# RELIABLE PATTERN RECOGNITION SYSTEM WITH NOVEL SEMI-SUPERVISED LEARNING APPROACH

CHUN LEI HE

A THESIS

IN THE DEPARTMENT

OF

COMPUTER SCIENCE AND SOFTWARE ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY AT

CONCORDIA UNIVERSITY

MONTREAL, QUEBEC, CANADA

AUGUST 2010

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

*Your file Votre référence*
*ISBN:* 978-0-494-71138-5
*Our file Notre référence*
*ISBN:* 978-0-494-71138-5

# ABSTRACT

## RELIABLE PATTERN RECOGNITION SYSTEM WITH NOVEL SEMI-SUPERVISED LEARNING APPROACH

Chun Lei He, Ph.D.
Concordia University, 2010

Over the past decade, there has been considerable progress in the design of statistical machine learning strategies, including Semi-Supervised Learning (SSL) approaches. However, researchers still have difficulties in applying most of these learning strategies when two or more classes overlap, and/or when each class has a bimodal/multimodal distribution.

In this thesis, an efficient, robust, and reliable recognition system with a novel SSL scheme has been developed to overcome overlapping problems between two classes and bimodal distribution within each class. This system was based on the nature of category learning and recognition to enhance the system's performance in relevant applications. In the training procedure, besides the supervised learning strategy, the unsupervised learning approach was applied to retrieve the "extra information" that could not be obtained from the images themselves. This approach was very helpful for the classification between two confusing classes. In this SSL scheme, both the training data and the test data were utilized in the final classification.

In this thesis, the design of a promising supervised learning model with advanced state-of-the-art technologies is firstly presented, and a novel rejection measurement for verification of rejected samples, namely Linear Discriminant Analysis

Measurement (LDAM), is defined. Experiments on CENPARMI's Hindu-Arabic Handwritten Numeral Database, CENPARMI's Numerals Database, and NIST's Numerals Database were conducted in order to evaluate the efficiency of LDAM.

Moreover, multiple verification modules, including a Writing Style Verification (WSV) module, have been developed according to four newly defined error categories. The error categorization was based on the different costs of misclassification. The WSV module has been developed by the unsupervised learning approach to automatically retrieve the person's writing styles so that the rejected samples can be classified and verified accordingly.

As a result, errors on CENPARMI's Hindu-Arabic Handwritten Numeral Database (24,784 training samples, 6,199 testing samples) were reduced drastically from 397 to 59, and the final recognition rate of this HAHNR reached 99.05%, a significantly higher rate compared to other experiments on the same database. When the rejection option was applied on this database, the recognition rate, error rate, and reliability were 97.89%, 0.63%, and 99.28%, respectively.

# ACKNOWLEDGEMENTS

It is a pleasure to thank the many people who helped in making this thesis possible. Working on my Ph.D. included not only having a study life, but also a life's study. I learned not only the expertise from my Supervisor and to share ideas and research with new friends, but also to manage life with perseverance and a smile.

Thus, I would firstly like to express my sincere gratitude to my Supervisor, Dr. Ching Y. Suen, for his inspirational instruction and persistent guidance, for his continued support and his warm encouragement. I feel extremely fortunate to have been guided by him for both my Ph.D. and Master's studies. It has been an honor to work with him.

Moreover, special thanks to Dr. Louisa Lam, a co-author of several papers. Her insights have strengthened this thesis significantly. She provided me with many inspiring comments and wise advice. I am grateful to Dr. Cheng-Lin Liu for helpful discussions during my visit to at the Chinese Academy of Sciences as a student visitor. As a Natural Sciences and Engineering Research Council (NSERC) Scholar, I want to thank NSERC for their financial support.

I am indebted to my many student colleagues and the staff at the Center for Pattern Recognition and Machine Intelligence (CENPARMI), for providing a stimulating and happy environment in which to learn and grow. I am especially grateful to Guiling Guo, Dr. Wumo Pan, Ying Li, Xiaoxiao Niu, Bing Zhang, Xiaohua Xie, Malik Waqas Sagheer, Khoa Luu, Muna G. Khayyat, Graciela Lecireth Meza Lovon, Shima Nikfal, David Yu, Huda Alamri, Puntis Jifroodian Haghighi, Mohammad Mehdi Haji, and Muhanmmad Ismail Shah. The group has been a "family" within an ideal research environment, both socially and technically. A special thank you to CENPARMI's Research Manager, Mr. Nicola Nobile, for his technical assistance and to Ms. Shira Katz for her editing and proofreading of this thesis. Also, I want to thank our Office Assistant, Ms. Marleah Blom, for her administrative assistance.

I wish to thank my husband, Hao Meng, and my son, David Meng, for all of their understanding, love, and support during these years.

Lastly, and most importantly, the greatest thank you goes out to my parents and parents-in-law. They raised me, supported me, taught me, and loved me. To them, I dedicate this thesis.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# Chapter 1

# Introduction

In this chapter, the motivation, objectives, and structure of this thesis are introduced. Section 1.1 includes the motivation, which is based on the concepts of human category learning, human cognition and recognition, and challenges in computer applications. In Section 1.2, the objectives are discussed, and the outline is described in Section 1.3.

## 1.1 Motivation

Pattern recognition, which is "the act of taking in raw data and taking an action based on the category of the data" [35], is an innate ability of animals. It has been studied in many fields, including Psychology [62] and Ethology [97].

While Artificial Intelligence (AI) achieved its greatest successes in the 1990's and early 21st century, pattern recognition/machine learning in Computer Science [35, 12, 20, 53] arose as a field of interest to researchers. These researchers endeavored to design and develop the algorithms that allow computers to simulate human beings by recognizing (classifying) patterns based either on a *priori* knowledge or on statistical information extracted from the patterns.

Due to the successes of these researchers, many applications of pattern recognition systems and techniques are available, and they cover a broad scope of fields, such as Engineering, Agriculture, Biology, Economics, Medicine, and so forth. It is even applied back to studies in Psychology/Cognitive Science and Ethology.

Even within one field of study, many applications can be used to various subjects. For example, in Computer Science & Engineering, applications can include the following topics: handwriting recognition, speech recognition, face recognition, computer vision, natural language processing, syntactic pattern recognition, classification of text into several categories (e.g. spam/non-spam email messages), search engines, object recognition in computer vision, etc.

In the last decade, researchers have aimed to train machines to automatically learn complex patterns and to make intelligent decisions by themselves. They have attempted to understand human learning that may lead to new machine learning algorithms. However, the algorithms built so far have not been able to match (or in certain cases even get close to) human performance because our machines cannot completely simulate human learning and human recognition. Thus, studies in pattern recognition, machine learning, and data mining are still very challenging in the following aspects:

- It is difficult to collect representative data;
- It is difficult to find information and knowledge regarding the

relationships among data;

- It is difficult to represent data, information, and knowledge;
- It is difficult to design models with perfect classification and discrimination. Because training sets are finite and the future is uncertain, learning theory usually does not yield absolute guarantees of the performance of algorithms.

Therefore, exploration on the concepts of human category learning and human recognition is vital and necessary since simulating human learning may improve machine learning and recognition. Thus, in our research, we will discuss human learning and human recognition in the following sub-sections.

### 1.1.1 Human Category Learning

Learning is defined as acquiring new knowledge, behaviors, skills, values, preferences or understanding, and may involve synthesizing different types of information. The ability to learn is possessed by humans, animals and some machines [49].

Since machine learning is somehow similar to infants' learning, we start our study from the concept of their category learning. Infants display complex categorization abilities. Performance in any given task might reflect prior learning or within-task learning, or both. The extent to which either form of learning is deployed can be determined by the task context [65].

Accordingly, matching to the research in Machine Learning, learning from prior knowledge is called supervised learning, while within-task learning is called unsupervised learning, and learning from both is called semi-supervised learning (SSL). In Machine Learning, supervised learning generates a function that maps inputs to the desired outputs. For example, in a classification problem, the learner approximates a function by mapping a vector into classes and by looking at the input-output examples of the function. Unsupervised learning models a set of inputs such as clustering [30]. Since the 1960s, SSL was introduced with the concept of "self-learning" [84]. SSL combines both labeled and unlabeled samples to generate an appropriate function or classifier.

It is not difficult to understand supervised learning and unsupervised learning in human category learning, so we describe only one human experiment called infant word-object fast mapping, such that infants perform semi-supervised learning in some situations.

In cognitive psychology, fast mapping is a mental process whereby a new concept can be learned (or a new hypothesis formed) based on only a single exposure to a given unit of information. Fast mapping is particularly important during language acquisition in young children, and serves (at least in part) to explain the prodigious rate at which children gain vocabulary. The phenomenon was first formally observed, and the term fast mapping coined, by Harvard researchers Susan Carey and Elsa Bartlett in 1978. They found that when children hear a new word only once, they have already developed some hypotheses about what that new word means [16].

The process of fast-mapping is described as follows:

*Subject: baby, 20 months old*

A baby is presented with 4 objects, consisting of three familiar objects, and one novel object

Familiar objects: a ball, a cup, a watch

Unfamiliar object: a pair of scissors

The baby can indicate the ball when the experimenter asks for a ball. When the experimenter asks for a "zib" (a makeup word), the baby is capable of pointing to the pair of scissors (Figure 1).

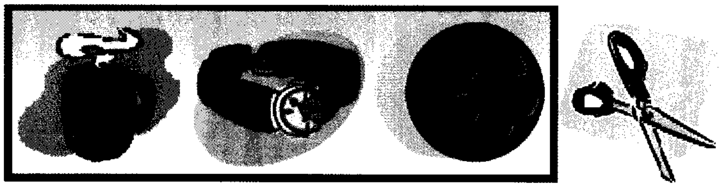


**Figure 1. Four objects in a fast mapping experiment**

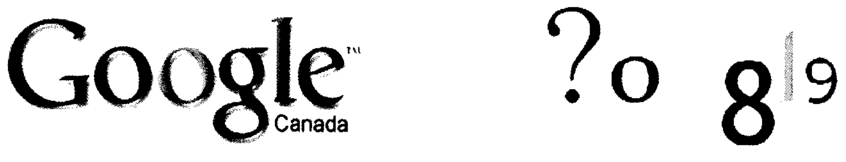
The result indicates that infants can recognize that different words refer to different kinds of things, and objects only have a single label. Therefore, new words can be used to label the unlabeled objects. Infants assume that a new word cannot be a synonym for any of the words they already know. This is similar to the learning method of cluster-then-label. Infants cluster known and unknown objects, and then match the unheard-of label to the unknown object. Thus, humans perform semi-supervised learning in some situations.

It is difficult to know which of the three methods is most suitable for Machine Learning (Supervised Learning, Unsupervised Learning, or Semi-Supervised

Learning). Learning should be task-orientated, as mentioned in [65]. Once supervised learning or unsupervised learning cannot perform with satisfactory results, semi-supervised learning should be taken into consideration. However, problems such as which part of the data should be applied with supervised learning or unsupervised learning, and how to combine these two methods in the learning procedure are major issues in machine learning.

### 1.1.2 Human Cognition and Recognition

A simple personal story may reveal or reflect how humans recognize patterns. I once opened Google's website together with my son, a two-year old boy. He started to read the "Google" logo as: "9, 1, 8, 0, 0, …" and stopped. When he read these letters, he had no prior knowledge about alphabets, and he only knew ten numerals from 0 to 9. It seems that he read the logo from right to left (easy-to-difficult), and he refused to read the Capital "G" because the Capital "G" did not look similar to any numerals. Obviously, he matched the letters to similar numerals and rejected the one without enough confidence (Figure 2).



(a) Google's logo (b) Simulation of Google's logo with numerals

**Figure 2. An example of word-numeral mapping with rejection**

There is no doubt that rejection should be part of problem-solving strategies. In addition, object information itself sometimes may not be enough for human recognition, and task constraints should be considered at the same time. These arguments have been proven in the field of psychology [2].

In psychology, some researchers have classified human problem-solving strategies as error-preventing (no response is chosen until one can be selected with a relatively high confidence) and error-correcting (a tentative solution is formulated immediately, subject to revision in the light of the subsequent evidence) [76]. These strategies match the definitions of high reliability and high recognition rate in the fields of Pattern Recognition and Machine Learning. These definitions will be provided later in this chapter.

On the other hand, in cognitive psychology, P. G. Schyns found that the recognition performance can be formulated as an interaction of task constraints and object information [83]. K. J. Malmberg [64] also mentioned that strong constraints are valuable because they expose the limitations of the models and inspire researchers to organize the models themselves.

In summary, we should find a good error-correcting "behavior" in order to facilitate our Machine Learning procedure, which includes rejection and error-correction strategies in the training procedure in order to prevent and correct errors.

### 1.1.3 Challenges in Computer Applications

By understanding the learning behaviors discussed in the previous sections, we can design related applications. Optical Character Recognition (OCR) is one of the most successful applications in pattern recognition, and it has been under investigation since the mid-1950's. In handwriting recognition, the OCR systems deal with digital images as inputs. Offline handwritten character recognition in languages such as English, Chinese, and Japanese has been researched extensively for over thirty years. However, Arabic handwriting recognition is a relatively new area of research, even though Arabic is one of the most widely used languages in the world [78].

