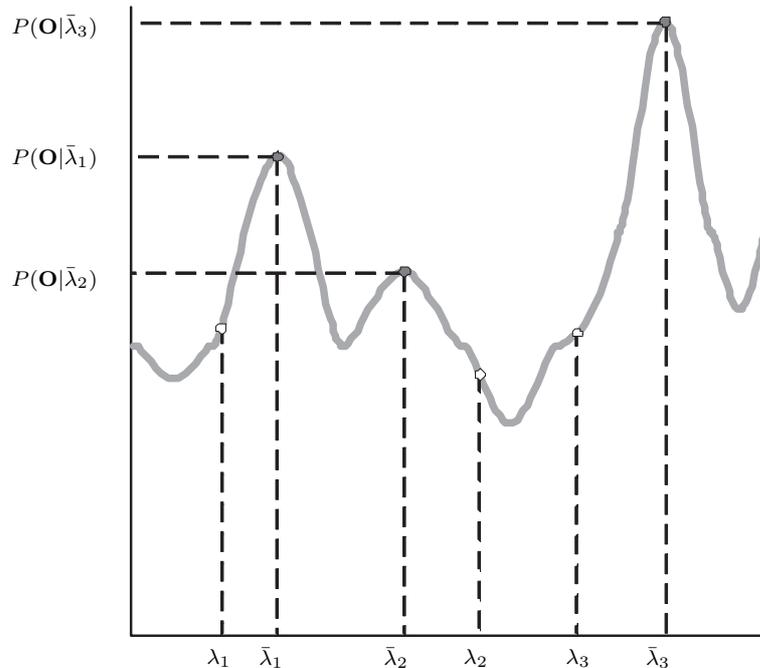


Figure B.1: Conceptualisation of the observation sequence probability $P(\mathbf{O}|\lambda_i)$ as a function of the model configuration λ_i . From this conceptualisation it is clear that the convergence of $P(\mathbf{O}|\lambda_i)$, to a local or global maximum $P(\mathbf{O}|\bar{\lambda}_i)$, is based solely on the corresponding parameter initialisation λ_i .

During model training, given an observation sequence \mathbf{O} and model λ , the Viterbi algorithm is used to obtain the optimal state sequence \mathbf{Q}^* . The adjusted HMM parameters

$\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ may subsequently be estimated by letting

$$\bar{a}_{ij} = \frac{\text{number of transitions from state } s_i \text{ to state } s_j \text{ in } \mathbf{Q}^*}{\text{number of times state } s_i \text{ occurs in } \mathbf{Q}^*}, \quad (\text{B.10})$$

$$\bar{b}_j(k) = \frac{\text{number of times state } s_j \text{ emits symbol } v_k \text{ in } \mathbf{Q}^*}{\text{number of times state } s_j \text{ occurs in } \mathbf{Q}^*}, \quad (\text{B.11})$$

whilst π remains fixed, thereby enforcing the left-right topology of the associated HMM model.

This process is repeated iteratively until convergence of the observation sequence probability $P(\mathbf{O}|\bar{\lambda})$ is achieved. As mentioned earlier, there is no guarantee that the final observed value for $P(\mathbf{O}|\bar{\lambda})$ represents a *global* maximum (see Figure B.1), reiterating the vital importance of suitable initial estimates of \mathbf{A} and \mathbf{B} .

B.3.2 Multiple observation sequences

If one is to effectively reestimate the model parameters λ , a single observation sequence is clearly insufficient. This is especially true in the case of left-right HMMs, where the transient nature of the states within the model ensures that each state is associated with a relatively small number of observations - not to mention the fact that certain states are allowed to bypass their immediate successors during a transition. The use of multiple observation sequences for model optimisation therefore becomes a definite necessity.

Consequently, certain minor modifications are made to the reestimation procedure. We denote the training set of K observation sequences by

$$\bar{\mathbf{O}} = \{\mathbf{O}^{(1)}, \mathbf{O}^{(2)}, \dots, \mathbf{O}^{(K)}\}, \quad (\text{B.12})$$

where $\mathbf{O}^{(k)} = \{o_1^{(k)}, o_2^{(k)}, \dots, o_T^{(k)}\}$ represents the k^{th} sequence of T observations. Assuming independence between each observation sequence, the modified reestimation procedure entails adjusting the model parameters λ in order to maximise

$$P(\bar{\mathbf{O}}|\lambda) = \prod_{k=1}^K P(\mathbf{O}^{(k)}|\lambda). \quad (\text{B.13})$$

B.3.3 Implementation issues

In order to accurately utilise the efforts of an HMM in a real-world application, certain additional measures are required to avoid numerical anomalies. In this section, we discuss two such measures, namely *scaling* and *probability flooring*, as proposed by Rabiner (1989) and Fink (2008), respectively.

Scaling

In order to understand the necessity of a suitable scaling procedure, consider, for example, the definition of Equation B.13. Each factor represents a quantity (usually significantly) less than 1. Depending on the number of observation sequences incorporated, as well as the length of each such sequence, the probability measure yielded by Equation B.13

is generally expected to exceed the precision provided by modern computing systems, resulting in *numerical underflow*.