Hindu-Arabic numerals have difficulties for recognition, even for human beings. In Figure 3, we show five samples from each of the 10 classes of Hindu-Arabic numerals from the Centre for Pattern Recognition and Machine Intelligence (CENPARMI) database [4]. For this figure, the class of the numeral is shown in the first column; its Hindu-Arabic printed form is shown in the second, followed by five examples of its handwritten form in the third column. These written samples are shown in the same vertical positions as they appear in the text lines of the database.

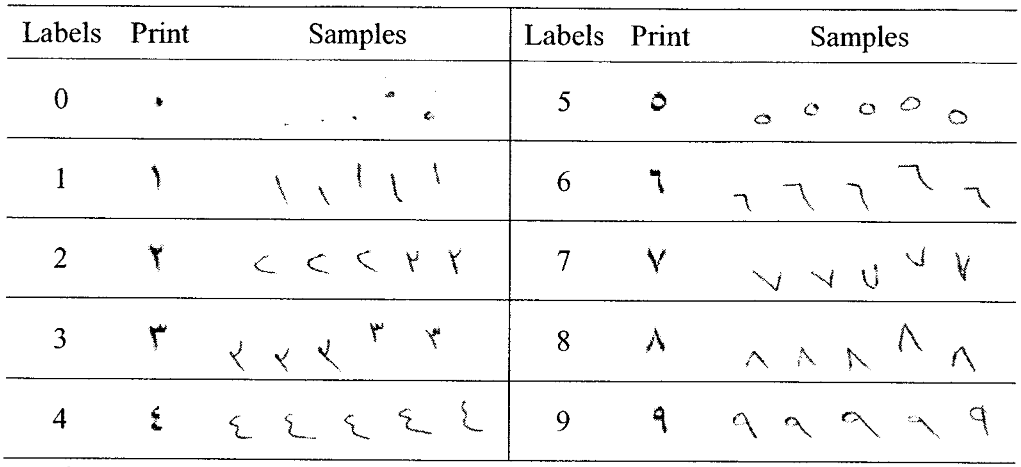

Labels Print Samples
0 ٠
1 ١
2 ٢
3 ٣
4 ٤
Labels Print Samples
5 ٥
6 ٦
7 ٧
8 ٨
9 ٩

**Figure 3. Samples from CENPARMI Hindu-Arabic Isolated Numerals Database**

In the current study, machines have most statistical learning difficulties or standard SSL difficulties when two or more classes have overlapping problems. In addition, most statistical machine learning or standard SSLs rely on another assumption that there is only one cluster in each class. However, in Hindu-Arabic Handwritten Numeral Recognition (HAHNR), the numerals two and three can look similar when written in almost the same form, as shown by some real samples in Figure 4. This similarity may account for the confusion of numerals and the lower performances when compared with handwritten numeral recognition in general [4]. Thus, we choose HAHNR as our focus for this thesis.

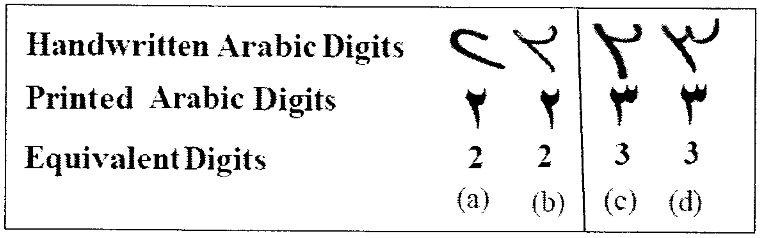

Handwritten Arabic Digits
Printed Arabic Digits ٢ ٢ ٣ ٣
Equivalent Digits 2 2 3 3
(a) (b) (c) (d)

**Figure 4. Samples of Handwritten Hindu-Arabic numerals "2" and "3"**

In fact, although people from different cultures may share the same language, they may have different habits in writing or different writing styles. For example, Palestinians may write the numeral 2 in Hindu-Arabic as (b) in Figure 4, but they never write the numeral 3 in Hindu-Arabic as (c) in Figure 4. On the contrary, Saudi-Arabians may write the numeral 3 as (c) in Figure 4, which is almost the same shape as (b).

Thus, this spatial factor is reflected in some databases, such as CENPARMI's HAHWR database. Actually, researchers have studied spatial data mining since the 1990's [48]. It is the process of discovering interesting and previously unknown, but potentially useful patterns from large spatial datasets. It is difficult to extract interesting and useful patterns from spatial datasets due to the complexity of spatial data types, spatial relationships, and spatial autocorrelation [88].

Therefore, writing styles that share the same spatial properties should be co-occurrence patterns, and they should belong to one cluster. Based on co-occurrence patterns, information can be retrieved. If two writers have the same writing style, their writings of numeral 2 and 3 should be linked by a path of high density (or a function), and then their outputs are likely to be close to each other and can be classified to the same class [25].

Hence, in addition to object information itself, the context information or writer's spatial information should be helpful for recognition. Accordingly, context information retrieval should be considered to disambiguate the confusing shapes between two overlapping classes. Accordingly, we should design an effective learning procedure to solve the problems in classification, such as overlapping in two or more classes and/or the distribution within a class is not unimodal [100]. In statistics, a unimodal probability distribution (or when referring to the distribution, a unimodal distribution) is a probability distribution which has a single mode..In this study, we apply the unsupervised learning method to solve overlapping problems and retrieve the information that cannot be done with supervised learning, and to classify samples in the rejection class.

## 1.2 Objective

In document recognition applications, it is very important to achieve high levels of accuracy as well as high reliability because even a low percentage of recognition errors can have serious consequences. For example, while OCR algorithms have resulted in recognition rates in excess of 99% on the numeral databases of MNIST [57, 108] and CENPARMI [60], the resulting low error rates can be extremely costly in applications such as the processing of financial documents. For these applications, errors should be reduced as much as possible, and it is preferable to reject some classification results in order to achieve a very high reliability while maintaining a high recognition rate as defined by:

$$Recognition\ rate = \frac{Number\ of\ correctly\ classified\ samples}{Total\ number\ of\ test\ samples} \times 100\%$$

$$Rejection\ rate = \frac{Number\ of\ rejected\ samples}{Total\ number\ of\ test\ samples} \times 100\%$$

$$Reliability = \frac{Recognition\,rate}{100\% - Rejection\,rate} \times 100\%$$

Our objectives are to design an efficient and robust recognition system to help us better understand the nature of learning and to solve these real-life/industrial problems.

Firstly, a rejection process needs to be designed such that it can be adapted to different recognition algorithms as well as datasets. As mentioned before, in human cognition, rejection is a part of problem-solving strategies. Similarly, rejection during recognition should be considered in machine learning. The reject option can be very useful in preventing excessive misclassifications in applications that require high classification reliability [37]. Rejected patterns must be manually handled or fed to a more accurate and more costly classifier. It is thus necessary to find a trade-off between rejection and misclassification rates. Moreover, before discussing the development of a system to reduce errors and achieve a high reliability, we should also study misclassified data and find ways of preventing their occurrences. Therefore, we will analyze and categorize errors in the training procedure so that we can understand the reasons for the errors and so that we can design target-oriented verifiers in testing.

The research goals of this thesis are twofold: theory and application. The theoretical part is focused on the following aspects: research on a novel semi-supervised learning scheme, a promising supervised learning system with advanced state-of-the-art technologies, an effective rejection measurement, and verifications based on error categorization and writing style retrieval. The applications use algorithms that are based on the proposed theories, to be implemented in the OCR system. The details of these goals are described below.

- **Theoretical issues:**

1. Propose a novel Semi-Supervised Support Vector Machine (S3VM) with a rejection option to improve the global performance.
2. Discover a promising supervised learning system with advanced state-of-the-art technologies.
3. Research an effective rejection measurement.
4. Analyze the rejections and categorize the errors.
5. Verify the rejected patterns based on error categorization and based on writing style retrieval by unsupervised learning.

- **Algorithm issues:**

1. To implement a supervised learning system with advanced state-of-the-art technologies.
2. To implement a rejection measurement with Linear Discriminant Analysis principles.
3. To implement different pair-wise verifiers based on different error categories.
4. To implement unsupervised learning on the test set in order to retrieve the writers' writing styles.
5. To implement a Semi-Supervised Support Vector Machine with a rejection class in order to pursue the highest recognition rate with a lowest error rate.

In S3VM, we propose to apply unsupervised learning to retrieve some extra information as a form of to compensation to the classification result from the supervised learning procedure. Traditionally, researchers have applied the unsupervised learning method on the test data to help re-locate the boundary between a pair of classes. However, in this thesis, we apply the unsupervised learning method to retrieve the writers' Writing Styles, so that the rejected data in the test set can be re-classified according to their writing styles. This method of retrieval cannot be achieved with supervised learning in the training procedure. This approach can also be adapted for other pattern recognition contexts to distinguish between classes of highly similar patterns.

## 1.3 Outline of the Thesis

In this thesis, we focus on a domain-specific problem by designing semi-supervised learning algorithms with a rejection option using large data sets. We apply the supervised learning method to classify the samples with a rejection option. We verify the results with the unsupervised learning method and other strategies in order to retrieve the information that cannot be done with supervised learning. This thesis is organized into eight chapters, described below.

- In Chapter 2, we review the studies on different related topics, including general Semi-Supervised Learning modules, handwritten Hindu-Arabic numeral recognition systems, rejection measurements, error categorization, handwritten numeral verification, and recognition with writing Adaptation/writing style adaptive information. In addition, we point out the difficulty of applying the existing Semi-Supervised Learning methodologies for handwritten Hindu-Arabic numeral recognition systems.

- In Chapter 3, we introduce the theory of the standard Support Vector Machine (SVM), SVM with a rejection option (RO-SVM), and Semi-supervised SVM with a rejection option (RO-S3VM). Moreover, the framework of this thesis is illustrated.
- In Chapter 4, we propose a standard recognition system with supervised learning. We apply some state-of-the-art technologies for the recognition. Although the key technologies such as pre-processing, feature extraction, and the design of classifiers are all existing methods, satisfactory recognition results were achieved when we used this standard recognition system in this thesis. Moreover, the state-of-the-art performances on several databases are described.
- In Chapter 5, we define a novel rejection measurement called LDA Measurement (LDAM). This LDAM is designed to take into consideration the confidence values of the classifier outputs and the relations between those values. We compare LDAM to the rejection measurements of First Rank Measurement (FRM) and First Two Ranks Measurement (FTRM), and then we describe the experiments and compare the results obtained from using these three measurements, with outputs that can represent distances or probabilities from different

classifiers. The results show that the use of LDAM is more optimal than FRM and FTRM in producing reliable recognition results.

- In Chapter 6, we categorize errors and design target-oriented strategies for verification. We firstly analyze errors from the Training Set and divide those errors into four categories and figure out the corresponding strategies for verification. The experiments and error analysis after verification are described as well.
- In Chapter 7, we propose the Writing Style Verification (WSV) method based on applying the unsupervised learning method on the test set. We define a Confusing Pair (CP) of clusters and a Writing Style (WS), and devise a methodology to automatically detect a CP and WS with unsupervised learning. The experiments and error analysis based on writing style verification are also described.
- Finally, we summarize this thesis in Chapter 8 with some concluding remarks.

# Chapter 2

# Literature Review

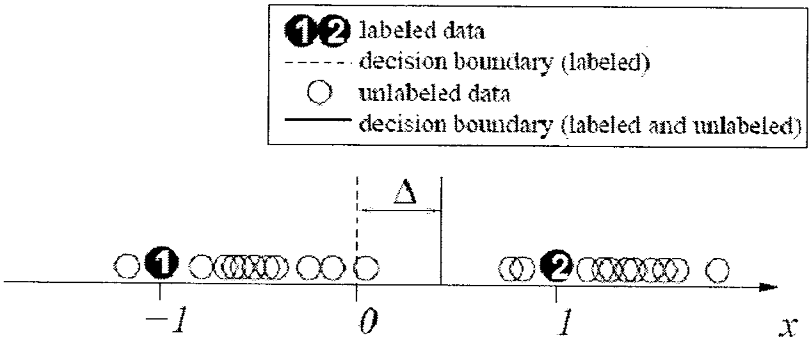
In this chapter, we review the general Semi-Supervised Learning modules, handwritten Hindu-Arabic numeral recognition systems, rejection measurements, error categorization, handwritten numeral verification, and recognition with writer adaptive/writing style adaptive information, respectively. In addition, we describe the problems encountered by the existing Semi-Supervised Learning methodologies in handwritten Hindu-Arabic numeral recognition systems.

## 2.1 Semi-Supervised Learning (SSL)

In the literature, a number of learning strategies have been proposed for various underlying classifiers and applications. In this section, we review SSL strategies and analyze the difficulties in recognizing Hindu-Arabic numerals with the existing SSL models. Finally, we redefine SSL in a more general and broader way.

Semi-supervised learning is a learning method that falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). It is a machine learning technique that makes use of both labeled and unlabeled data for making predictions. Semi-supervised learning attempts to take advantage of this state by using the available labeled data as known examples of mappings while still looking at the unlabeled data to learn even more. In general, unlabeled data can help to adjust (optimize) the boundary determined by both labeled and unlabeled data.

Mostly, researchers have used unlabeled data in conjunction with a small amount of labeled data to produce considerable improvements in learning accuracy. Since the cost associated with the labeling process is too high, and the acquisition of unlabeled data is relatively inexpensive, semi-supervised learning has a great practical value. Here is an example of how unlabeled data can help classification: assuming each class is a coherent group (e.g. Gaussian), the decision boundary will shift to a solid line (Figure 5) [111] once the unlabeled data involve in classification.



❶❷ labeled data
----- decision boundary (labeled)
○ unlabeled data
—— decision boundary (labeled and unlabeled)
Δ
−1
0
1
x

**Figure 5. Different decision boundaries with and without unlabeled data**

Since the 1960s, Semi-Supervised Learning (SSL) started with the concept of "self-learning" [84]. Self-training has been applied to several natural language processing tasks. Yarowsky uses self-training for word sense disambiguation, e.g.

deciding whether the word 'plant' means a living organism or a factory in a given context [107]. Riloff et al. uses self-training to identify subjective nouns [77]. Maeireizo et al. classify dialogues as 'emotional' or 'non-emotional' with a procedure involving two classifiers [63].

The model assumption plays an important role in semi-supervised learning. It makes up for the lack of labeled data, and can determine the quality of the predictor. In general, there are several existing SSL models involving to different assumptions. For instance, there is a generative model which is a probabilistic model with two Gaussian distributions learned with Expectation Maximization (EM) [17]; a semi-supervised support vector machine which assumes that the decision boundary should not pass through dense unlabeled data regions [103]; and a graph-based model, with a typical way to generate the graph, such that any two instances in the labeled and unlabeled data are connected by an edge [51]. The model assumption is that instances connected with large-weight edges tend to have the same label.

We provide an example with different assumptions on an overlapping problem [111]: consider a classification task where there are two classes, each with a Gaussian distribution. The two Gaussian distributions heavily overlap (top panel of Figure 6). The true decision boundary lies in the middle of the two distributions, shown as a dotted line. Samples (instances) in five Training sets (Training set 1 to Training set 5) in Figure 6 are randomly drawn from the two overlapping classes.

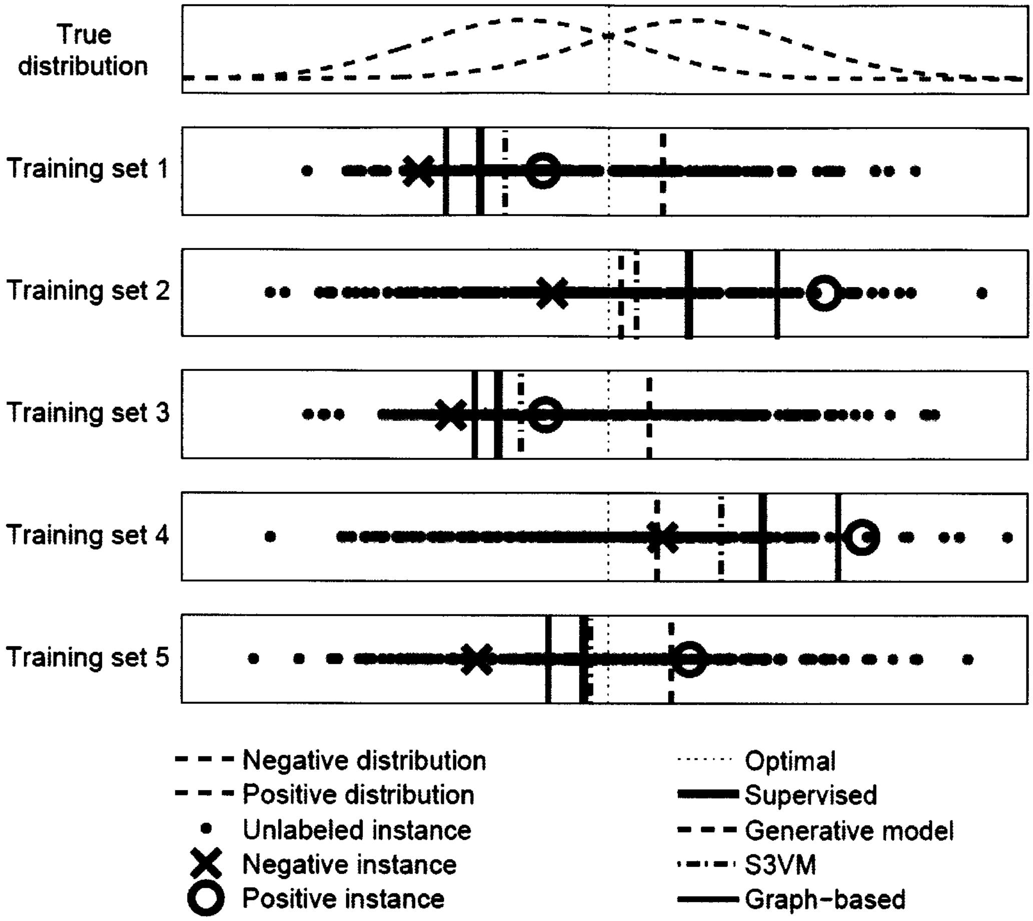


**Figure 6. Decision boundaries learned by several algorithms for two overlapping classes [111]**

For supervised learning, the learned decision boundary is in the middle of the two labeled instances, and the unlabeled instances are ignored. In Figure 6, the thick solid line in Training set 1 is the decision boundary with supervised learning, which is located away from the true decision boundary because the two labeled instances are randomly sampled. If we were to draw two other labeled instances, the learned decision boundary would change, but most likely would still be located incorrectly (see Training set 2 to Training set 5 of Figure 6). On average, the expected learned decision boundary will coincide with the true boundary, but for any given drawing of labeled data, it will be off quite a bit. We can say that the learned boundary has a high variance.

To evaluate supervised learning, and the semi-supervised learning methods introduced before, we examined 100 training samples, each with one labeled and 99 unlabeled instances per class. Now without presenting the details, we show the learned decision boundaries of three semi-supervised learning models for the training data.

The first one is a probabilistic generative model, shown in Figure 6 as dashed lines. In this case, the boundaries tend to be closer to the true boundary and similar to one another, i.e., this algorithm has a low variance. The second model is an S3VM, which assumes that the decision boundary should not pass through dense unlabeled data regions. However, since the two classes strongly overlap, the true decision boundary actually passes through the densest region. The learned decision boundaries are shown in Figure 6 as dash-dotted lines. The third approach is a graph-based model, with a typical way to generate the graph such that any two instances in the labeled and unlabeled data are connected by an edge. The edge weight is large if the two instances are close to each other and small if they are far apart. However, in this particular example, where the two classes overlap, instances from different classes can be quite close and connected by large-weight edges. The results produced by this model are shown in Figure 6 as thin solid lines.

In this example, although the generative model and S3VM are more accurate and more stable than the supervised model, the error rates on the 100-trial average test samples for these algorithms is still around 30%. Thus, on overlapping problems, neither the standard supervised algorithms nor SSLs can yield optimal solutions in classification. If and only if we retrieve extra information rather than focusing only on object image information, overlapping problems may be solved, and confusing samples can be distinguished. However, overlapping problems often occur in challenging applications, such as handwriting. Incorporating the diversity of writing styles into a single model leads to over-generalization, therefore it is useful to study a new model to retrieve the writers' writing styles.

In addition, there is another assumption in SSL that there is only one cluster in each class. However, in handwriting, writers may write with different writing styles, and it is possible to have more than one cluster in each class. When algorithms are presented with samples of writing by a single writer to be analyzed (for example, for recognition), the model is not as efficient in terms of accuracy as a model trained specifically to that writer's style. If training is not performed according to the writer's style, the performance will not be ideal [8]. Since the learning takes place concurrently with the ultimate desired task (e.g. recognition), modifications to the standard approaches need to be made.

Fortunately, we can apply supervised learning to modeling and then use unsupervised learning for the retrieval of the extra information, such as writing styles, etc., and to verify the results of supervised learning. We should make the right assumptions in semi-supervised learning rather than directly applying any existing models.

Therefore, we redefine SSL with a more general definition. Unlabeled data should not only be used for modeling but also for retrieving extra information that cannot be obtained from the labeled data. This learning procedure with both modeling of the labeled data and extra information obtained/retrieved from the unlabeled data is called Semi-supervised Learning.

## 2.2 Hindu-Arabic Handwritten Numeral Recognition (HAHNR) & Verification

In this section, we review the Arabic Databases in the literature. Offline handwritten character recognition in languages such as English, Chinese, and Japanese has been researched extensively for over thirty years. However, Arabic handwriting recognition is a relatively new area of research, even though Arabic is one of the most widely used languages in the world [78]. There are a few databases consisting of Arabic handwriting. For instance, the IFN/ENIT databases [73], developed in 2002, consist of 26,549 images of Tunisian town/village names written by 411 writers. Another database is the AHDB database [6], which contains words frequently used in legal amounts on Arabic checks, together with some other frequently used Arabic words. At the Centre for Pattern Recognition and Machine Intelligence (CENPARMI), a number of Arabic Script databases have been developed. Al-Ohali et al. developed an Arabic check database for research on the recognition of Arabic handwritten checks in 2000 [3]. The data includes Arabic legal amounts and Arabic sub-words presented in checks. Solimanpour et al. designed a

Farsi database consisting of Farsi isolated digits, numeral strings, letters, legal amounts, and dates [89]. Recently, Alamri et al. developed the CENPARMI Arabic database, which contains isolated Hindu-Arabic numerals, numeral strings, Arabic isolated letters, and Arabic words [4]. This database was compiled by including the samples from many writers of different genders, ages, educational levels and nationalities, with both left-handed and right-handed writers. The experiments reported in this thesis were conducted on the isolated numerals from this database.

In order to achieve a high level of accuracy, researchers have explored different methodologies in different stages of pattern recognition. For example, in the pre-processing stage, normalization, filtering, segmentation, and thinning, etc., are commonly adopted so that image qualities are enhanced. In feature extraction, multi-features, such as those based on zones, directions, and structures, etc., are commonly used, combined or selected in order to reduce the dimension of the data while extracting or maintaining the relevant information. For the purpose of satisfying the requirement of high reliability, the classifiers must perform with minimal errors, or eventually be free from errors. In classification, the methods of supervised learning, unsupervised learning, and even semi-supervised learning have been commonly applied.

The verification of confusing handwritten numeral pairs is a challenging task because the confusing character pairs look quite alike in terms of the features used in classification or in terms of their shapes. There are four types of verifiers according to the number of classes. Let $\Omega$ denote the working space of a verifier, and let $|\Omega|$

denote the dimension of the space. The four verifiers are:

- $|\Omega| = n$: General verifier, working on all classes in the problem.
- $0 < |\Omega| < n$: Cluster verifier, with verification of clustered categories, e.g. (Is it a "2", "3", or "4"?).
- $|\Omega| = 2$: Pair-wise verifier, with verification between two categories, e.g. (Is it a "2" or "3"?).
- $|\Omega| = 1$: Class-specific verifier, working on one candidate class, e.g. (Is it a "2"?).

Due to the error analysis in the training set for HAHNR, we found that most errors occurred between a pair of classes. Thus, we designed verifiers between pairs of classes, for example, classes "2" v.s. "3" and classes "0" v.s. "1".

## 2.3 Rejection Measurement

In the literature, a number of rejection strategies have been proposed for various underlying classifiers and applications. In this section, we review the state-of-the-art rejection strategies that have been implemented by various offline handwriting recognition systems, including strategies that make use of different levels of classifier outputs.

### 2.3.1 Outputs from Classifiers

Generally speaking, classification algorithms supply outputs at three levels [40]:

1) The abstract level: a classifier *e* outputs a likely unique label/class *j*; or in

some extensions, $e$ outputs a subset $J \subset \Lambda$, where $\Lambda$ is the set of all classes.

2) The rank level: $e$ ranks all the labels in $\Lambda$ (or a subset $J \subset \Lambda$) according to the likelihoods that the input sample $x$ has those labels.

3) The measurement level: $e$ attributes to each label in $\Lambda$ a measurement value. This measurement can be a probability that $x$ has that label, or the distance of $x$ from the class having that label.

Among the three levels, the measurement level provides the most information, and the abstract level provides the least amount of information since both ranks and measurements are provided in measurement level. From the measurement attributed to each label, we could rank all the labels in $\Lambda$, in ascending or descending order. By choosing the label at the top rank, or by directly choosing the label with the maximal or minimal value at the measurement level, we can assign a unique label to $x$. In other words, from the measurement level to the abstract level, there is an information reduction or abstraction process. On the other hand, when classification provides only abstract level outputs, it is difficult to design a rejection strategy.