In order to ensure optimal accuracy, it is therefore recommended to scale any computation involving probabilities, thereby minimising loss of precision. The most popular method for algorithmic scaling involves the use of logarithms. This method not only maps particularly small values to sufficiently large ones, but also enables the use of summation rather than multiplication.

Logarithmic scaling is easily incorporated into the Viterbi algorithm by introducing the quantity

$$\phi_t(i) = \max_{q_1, q_2, \dots, q_t} [\ln P(q_t = s_i, \mathbf{O}|\lambda)], \quad (\text{B.14})$$

followed by the modifications to the Viterbi algorithm described below:

- Initialisation:

$$\phi_1(i) = \ln \pi_i + \ln b_i(o_1), \quad 1 \leq i \leq N \quad (\text{B.15})$$

- Recursion:

$$\phi_t(j) = \max_{1 \leq i \leq N} [\phi_{t-1}(i) + \ln a_{ij}] + \ln b_j(o_t), \quad \begin{matrix} 2 \leq t \leq T \\ 1 \leq j \leq N \end{matrix} \quad (\text{B.16})$$

- Termination:

$$\ln P^* = \max_{1 \leq i \leq N} [\phi_T(i)] \quad (\text{B.17})$$

In addition, Equation B.13 is also modified, such that the maximisation objective function becomes

$$\begin{aligned} \ln P(\bar{\mathbf{O}}|\lambda) &= \ln \prod_{k=1}^K P(\mathbf{O}^{(k)}|\lambda) \\ &= \sum_{k=1}^K \ln P(\mathbf{O}^{(k)}|\lambda). \end{aligned} \quad (\text{B.18})$$

Probability flooring

During model training, it is quite possible that certain states are not considered for parameter reestimation. This leads to zero-valued probabilities being associated with each unobserved event. As a result, any subsequent computations involving the model would possess a diminished solution space. Furthermore, it is highly inadvisable to assume the impossibility of a certain event, solely because it is granted zero probability by Viterbi reestimation.

In order to address this issue, the concept of *probability flooring* is introduced, whereby it is strictly required that any transitional or output probability remains greater than a certain threshold p_{\min} . This threshold value may be assigned arbitrarily, although it is recommended that it be optimised within the context of the application.

It should remain clear, though, that any zero-valued probabilities introduced by the topological constraints (see Section 4.4.3) are not subject to probability flooring. These event impossibilities may not only be safely assumed, but are in fact essential in ensuring a valid solution.

Appendix C

Haker's Algorithm

Algorithm 1: Haker's algorithm for combining two continuous classifiers C_A and C_B in ROC-space. Let \mathbf{t}_A^+ and \mathbf{t}_B^+ have length J and K , respectively.

Data: $\mathbf{t}_A^+, \mathbf{t}_B^+, \mathbf{f}_A^+, \mathbf{f}_B^+$
Result: T_{AB}^+, F_{AB}^+

```
1 for  $j = 1$  to  $J$  do
2    $t_A^+ \leftarrow \mathbf{t}_A^+(j)$ ;
3    $f_A^+ \leftarrow \mathbf{f}_A^+(j)$ ;
4   for  $k = 1$  to  $K$  do
5      $t_B^+ \leftarrow \mathbf{t}_B^+(k)$ ;
6      $f_B^+ \leftarrow \mathbf{f}_B^+(k)$ ;
7      $MLE1 \leftarrow t_A^+ t_B^+ \geq f_A^+ f_B^+$ ;
8      $MLE2 \leftarrow t_A^+ (1 - t_B^+) \geq f_A^+ (1 - f_B^+)$ ;
9      $MLE3 \leftarrow (1 - t_A^+) t_B^+ \geq (1 - f_A^+) f_B^+$ ;
10     $MLE4 \leftarrow (1 - t_A^+) (1 - t_B^+) \geq (1 - f_A^+) (1 - f_B^+)$ ;
11    if  $MLE1 \& \neg MLE2 \& \neg MLE3 \& \neg MLE4$  then
12       $T_{AB}^+(j, k) \leftarrow t_A^+ t_B^+$ ;
13       $F_{AB}^+(j, k) \leftarrow f_A^+ f_B^+$ ;
14    else if  $MLE1 \& MLE2 \& \neg MLE3 \& \neg MLE4$  then
15       $T_{AB}^+(j, k) \leftarrow t_A^+$ ;
16       $F_{AB}^+(j, k) \leftarrow f_A^+$ ;
17    else if  $MLE1 \& \neg MLE2 \& MLE3 \& \neg MLE4$  then
18       $T_{AB}^+(j, k) \leftarrow t_B^+$ ;
19       $F_{AB}^+(j, k) \leftarrow f_B^+$ ;
20    else if  $MLE1 \& MLE2 \& MLE3 \& \neg MLE4$  then
21       $T_{AB}^+(j, k) \leftarrow t_A^+ + t_B^+ - t_A^+ t_B^+$ ;
22       $F_{AB}^+(j, k) \leftarrow f_A^+ + f_B^+ - f_A^+ f_B^+$ ;
23    end
24  end
25 end
```