In this thesis, both support vector classifiers, HeroSVM [28] and LibSVM [18], provide the measurement level outputs.

In HeroSVM, the outputs represent the distances between the input vector and the margins of each class. HeroSVM[1] is a fast and high-performance SVM software package that introduces a parallel optimization step to quickly remove most of the nonsupport vectors, and that applies an effective integration of kernel caching and

[1] Available at: http://www.cenparmi.concordia.ca/~jdong/HeroSvm.html.

kernel matrix computation for classification. The strategy applied in multi-class problems is to consider one class against all the others [110]. Taking the training samples with one label as one class and all others as the other class, the procedure is reduced to a two-class problem. For $k$ classes of data ($k > 2$), $k$ SVM classifiers are formed and denoted by SVM$_i$, *i=1,2, ...k*. For the test sample $x$, $d_i(x) = w_i \cdot x + b_i$ can be obtained by using SVM$_i$, where $d_i$ is the decision function for class *i,* $w_i$ is a normal vector, perpendicular to the hyperplane that separates class $i$ from all the other classes, and the parameter $b_i$ is the distance from the origin to the hyperplane along the normal vector $w_i$. The test sample $x$ is considered to belong to the $j$th class where$d_j(x) = \max_{i=1,2,\ldots,k} d_i(x)$. .

Unlike HeroSVM, the classification outputs of LibSVM represent probabilities. LibSVM[2] is an implementation of SVM which applies a one-against-one (or pairwise) strategy in multi-class problems. With the pairwise approach, $k^2$ support vector machines are trained for a $k$-class problem. Given $k$ classes of data and any test sample $x$, the goal is to estimate $p_i$ (the probability that $x$ belongs to class $i$), which is obtained from $r_{ij}$ (*i, j=1,2, ...k*). $r_{ij}$ is a one-against-one class probability obtained from the known training data by solving the following optimization problem:

$$\begin{aligned} &\min_p \tfrac{1}{2}\sum_{i=1}^{k} \sum_{j:j\neq i} (r_{ji} p_i - r_{ij} p_j)^2 \\ &\textit{subject to} \sum_{i=1}^{k} p_i = 1,\ p_i \geq 0, \forall i, \end{aligned} \quad (1)$$

where $p_i = p(y = i \mid x)$, for class label $y$ of $x$, and $r_{ij} \approx p(y=i \mid y=i \; or \; j, x)$.

[2] Available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm

### 2.3.2 Rejection Strategies for Offline Handwriting Recognition

A recognition rule can be considered optimum if for a given recognition rate, it minimizes the error rate (error probability) and places the testing candidates into a reject category when their identities cannot be established with a high confidence [22]. When a feature vector has the highest conditional probability for the correct class and low conditional probabilities in all other classes, it should be accepted; otherwise it should be rejected. Rejections can be applied in single classifiers as well as Multiple Classifier Systems (MCSs) [43] in order to increase the reliability of the recognition results. Various researchers who developed handwriting recognition systems for offline handwritten numerals [19, 108], characters [74], words [54], and text lines [11], as well as check processing [39] and address reading systems [14] have explored rejection methodologies in their implementations.

Achieving both high recognition and high reliability requires methods capable of assigning generally higher confidences to correct recognition results rather than to incorrect ones. This confidence scoring method may consist of implementing a simple function with appropriate parameters drawn directly from the recognition process, or it may be a learning task in which a classifier is trained to use an array of parameters to distinguish correct recognitions from misclassifications [74].

In general, the recognizer estimates posterior probabilities for the various classes, so it is possible to make optimal (Bayesian) decisions by comparing the probabilities of samples to a threshold (so that probabilities below the threshold will result in rejections). Generally, rejection strategies can be divided into two categories: absolute and relative rejections.

In absolute rejections, only the top choice among the outputs (called FRM in this thesis) is used as a criterion for rejection. This strategy has been implemented in handwritten numeral recognition [19] and in character recognition [74]. In the latter work, a variety of scoring functions were evaluated and explored, including the "raw" recognition score, and a sample was assigned to the class with the highest score, provided that this score was large enough.

In relative rejections, the relationship between various confidence measurements is taken into consideration. Examples of such relationships include the likelihood ratio (ratio of the highest and second-highest confidence values) and the estimated posterior probability (ratio of the highest confidence value to the sum of all confidence values) [74]. Other examples include class-dependent and hypothesis-dependent thresholds, since they consider the average of certain confidence measurements or the difference between the top two confidence values, as applied to word recognition in [54]. The distance between the first and second choices is used as a rejection criterion in handwritten numeral recognition [19] and in a German address reading system [14]. This is called First Two Ranks Measurement or FTRM in this thesis.

Rejections have been applied to recognition systems with a single classifier or multiple classifiers. In [19], a rejection strategy for convolutional neural network models is proposed. In [39, 74], all confidence measures are used as inputs to a

Multi-Layer Perceptron to finalize the result of recognition. Hidden Markov Models (HMMs) have been used in error-rejection of a word recognition system [11, 14, 54]. In Dong et al., rejection with the FRM is used for a Support Vector Machine (SVM) [28].

In MCSs, cooperation is placed in a sequential (as opposed to a parallel) architecture [40]. With this topology, classifiers can be applied in succession, with each classifier producing a reduced set of possible classes for each pattern, so that the individual classifiers or experts can become increasingly the main focus [56]. In handwritten numeral recognition, Zhang et al. implemented the rejection in a cascade ensemble classifier system, which is a sequential combination of multiple classifier ensembles [108]. Rejections from one layer of classification are applied as input to the next layer's classifier. The relationship of the error, rejection, and recognition rates of each Multi-Layer Perceptron classifier is analyzed with the use of Bayesian probability theory. In [36], after linearly combining four types of classifiers with a posteriori probabilities estimation, the absolute rejection strategy with FRM is applied.

In this thesis, we design and implement a method that applies Linear Discriminant Analysis (LDA) [32] to the measurement level outputs of a classifier, in order to determine an optimal threshold for the rejection option. LDA is a supervised classification method, widely used to find an optimal linear combination of features for separating two or more classes. The main idea of LDA is to project high-dimensional data onto a line and perform classification in this one-dimensional space. LDA provides a linear projection of the data with the outcome of maximum between-class variance and minimum within-class variance. Since this discriminative method can find the feature space that can best discriminate an object from others, LDA has been successfully used in pattern classification applications including Chinese character recognition [38], face recognition [10, 96], image retrieval [95], tracking [59], and marketing [ 26].

## 2.4 Error Categorization

In Plato's *Timaeus* [7], Plato stated the principle of causality in 1888: "everything that becomes or changes must do so owing to some cause; for nothing can come to be without a cause." Accordingly, errors should happen with certain causes that may help to prevent the errors from happening in the future. Thus, error analysis in the training procedure should help in avoiding or reducing errors in testing.

In fact, errors should not be treated equally, but conditionally. In standard learning algorithms [37], most researchers assume a constant error cost for all errors, and only the accuracy and error rates are considered. Accordingly, the classifiers usually try to minimize the number of errors they will make in dealing with new data. Such a setting is valid only when the costs of different errors are equal. Unfortunately, in many real-world applications, the costs of different errors are often unequal. For example, in a medical diagnosis, the cost of erroneously diagnosing a patient to be healthy may be much higher than that of mistakenly diagnosing a healthy person as being sick, because the former kind of error is more likely to result in the loss of a life. Accordingly, most costs of errors are conditional. Thus, errors should be categorized, and we must be able to deal with some missing information in classification.

Although some researchers have given the definitions of error categories [99], they may have had some difficulties to obviously match all misclassification errors into a certain error category based on this categorization. For example, Turney defined taxonomy of the costs in Inductive Concept Learning (ICL) [31, 98] and defined four error categories due to the different costs of misclassification errors:

I) Error cost conditional on time of classification,

II) Error cost conditional on individual case,

III) Error cost conditional on feature value,

IV) Error cost conditional on classification of other cases.

However, in offline handwriting recognition, since the time property of samples is not recorded, the Error category I should be re-defined or its correlation to the application should be found.

In addition, even if errors can be categorized correctly, strategies to reduce these errors based on their categories should be studied and designed. For instance, in offline handwriting numeral recognition, Suen et al. summarized misclassification errors into three categories [94]: errors with confusing natures, errors that humans have difficulty in identifying, and errors that are easily recognized by humans. However, it is difficult to identify the errors which cannot be recognized by human beings. Strategies regarding different error categories should be designed and should have the ability to be transplanted into different applications so that the cost of instability in a learning system can be reduced.

In this thesis, we categorize all the errors from a standard recognition system based on different costs of misclassification errors, and verify the recognition results with different strategies for different error categories. Because most samples can be classified correctly, it is redundant to verify all the recognition results. Instead, rejection based on classification should be applied, and verification should be done only on the rejected samples. Since there are difficulties in Hindu-Arabic Handwritten Numeral Recognition (HAHNR) of some samples, even for human beings, we propose to categorize errors in an HAHNR system, and design corresponding strategies to reduce errors in different categories.

## 2.5 Recognition with Writing Adaptation/Writing Style Adaptation Information

Ambiguous shapes that result in confusing pairs of handwritten characters often cause irreducible errors in the recognition process. In handwriting recognition, some researchers have applied different strategies to distinguish between confusing pairs. For example, Zhang et al. designed a method based on multi-modal discriminant analysis in order to reduce the feature dimensionality and to verify the recognition result of handwritten numerals within confusing pairs [108], while Rahman et al. [75] applied combinations of multiple experts to the confusing pairs. However, these methodologies could not solve the problems in HAHNR effectively due to the overlapping of shapes between classes 2 and 3. More information (besides shapes) should have been extracted so that samples in these two classes could be classified correctly.

In fact, some researchers have applied the writer's personal information/writing information in handwriting recognition both in the context of online recognition [23, 50, 92] and offline recognition [29, 67, 104].

Online handwriting recognition can use writer adaptation to create personalized systems by implementing supervised learning. Researchers were able to use a small amount of personalized training data to reduce the error rate in their systems. Hand-held devices, for example, can go through a training process to better recognize a writer's handwriting. As mentioned in [93], "for the users who will make extended use of such a system the gain in productivity due to increased accuracy will offset the initial inconvenience of training." Senior and Nathan [85] were able to use a much smaller set of training words (as few as five) in order to reduce the error rate. Connell and Jain identified character styles (lexemes) of individual writers, and specialized the lexeme model to match the writer's training data in order to deal with limited training data [23]. More recently, Huang et al. utilized a writer-dependent system in online handwriting recognition with Incremental Linear Discriminant Analysis (ILDA) in [50], while Vuori clustered writing styles in an online model for over 700 objects with a self-organizing map [105].

On the other hand, offline handwriting recognition models can also be adapted as their independent models with relatively few words. For example, Vinciarelli and

Bengio noted that they were able to adapt a writer-independent system with 30 words [104]. Nosary et al. used the recognition output from their system as training data, using batch adaptation as the recognition progressed [67]. In batch adaptation, the system's recognition output is stored and used at a later stage. In [29], a writer adaptive training method is proposed with a character-dependent Hidden Markov Model (HMM) in offline Arabic word recognition, so that writers' writings can be learned in training and utilized in testing.

Therefore, if we can retrieve writers' writing styles in our offline handwriting recognition system, then the system's performance should be enhanced, and the writing styles should be helpful to disambiguate the confusing shapes between two overlapping classes.

# Chapter 3

# Theory & Framework

In this chapter, we introduce the theories of standard Support Vector Machine (SVM), SVM with Rejection Option (RO-SVM), and Semi-Supervised SVM with Rejection Option (RO-S3VM). Moreover, the framework of this thesis is illustrated.

In Section 3.1.1, we try to keep the presentation of SVM in a self-contained way to ensure that this information can be easily understood for the interested readers who may not work directly in the machine learning and pattern recognition domain. Concepts for SVM such as margin and dual representation are introduced first, followed by the explanation of a soft margin classifier to handle non-separable cases in the original space. Afterwards, nonlinear SVM is introduced, which applies a kernel to enhance the separable capability while keeping the computational efficiency. Then, we explain the generation theory of SVM. After that, theories of SVM with Rejection Option (RO-SVM) and Semi-Supervised SVM with Rejection Option (RO-S3VM) are introduced as extensions of SVM's theory (in Section 3.1.2). In this RO-S3VM, the margins for the generative models are the same as the ones in RO-SVM, and the unsupervised learning detects only the extra information for final classification results. Thus, it was not necessary to introduce RO-S3VM in theory again. Finally, the framework of this thesis including in both training and testing procedures is described in Section 3.2.

## 3.1 Theory of Semi-Supervised SVM with Rejection Option (S3VM-RO)

This section gives a brief introduction to Support Vector Machines (SVMs) and provides readers with a basic background for SVM with Rejection Option (RO-SVM) and Semi-Supervised SVM with Rejection Option (RO- S3VM), which are both described in Sections 3.1.2 and 3.1.3, respectively. Readers who have a good background in SVM can skip Section 3.1.2 and go to the next one.

### 3.1.1 Support Vector Machines (SVMs)

Suppose we are given a dot product space $\mathcal{H}$, and a set of pattern vectors $x_1, \ldots, x_l \in \mathcal{H}$. Any hyperplane in $\mathcal{H}$ can be represented as:

$$\{x \in \mathcal{H} | \langle w, x \rangle + b = 0\}, w \in \mathcal{H}, b \in \mathbb{R}^d.$$

where $w$ is a vector orthogonal to the hyperplane. The hyperplane splits the input space $\mathbb{R}$ into two half spaces which correspond to the inputs of two classes. Figure 7 illustrates a hyperplane for separating the data set into two classes.

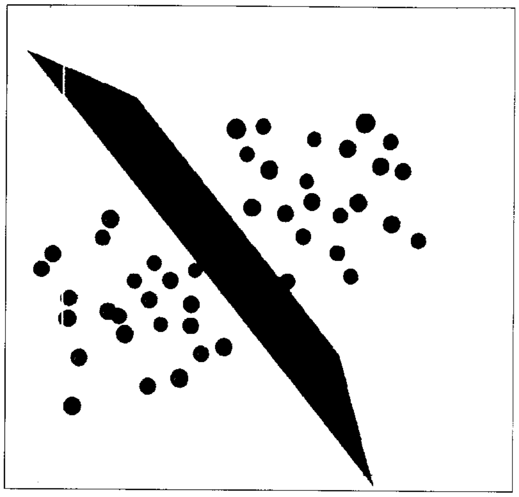


**Figure 7. A hyperplane for separating the data set into two classes**

Accordingly, we provide the definitions of a linearly separable data set and a canonical hyperplane:

**Definition 3.1. (Linearly separable data set)** [9] Given that training samples $\{(x_i, y_i)\}\epsilon X \times Y$, where $X \subseteq \mathbb{R}^d$, $Y = \{-1, 1\}$ and $i = 1, \dots, l$. The data is linearly separable if a hyperplane exists such that $y_i(\langle w, x \rangle + b) > 0$.

**Definition 3.2. (Canonical hyperplane)** [82] The pair $(w, b)$ is called a canonical form of the hyperplane with respect to $x_1, x_2, \dots, x_l$, if it is scaled such that:

$$\min_{i=1,\dots,l} |\langle w, x_i \rangle + b| = 1, \tag{2}$$

which indicates that the points closest to the hyperplane have a distance of $1/\|w\|$.

Now, we need to find the optimal hyperplane with the maximal margin. The problem can be formulated as a linearly constrained quadratic programming problem as follows:

$$\min_{w,b} \tfrac{1}{2}\|w\|^2 \text{ subject to } y_i(\langle w, x_i \rangle + b) \geq 1, i = 1, \dots, l. \tag{3}$$

This is a convex quadratic programming problem since the objective function is convex and these points which satisfy the linear constraints define a convex set. By introducing positive Lagrange multipliers $\alpha_i, i = 1, \dots, l,$ one for each of the inequality constraints, we define the Lagrangian function as follows:

$$L_P(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i [y_i(\langle w, x_i \rangle + b) - 1] \quad (4)$$

In linear programming, the *primary problem* and the *dual problem* are complementary. A solution to either one determines a solution to both, so we can solve the equivalent dual problem [33] with the following formula: Maximize $L_P$ subject to the constraints such that the gradients of $L_P$ with respect to $w$ and $b$ vanish, and subject to the constraints such that $\alpha_i \geq 0$:

$$\frac{\partial L_P}{\partial w} = 0, \quad (5)$$

$$\frac{\partial L_P}{\partial b} = 0, \quad (6)$$

$$\alpha_i \geq 0. \quad (7)$$

From Eqs.(4) and (5), we have

$$w = \sum_{i=1}^{l} y_i \alpha_i x_i, \quad (8)$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0. \quad (9)$$

We substitute the equality constraints in Eqs. (5) and (6) into Eq. (4), to give the dual formulation together with the constraints of $\alpha_i$:

$$\text{maximize } L_D(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j \, y_i y_j \langle x_i, x_j \rangle \quad (10)$$

$$\text{subject to } \alpha_i \geq 0,$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0.$$

After solving $\alpha_i$ in the dual problem, the decision function can be written as:

$$\begin{aligned} f(x) &= sgn(\langle w, x \rangle + b) \\ &= sgn\left(\sum_{i=0}^{l} \alpha_i y_i \langle x_i, x \rangle + b\right), \quad (11) \end{aligned}$$

where

$$sgn(u) = \begin{cases} 1 & if \quad u > 0 \\ -1 & otherwise \end{cases} . \qquad (12)$$

It can be observed in the dual problem (10) and in the decision function (11) that training vectors $x_i$ only occur in the form of a dot product.

When data cannot be perfectly separated due to noises and outliers, slack variables are introduced to allow the margin inequality constraints [24] in the primal problem (4) to be violated:

$$\text{subject to } y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i, i = 1, \dots, l, \qquad (13)$$

$$\xi_i \geq 0, i = 1, \dots, l.$$

When an error occurs, $\xi_i$ is greater than 1. Then, $\sum_i \xi_i$ can be regarded as the upper bound of training errors. It is expected to maximize the margin and minimize the training errors. The primal problem (4) can be re-defined as:

$$\min_{w,b,\xi} \tfrac{1}{2}\|w\|^2 + C \textstyle\sum_i \xi_i \qquad (14)$$

$$\text{subject to } y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i, i = 1, \dots, l,$$

$$\xi_i \geq 0, i = 1, \dots, l.$$

This is still a convex quadratic programming problem and the positive parameter $C$ is chosen by the user. A large $C$ represents a higher penalty to the training errors. The corresponding Lagrangian of (14) is:

$$L_P(w, b, \xi, \alpha, \beta) = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

$$- \textstyle\sum_{i=1}^{l} \alpha_i [y_i(\langle w, x_i \rangle + b) - 1 + \xi_i] - \sum_{i=1}^{l} \beta_i \xi_i \qquad (15)$$

with $\alpha_i \geq 0$ and $\beta_i \geq 0$. The Karush-Kuhn-Tucker (KKT) optimality conditions [55] are given by:

$$\frac{\partial L_P}{\partial w} = w - \sum_{i=1}^{l} y_i \alpha_i x_i = 0, \tag{16}$$

$$\frac{\partial L_P}{\partial \xi_i} = C - \alpha_i - \beta_i = 0, \forall i \tag{17}$$

$$\frac{\partial L_P}{\partial b} = -\sum_{i=1}^{l} \alpha_i y_i = 0, \tag{18}$$

$$y_i(\langle w, x_i \rangle + b) - 1 + \xi_i \geq 0, \forall i \tag{19}$$

$$\alpha_i [y_i(\langle w, x_i \rangle + b) - 1 + \xi_i] = 0, \forall i \tag{20}$$

$$\beta_i \xi_i = 0, \forall i \tag{21}$$

$$\alpha_i \geq 0, \forall i \tag{22}$$

$$\beta_i \geq 0, \forall i \tag{23}$$

$$\xi_i \geq 0. \forall i \tag{24}$$

where Eqs. (20) and (21) are called KKT "complementary" conditions. Substitute Eqs. (16), (17), and (18) into Eq. (15) and obtain the dual objective function:

$$L_D(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j \, y_i y_j \langle x_i, x_j \rangle \tag{25}$$

which is the same as that in the maximal margin case. The difference is that from the constraint (17), we obtain $\alpha_i \leq C$ since $\beta_i \geq 0$. Therefore, the dual formulation in the soft margin case is given by:

$$\text{maximize } L_D(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j \, y_i y_j \langle x_i, x_j \rangle, \tag{26}$$

$$\text{subject to } 0 \leq \alpha_i \leq C, i = 1, \dots, l,$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0.$$

The decision function in Eq. (11) is a linear function of the data. Its limitation motivates researchers to generalize to the nonlinear case. It can be observed that the data is the gaining problem in Eq. (26) and the decision function in Eq. (11) is in the form of a dot product. A nonlinear function $\Phi$ is introduced to map the data to a high dimensional inner product space $\mathcal{H}$ by [13]:

$$\Phi: \mathbb{R}^d \rightarrow \mathcal{H}.$$

The mapping $\Phi$ is implemented by a kernel function $K$ that satisfies Mercer's conditions [66], such that $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$. The kernel trick is that we never need to explicitly represent the nonlinear mapping $\Phi$ and then just replace $\langle x_i, x_j \rangle$ by $K(x_i, x_j)$ in the training algorithm.

In the design of an SVM training algorithm, we expect to find a hyperplane with a large margin to separate the data. Intuitively, the hyperplane with a large margin has a good generalization performance. It is necessary to know why the margin plays a crucial role in SVM from a technical viewpoint. Let us start to explain it by means of Vapnik's statistical learning theory [103].

The Structural Risk Minimization (SRM) principle was derived from a result of statistical learning theory, consisting in the definition of an upper bound for the expected risk of a given classifier. For a *k*-class problem, decision functions $f(\boldsymbol{x}, \alpha)$ take on exactly *c* values, corresponding to the *k* class labels.

Let data $(x_1, y_1)$, …,$(x_l, y_l) \in X \times Y$, be generated and i.i.d. (independently drawn and identically distributed) from a cumulative probability distribution $P(x, y)$, where $X \subseteq \mathbb{R}^d$ and $Y$={1, -1}. The learning function is to find one function from a set of functions $f(x, \alpha): X \rightarrow \{1, -1\}$ such that the expected misclassification error on the test set, also drawn from $P(x, y)$, is minimal:

$$R(\alpha) = \int \frac{1}{2} |f(x, \alpha) - y| dP(x, y) \tag{27}$$

We use the 0/1 (indicator) loss function [102]:

$$L(x,y,\alpha) = \begin{cases} 0, & if\ f(x,\alpha) = y, \\ 1, & if\ f(x,\alpha) \neq y, \end{cases} \tag{28}$$

Then

$$R(\alpha) = \int \frac{1}{2} L(x,y,\alpha)\, dP(x,y) \tag{29}$$

Eq. (27) is called the expected risk (or actual risk). But since $P(x,y)$ is usually unknown, the corresponding empirical risk, $R_{emp}(\alpha)$, is an approximation of $R(\alpha)$, constructed on the basis of the given training samples $(x_1, y_1), \dots, (x_l, y_l)$, defined by:

$$R_{emp}(\alpha) = \frac{1}{l} \sum_{i=1}^{l} L(x_i, y_i, \alpha). \tag{30}$$

The $R_{emp}(\alpha)$ is called "empirical risk". The empirical risk can be connected with the expected risk by a probability bound [102]. That is, for any $f(x,\alpha)$ and $l > h$, with a probability of at least $1-\eta$,

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{h\left(log\frac{2l}{h}+1\right) - \log\left(\frac{\eta}{4}\right)}{l}} \tag{31}$$

holds, where $h$ is a non-negative integer called the Vapnik Chervonenkis (VC) dimension, and $l$ is a measure of the capacity of the function class $f(x,\alpha)$. The second term in Eq. (31) is called the "confidence (capacity) term", which is an increasing function of $h$ for the fixed $\eta$.

In summary, an SVM constructs a hyperplane or a set of hyperplanes, which have the largest distance and lowest generation error to the nearest training data points of any class, in a high or infinite dimensional space. SVM can be used for classification, regression or other tasks. The linear SVM can be extended with a soft margin to tolerate a certain error rate in the training procedure. In addition, by applying the kernel trick to maximum-margin hyperplanes, the SVMs become nonlinear classifiers.

### 3.1.2 S3VM with Rejection Option (RO-S3VM)

In this study, we propose a Semi-Supervised SVM with Rejection Option (RO-S3VM), that minimizes both the misclassification error and the function capacity based on all the available data and information.

Since we do not change the generative model in SVM with a Rejection Option (RO-SVM), $L_P$ and $L_D$ in RO-S3VM are identical to as the ones in RO-SVM. Thus, let us start to introduce RO-S3VM from the theory of RO-SVM. The reject option is very useful to safeguard against excessive misclassifications in pattern recognition applications that require high classification reliability. In the framework of the minimum risk theory, Chow defined the optimal classification rule with a reject option [21]. In the simplest case where the classification costs do not depend on the classes, Chow's rule consists of rejecting a pattern if its maximum a posteriori probability is lower than a given threshold [22]. The optimality of this rule relies on the exact knowledge of the a posteriori probabilities. However, in practical applications, the a posteriori probabilities are usually unknown [37]. In Chapter 5, details about rejection measurement are discussed, and a new rejection measurement, so-called Linear Discriminant Analysis Measurement (LDAM), is defined and applied in this case.

Moreover, as pointed out by Fumera et al. in [37], the rejection region must be determined during the training phase in order to obey the Structural Risk

Minimization (SRM) principle, in which SVMs are based. Thus, we will discuss classification with a rejection class in the framework of the SRM principle as an extension of the SVM classifier in this section.

Consider now the problem of classification with a rejection option. For a $k$-class problem, decision functions $f(\boldsymbol{x}, \alpha)$ now take on $k+1$ values such that $c$ of them correspond to the $c$ class labels, while the ($k$+1)st one corresponds to the rejected class. Moreover, loss functions take on at least three values: correct classification, misclassification, and rejection.

The SVM classification technique was originally derived by applying the SRM principle to a two-class problem as mentioned earlier. The technique uses a classifier that implements linear decision functions in Eq. (11) and the 0/1 (indicator) loss function in Eq. (28). The simplest generalization of linear decision functions in Eq. (11) to classification with a rejection option is that functions are defined by means of pairs of parallel hyperplanes, so that the rejection region is the space delimited by such hyperplanes. Formally, let us denote a pair of parallel hyperplanes as:

$$\boldsymbol{w} \cdot \boldsymbol{x} + b \pm \varepsilon = 0, \varepsilon \geq 0\,. \tag{32}$$

The corresponding decision function is then defined as follows:

$$f^1(x, \alpha) = +1, \quad if\; \boldsymbol{w} \cdot \boldsymbol{x} + b \geq \varepsilon, \tag{33}$$

$$f^1(x, \alpha) = -1, \quad if\; \boldsymbol{w} \cdot \boldsymbol{x} + b \leq -\varepsilon,$$

$$f^1(x, \alpha) = 0, \quad if - \varepsilon < \boldsymbol{w} \cdot \boldsymbol{x} + b < \varepsilon,$$

where $\alpha$ denotes the parameters $\boldsymbol{w}, b,\ \varepsilon$ , while the class labels are denoted by $y = +1$ and $y = -1$, and the rejection decision by $y = 0$ . The distance between the hyperplanes, that is, the width of the rejection region, is equal to $2\varepsilon / \| \boldsymbol{w} \|$ . Analogously, the simplest extension of the indicator loss function [Eq. (28)] to classification with a rejection option is the following loss function:

$$L^1(x,y,\alpha) = \begin{cases} 0, & if\ f^1(x,\alpha) = y, \\ w_R, & if\ f^1(x,\alpha) = 0, \\ 1, & if\ f^1(x,\alpha) \neq y, and\, f^1(x,\alpha) \neq 0, \end{cases} \tag{34}$$

where $w_R$ denotes the cost of a rejection. Obviously $0 \leq w_R \leq 1$. The corresponding expected risk is:

$$R^1(\alpha) = w_R\, P(rejection) + P(error), \tag{35}$$

where $P(error)$ and $P(rejection)$ denote respectively the misclassification and rejection probabilities achieved when using the function $f^1(\boldsymbol{x},\alpha)$. Accordingly, the expression of the empirical risk [Eq. (30)], for a given decision function and a given training set is:

$$R^1_{emp}(\alpha) = w_R R + M, \tag{36}$$

where $R$ and $M$ represent the rejection and misclassification rates achieved by $f^1(\boldsymbol{x},\alpha)$ on training samples, respectively. According to the SRM principle, training this classifier consists of finding the pair of parallel hyperplanes [Eq. (32)], which provide the best trade-off between the VC dimension and the empirical risk. We call such a pair the Generalized Optimal Plane with Rejection Option (RO-GOP).

By analogy, we assume that the RO-GOP can be defined as a pair of parallel hyperplanes [Eq. (32)] which minimize the empirical risk [Eq. (36)], and we separate the samples that have been correctly classified and *accepted* with a maximum margin. It is important to remember that a pattern $\mathbf{x}_i$ is accepted if $|\boldsymbol{w} \cdot \boldsymbol{x}_i + b| \geq \varepsilon$ . For a pair of parallel hyperplanes [Eq. (32)], we define the margin of an accepted pattern as its distance from the hyperplane $\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$.

In multi-class SVM, the rejection class is denoted by $y = -1$. For $k$ classes of data $(k > 2)$, $k$ SVM classifiers are formed and denoted by SVM$_i$, $i$=*1,2, ...k*. For the test sample $x$, $d_i(x) = w_i \cdot x + b$ can be obtained by using SVM$_i$, where $d_i$ is the decision function for class $i$, $w_i$ is a normal vector perpendicular to the hyperplane that separates class $i$ from all the other classes, and the parameter $b_i$ is the distance from the origin to the hyperplane along the normal vector $w_i$. The test sample $x$ is considered to belong to the $jth$ class, where $d_j(x) = \max_{i=1,2,\ldots,k} d_i(x)$. Thus, the rejection decision in this $k$-class problem should be determined on $\varepsilon_1, \ldots, \varepsilon_k$, which are thresholds to each of the corresponding margins. One optimal way is to find a global rejection measurement to define the rejection class in the training procedure. The measurement is based on all the confidence values of the classifier outputs and the relations between them so that we do not need to determine the $\varepsilon_1, \ldots, \varepsilon_k$ one by one. Hence, once one rejection class can be determined in the training procedure, the classifier can be re-trained with *(k+1)* classes. Details can be found in Chapter 5.

In this RO-S3VM, the margins for the generative models are the same as the ones in RO-SVM. Only the extra information detected from the testing procedure may change the classification results on certain patterns. This extra information may rely on the evaluation results from the rejection measurements. Thus, we did not need to introduce the RO-S3VM in theory in this Section.

## 3.2 Framework for Training and Testing RO-S3VM

In this section, we will describe all the procedures in both training and testing of the RO-S3VM applied in this thesis. Firstly, all the operations in the training process are introduced. Besides training in the standard supervised learning method, two verifiers should be trained as well. In the testing procedure, we classify samples with supervised learning, reject ones with low rejection measurements, and verify them with the extra information retrieved from the unlabeled data. Three flowcharts for the training, verification, and testing procedures will be described in this section.

Initially, we trained an SVM classifier on the training set. In the recognition process, the standard procedures of image pre-processing, feature extraction, and classification were implemented. In image pre-processing, we performed noise removal, grayscale normalization, and sizes were normalized to 32 by 32 pixels. Gradient features were extracted from the gray-scale images, and the Support Vector Machine (SVM) was chosen as a classifier with a Radial Basis Function (RBF) kernel. Then, we applied a rejection measurement (LDAM) to reject the unreliable samples and to find the rejection classes (Figure 8).

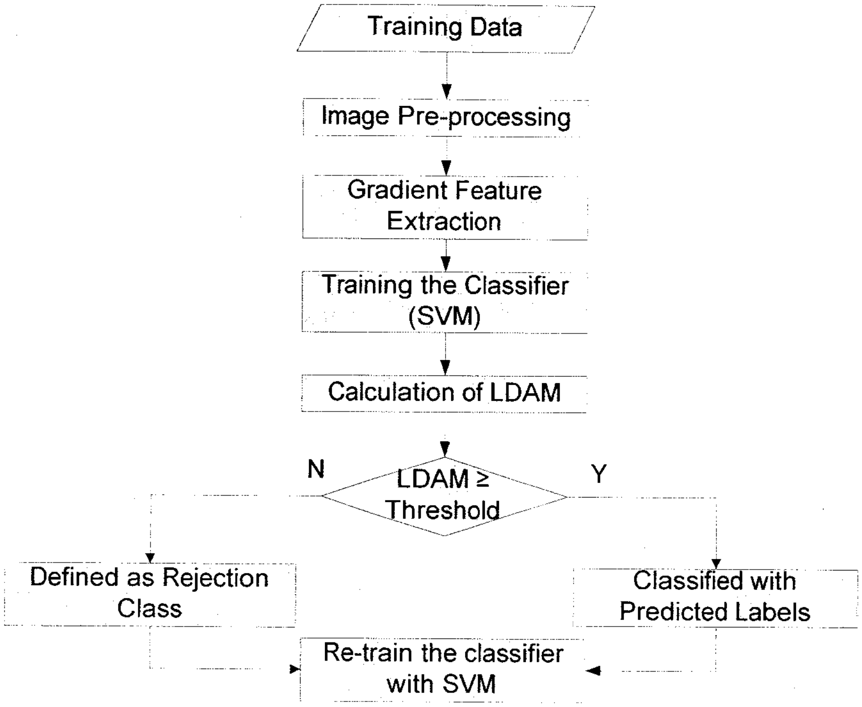

Training Data
Image Pre-processing
Gradient Feature Extraction
Training the Classifier (SVM)
Calculation of LDAM
N
LDAM ≥ Threshold
Y
Defined as Rejection Class
Classified with Predicted Labels
Re-train the classifier with SVM

**Figure 8. Flowchart for the training procedure**

Due to the error categorization, two verifiers had to be built in the training procedure as well. One was a verifier between Classes "0" and "1", and we called it Verifier I. For the two confusing classes of 0's and 1's, we re-trained a pair-wise classifier with only two dimensional features (height and width) among all the samples in classes 0 and 1 in the Training Set. Since size normalization causes Classes "0" and "1" in Hindu-Arabic to become similar, global features, such as height and width in the original images, needed to be considered and re-trained. Thus, we trained Verifier I with a new feature set of samples between Classes "0" and "1".

On the other hand, we built another Verifier II between Classes "2" and "3". Since even human beings may have problems to distinguish some samples in Classes "2" and "3" in Hindu-Arabic, due to their confusing shapes, extra information rather than shapes need to be detected and applied to the verifier. We designed a procedure to detect the writer's writing styles with the semi-supervised learning method. Accordingly, we applied the clustering process to group all the samples in each class into two clusters. We assigned the sub-class number to each pattern in the two confusing classes and re-trained a sub-class classifier with all the samples in the two confusing classes. Each writing style was determined as described in Chapter 7. The flowcharts for Verifiers I & II are shown in Figure 9.

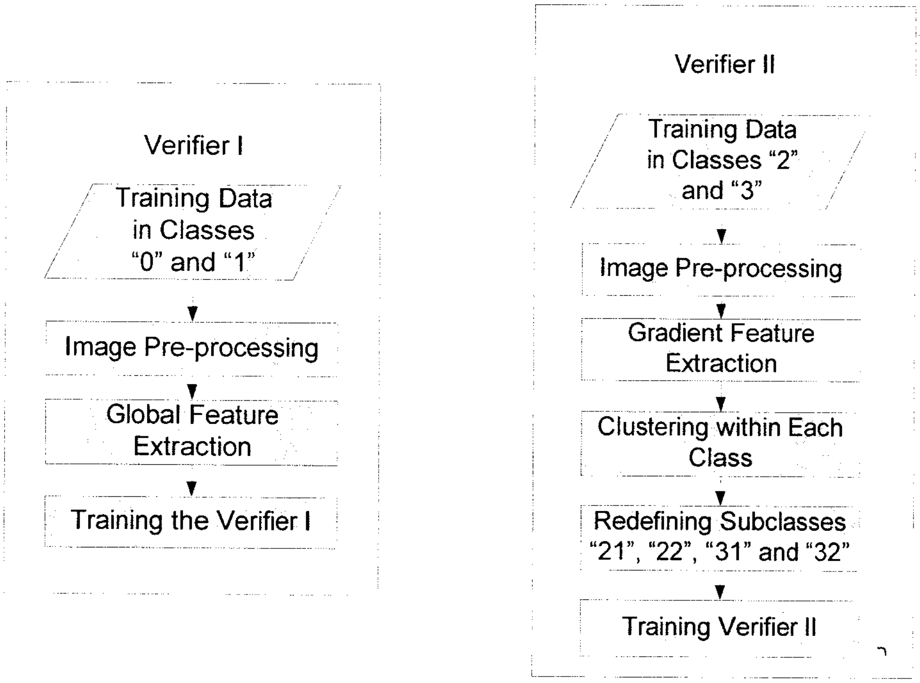

Verifier I
Training Data in Classes "0" and "1"
Image Pre-processing
Global Feature Extraction
Training the Verifier I
Verifier II
Training Data in Classes "2" and "3"
Image Pre-processing
Gradient Feature Extraction
Clustering within Each Class
Redefining Subclasses "21", "22", "31" and "32"
Training Verifier II

**Figure 9. Flowchart for Verifier I & Verifier II**

In the testing procedure, recognition and rejection were applied (Figure 10). Only samples rejected by LDAM need to be verified with two sub-classifiers. The samples classified to one of the two confusing classes (2's and 3's) and rejected by the previous step should go through verification by the sub-class classifier, which returned one of the sub-classes. Combined with the writer's writing style, the final recognition results could be improved. If this sub-class was Subclass (SC) 22 or

Subclass 31, the sample had an ambiguous shape and could have been a sample of either 2 or 3. See Chapter 7 for details. In this case, the writer's Combined Writing Style (CWS) could be applied to arrive at a classification. Whereas if the sub-class was SC21 (SC32), the sample would be assigned to class 2 (3), respectively. The samples classified to one of the two confusing classes (0's and 1's) and rejected by the previous step went through verification by the sub-class classifier. Moreover, errors with high confidence values in LDAM have to be verified with the original documents to correct the mislabeling.

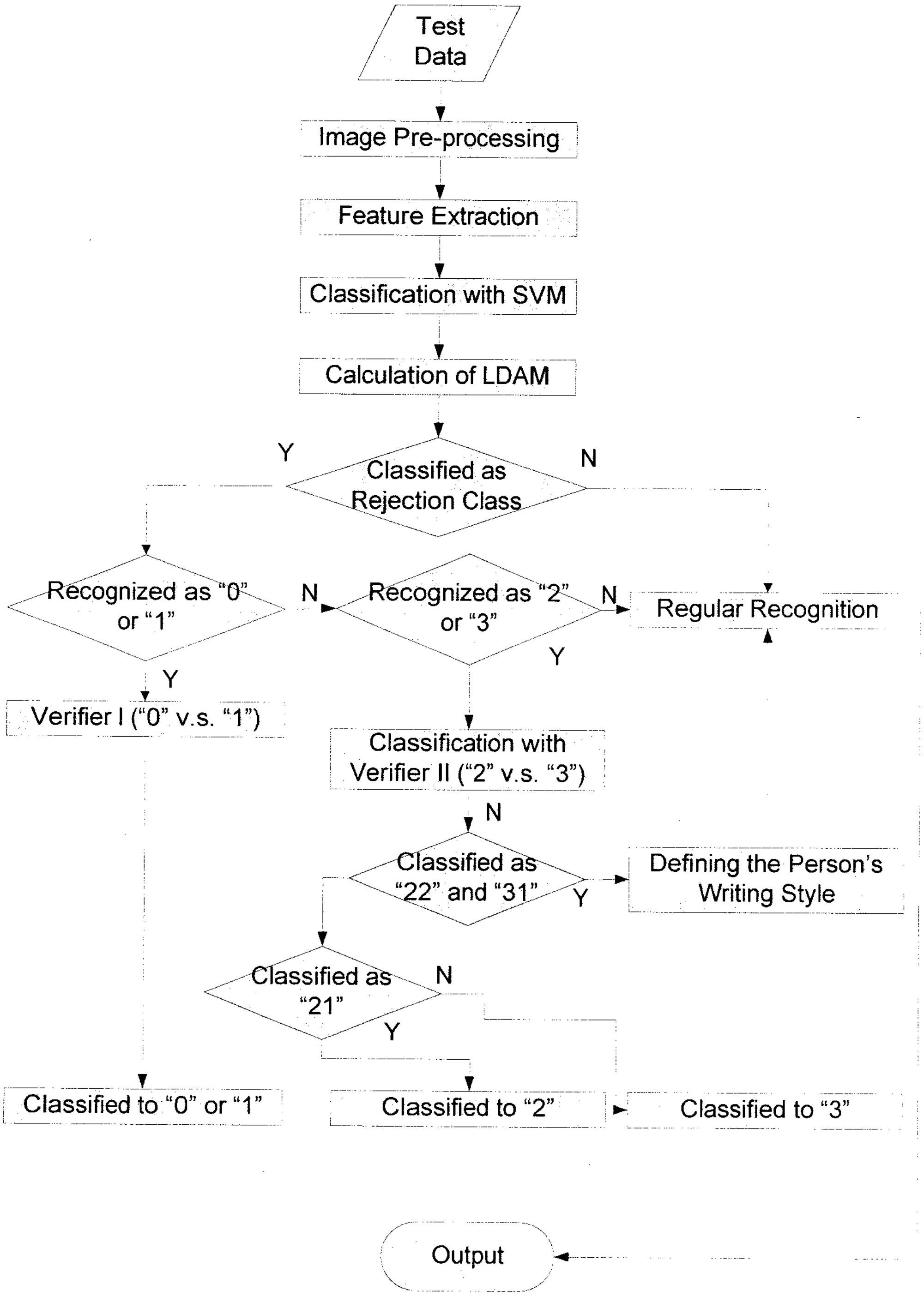


**Figure 10. Flowchart for the testing procedure**

# Chapter 4

# Supervised Learning

In this chapter, we propose a standard recognition system with supervised learning. We apply some state-of-the-art technologies for the recognition. A recognition algorithm consists of three main tasks that are discussed in this chapter: pre-processing, feature extraction, and classification. In Section 4.1, we discuss the performance of noise removal, grayscale normalization, and size normalization in image pre-processing. Gradient features are introduced in Section 4.2, which are extracted from the gray-scale images, and Support Vector Machines (SVMs) are applied as a classifier with a Radial Basis Function (RBF) kernel, briefly described in Section 4.3. Gradient features and downsampling are image processing techniques commonly used in the recognition of handwritten characters from various languages, including Arabic numerals [90], Devangari characters [70], etc. In each pattern, a feature vector with a size of 400 (5 horizontal, 5 vertical, 16 directions) is produced.

Satisfying recognition results in this thesis have been achieved where the results are compared with Alamri [4]. Details can be found in Section 4.4. Therefore, this SVM classifier is designed as a supervised learning classifier. Finally, a summary of this chapter is presented in Section 4.5.

This proposed (novel) system has been successfully used with different applications, such as Numeral Recognitions in Urdu [79], Farsi [41], Pashto [87], and Dari [86], Word Recognition in Urdu [80] and Word Spotting in Urdu [81], and touching pair numeral recognition in Arabic and date recognition in Arabic [5].

## 4.1 Image Pre-processing

In image pre-processing, researchers normally perform noise filtering, binarization, thinning [109], skew correction [52], slant normalization [15], size normalization, etc., to enhance the quality of images and to correct distortion. All of these factors influence the performance of a character recognition system.

Since image normalization can be used as a preprocessing stage to assist computer or human object perception, various normalization methods have been adopted [60] with different functions, such as dimension-based normalization and moment-based normalization. Normally, the image is linearly mapped onto a standard plane by interpolation/extrapolation. The size and position of a character is controlled such that the x/y dimensions of a normalized plane are filled. The implementation of interpolation/extrapolation can influence to the recognition performance [68, 91].

Historically, when a database was constructed or before features were extracted, most researchers normalized the spatial resolution. The spatial resolution of an image is related to its height (rows) and width (columns) per dimension. For example, in the MNIST database [58], each image was normalized to 20 * 20 and filled in a 28 * 28 pad; Liu et al. in [60] normalized images to 35 * 35 when they investigated normalization and feature extraction techniques; and Perez et al. in [72] normalized images to 15 * 23 before creating prototypes. In [27], digits were normalized to 16 * 16 before feature extraction. It seems obvious that, with too small images, the recognition rate of a handwritten digit database will be reduced. But how many pixels as a height and a width for an image can be considered as too small? What is the effect of size normalization on handwritten numeral recognition? From many observations and experiments, we concluded that [42] when normalizing images to the size of 32 by 32, the performance of a handwritten digit recognition system is optimal because, on the one hand, the recognition rate is high; and on the other hand, space on the hard disk is not wasted.

In image pre-processing, besides size normalization we also performed noise removal and grayscale normalization. There are seven steps in image pre-processing. Firstly, we load the original grayscale images. By thresholding the original grayscale image, we obtain a background-eliminated grayscale image to remove some noises. Then, we bound the image with a rectangle to remove the blank boundaries. Afterwards, we normalize the image's grayscale to eliminate the dependence of feature values on gray levels. We rescale images' grayscale to a standard mean of 210 and standard deviation of 10. For size normalization, we use Moment Normalization (MN) to convert images to a size of 32 * 32, which aligns the centroid (center of gravity) to the geometric center of a normalized plane, and re-bounds the image based on second-order moments [61]. Finally, we binarize the images based on the threshold calculated with the Otsu Method [69]. The procedure with an example in image pre-processing is shown in Figure 11.

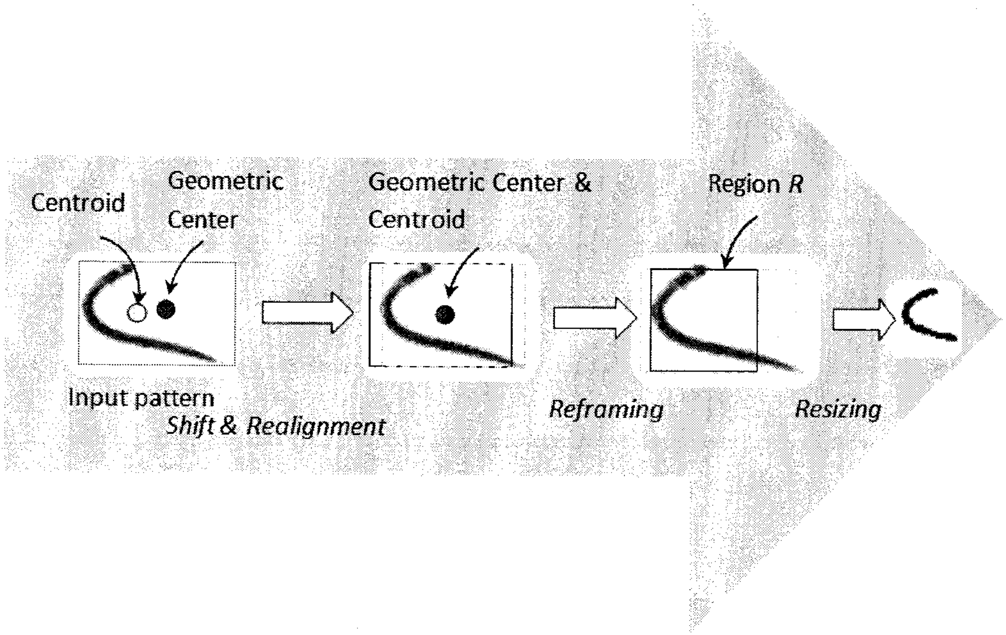


**Figure 11. Demonstration of moment-based normalization procedure**

## 4.2 Feature Extraction

Since many classifiers cannot efficiently process the raw images or data, feature extraction is necessarily applied, which aims to reduce the dimensions of the data while extracting useful information [57]. The performance of a classifier relies very much on the quality of the features. A good set of features should represent common characteristics that are particular to one class but also represent the obvious difference in characteristics between two classes. As the features are extracted from the original data, these features should maintain the distinguishable information as much as possible.

In this thesis, supervised learning will be performed on gradient features [89], which are extracted from the binary images. Gradient features maintain both the position and direction information in the images. These features were applied and achieved a high recognition rate by Dong et al. [27].

Gradient features are extracted from gray-scale images, so we should first convert binary images to gray-scale images. The gray-scale normalized image is standardized such that its mean and maximum values are 0 and 1.0, respectively. After centering a normalized image (e.g. 28 * 28) into a 32 * 32 box, as mentioned in Section 4.1, the Robert filter [24] is applied to calculate its gradient strengths and directions. In pattern recognition, edge detection is traditionally implemented by convolving the signal with some form of linear filter, and usually it is a filter that approximates a first or second derivative operator. The simplest gradient operator is the Robert's Cross operator and it uses the masks $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ and $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$. Thus, the Robert's Cross operator uses the diagonal directions to calculate the gradient vector. For example, the gradient magnitude and direction of pixel $g(m,n)$ are calculated as follows:

$$\frac{\partial I}{\partial x} = \Delta u = g(m,n) - g(m+1,n+1), \tag{43}$$

$$\frac{\partial I}{\partial y} = \Delta v = g(m,n+1) - g(m+1,n), \tag{44}$$

Direction: $$\theta(m,n) = \arctan(\frac{\Delta v}{\Delta u}), \tag{45}$$

Strength: $$s\,(m,n) = \sqrt{\Delta u^2 + \Delta v^2}, \tag{46}$$

where $\theta\,(m,n)$ and $s\,(m,n)$ specify the direction and gradient magnitude of pixel $g(m,n)$, respectively.

We calculate the strength of the gradient as a feature vector. The direction of the gradient is quantized to 32 levels with an interval of $\pi/16$. The normalized character image is divided into 81 (9 horizontal * 9 vertical) blocks. The strength of the gradient in each of the 32 directions is accumulated in each block to produce 81 local joint spectra of directions and curvatures. In Figure 12, we show an example with a normalized grayscale image in (a), its gradient strength in (b), and gradient direction in (c).

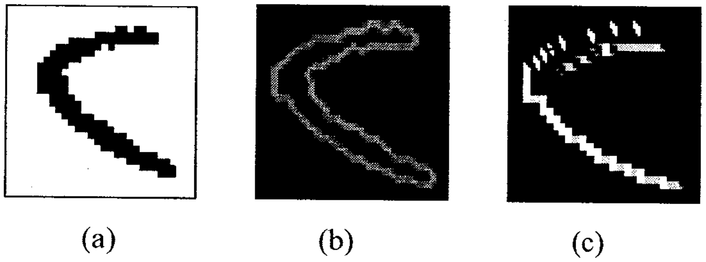


**Figure 12. Gradient features on a sample image:**
(a) Greyscale image of size 32x32, (b) Gradient Strength, and (c) Gradient Direction

After extracting the strength and directions in each image, the spatial resolution is reduced from 9*9 to 5*5 by down sampling every two horizontal and every two vertical blocks with a 5*5 Gaussian filter. Similarly, the directional resolution is reduced from 32 to 16 levels by down sampling with a weight vector of $[1\,4\,6\,4\,1]^T$, to produce a feature vector of size 400 (5 horizontal, 5 vertical, and 16 directions). Moreover, variable transformation ($y = x^{0.4}$) is applied to make the distribution of the feature Gaussian-like. The feature size is reduced to 400 by principal component analysis (KL transform). Finally, we scale the feature vectors by a constant factor such that the values of feature components range from 0 to 1.0.

## 4.3 Classification

Support Vector Machines [101] were chosen as a classifier. Details of the principles of SVMs can be found in Section 3.1 of Chapter 3. In this thesis, Radial Basis Function (RBF) was chosen with a kernel $k\ (x_i, x_j) = \exp\ (-\gamma\ \|x_i - x_j\|^2)$ in the SVM of this supervised learning method. Two parameters (c, $\gamma$) need to be determined when using RBF kernels, with $c > 0$ being the penalty parameter of the error term and $\gamma$ the kernel parameter. These parameters were optimally chosen by cross-validation via a parallel grid search on the training set [106]. These optimal parameter values were then applied on the test set.

## 4.4 Databases and Experimental Results

Our recognition system was applied to the CENPARMI Hindu-Arabic Isolated Numerals database [4]. This database contains 18,585, 6,199, and 6,199 samples in the Training, Validation, and Test sets, respectively, with the distribution shown in Table 1. Since validation was not implemented in this experiment, the Training and Validation sets were combined to form the Training set.

**Table 1. Distribution of samples in CENPARMI Hindu-Arabic Numerals Database**

| Label | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Training | 2,647 | 2,456 | 2,542 | 2,503 | 2,447 | 2,253 | 2,477 | 2,338 | 2,321 | 2,800 | 24,784 |
| Test | 662 | 612 | 637 | 627 | 613 | 564 | 618 | 585 | 581 | 700 | 6,199 |

The recognition rate on the test set was 98.47%, which is significantly higher than the performance (93.60%) of [4] on the same database (Table 2). The confusion matrix is also shown below (Table 3).

**Table 2. Performances on the test set with LDAM compared with [4]**

| Classifier | LibSVM | [4] |
|---|---|---|
| Recognition Rate (%) | 98.47 | 93.60 |
| Error Rate (%) | 1.53 | 6.40 |

**Table 3. Confusion matrix on the Test set**

| | | **Output** | | | | | | | | | | | | | Pct. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Cor. | Inc. | Total | (%) |
| **Truth Label** | 0 | 657 | 3 | | | | 1 | | 1 | | | 657 | 5 | 662 | 99.2 |
| | 1 | | 611 | | | | | 1 | | | | 611 | 1 | 612 | 99.8 |
| | 2 | | | 602 | **34** | 1 | | | | | | 602 | 35 | 637 | 94.5 |
| | 3 | | | **39** | 587 | | | | | | 1 | 587 | 40 | 627 | 93.6 |
| | 4 | | | 3 | | 609 | | | | | | 609 | 3 | 612 | 99.5 |
| | 5 | 4 | | | | | 560 | | | | | 560 | 4 | 564 | 99.3 |
| | 6 | | | | | 1 | | 618 | | | | 618 | 1 | 619 | 99.8 |
| | 7 | | | | | | | | 585 | | | 585 | 0 | 585 | 100.0 |
| | 8 | | | 1 | | | | | | 580 | | 580 | 1 | 581 | 99.8 |
| | 9 | | | | | | | 4 | | 1 | 695 | 695 | 5 | 700 | 99.3 |
| Cor. | | 657 | 611 | 602 | 587 | 609 | 560 | 618 | 585 | 580 | 695 | **6104** | **95** | **6199** | **98.5** |
| Inc. | | 4 | 3 | 43 | 34 | 2 | 1 | 5 | 1 | 1 | 1 | | | | |
| Total | | 661 | 614 | 645 | 621 | 611 | 561 | 623 | 586 | 581 | 696 | | | | |
| Pct. (%) | | 99.4 | 99.5 | 93.3 | 94.5 | 99.7 | 99.8 | 99.2 | 99.8 | 99.8 | 99.9 | | | | |

Each parameter in SVM was chosen and calculated by cross-validation. The result of cross-validation via a parallel grid is shown in Figure 13. When lg(c) = 1 and lg(γ) = -7, the performance on the training set achieved the highest recognition rate of 98.05%. Thus, we set c = 2 and γ = 0.0078125, and then tested it on the testing set. As a result, the recognition rate was 98.47% for the testing set.

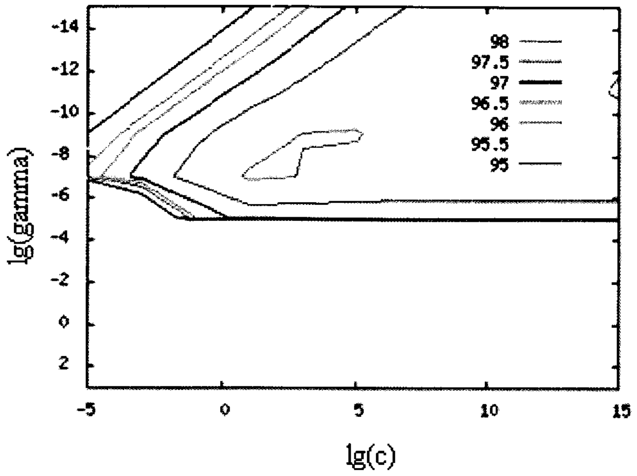


**Figure 13. Cross-validation via a parallel grid**

Out of the 6199 samples in the test set, the number of misclassifications was 95 (1.53%), and most of these errors happened between classes 2 and 3, and they cannot be correctly identified, even by human beings. All of the recognition errors (73 samples) that arose between Classes 2 and 3 in the supervised learning method are shown in Figures 14 and 15. Aside from the confusion between classes 2 and 3, there were 22 mistakes that happened among other classes. They are shown in Figure 16.

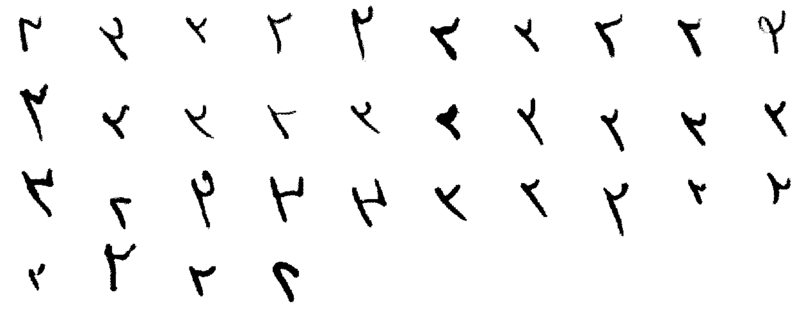


**Figure 14. Recognition errors: samples in Class 2 were recognized as 3 in the supervised learning**

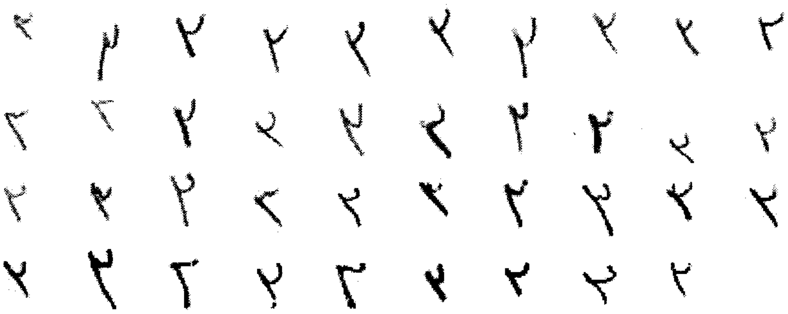


**Figure 15. Recognition errors: samples in Class 3 were recognized as 2 in the supervised learning**

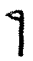
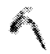
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Images | | | |  |  | | |
| Truth Label → Output | 0→1 | 0→1 | 0→1 | 0→5 | 0→7 | 1→6 | 2→4 |
| Images | | | | | | | |
| Truth Label → Output | 3→9 | 4→2 | 4→2 | 4→2 | 5→0 | 5→0 | 5→0 |
| Images | | | | | | |  |
| Truth Label → Output | 5→0 | 6→4 | 8→2 | 9→6 | 9→6 | 9→6 | 9→6 |
| Images |  | | | | | | |
| Truth Label → Output | 9→8 | | | | | | |

**Figure 16. Errors in supervised learning method**

## 4.5 Conclusion

In summary, we designed a recognition system with some state-of-the-art technologies. We performed noise removal, grayscale normalization, and size normalization in image pre-processing. Gradient features were extracted from the processed images, and we applied SVM to do the classification. As a result, the recognition rate on the test set was 98.47%, which is significantly higher than the previous research on the same database.

# Chapter 5

# Rejection Measurement

In this chapter, we define a novel rejection measurement that is called the Linear Discriminant Analysis Measurement (LDAM). This rejection measurement will be implemented to reject the data with unreliable classification results produced by the supervised learning method. To implement the rejection, which can be considered as a two-class problem of accepting the classification result or otherwise, an LDA-based measurement is used to determine a new rejection threshold. This measurement (LDAM) is designed to take into consideration the confidence values of the classifier outputs and the relations between them, and it represents a more comprehensive measurement than traditional rejection measurements such as First Rank Measurement (FRM) and First Two Ranks Measurement (FTRM).

Since the problem in current rejection measurement has motivated us to develop a new rejection measurement, we firstly point out the problem in Section 5.1. In Section 5.2, FRM and FTRM are defined and described so that we can define and compare the LDAM in Section 5.3. Moreover, the experiments conducted on rejection measurement on different databases and with different classifiers are described in

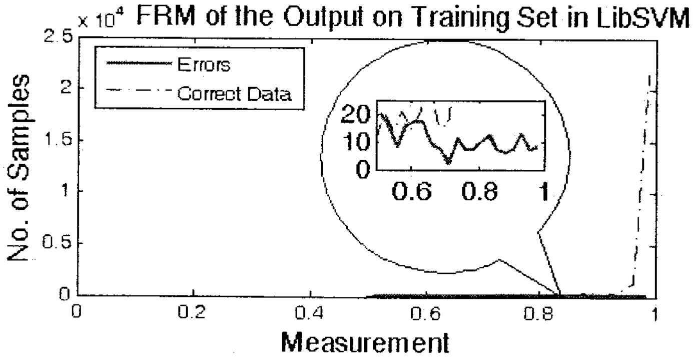
Section 5.4, and the conclusion of this chapter is given in Section 5.5.

## 5.1 Problem in Rejection Measurement

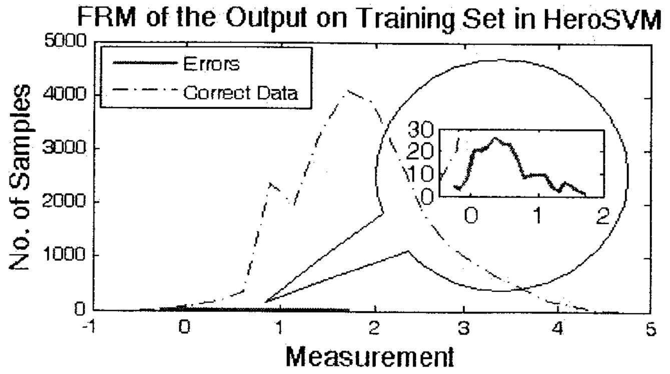
In document recognition, it is important to obtain a high accuracy or reliability and to reject patterns that cannot be classified with high confidences. This is the case for applications such as a system that processes financial documents without rejection, in which errors can be very costly and therefore can yield far less tolerable results compared to systems that have a reject option. When the cost of misclassifications is very high, it is useful to allow a pattern classification system to withhold the automatic classification of an input pattern, if it is considered unreliable. This is known as the reject option. In this research, we applied a rejection criterion on the results from the supervised learning method, which allowed us to design some verifiers for the final recognition.

In considering the outputs of classifiers for the rejection option as a two-class problem (accepted or rejected classification), the outputs at the measurement level can be considered as features for the rejection option. An output vector's components may represent distances or probabilities, and we expect the confidence value (measure) of the first rank (most likely class) to be far distant from the confidence values or measures of the other classes. In other words, good outputs should be easily separated into two classes: the confidence value of the first rank and the others. In the following discussion, we assume that the classifier outputs the probabilities of the patterns for each class, and the considerations would be analogous in the case when the classifier outputs the distances.

It seems that the Bayes' decision rule embodies a rejection rule; namely, the decision can be based on the maximum confidence value (called First Rank Measurement (FRM) in this thesis), provided that this maximum exceeds a certain threshold value. However, this approach did not perform satisfactorily when experiments were performed on the CENPARMI Hindu-Arabic Isolated Numerals Database with any SVM software package, such as LibSVM [18, 24] and HeroSVM [28]. LibSVM maps the outputs from a Support Vector Machine (SVM) [24] to posterior probabilities for all classes; and HeroSVM provides distances from an input pattern to the Optimal Separating Hyperplane (OSH) of each class. The results on the training set are shown in Figure 17. In LibSVM, the distribution [Figure 17(a)] of incorrectly classified samples is not Gaussian in shape, but remains flat throughout a range of confidence values. This is the case while only the correctly classified samples follow a Gaussian distribution. In HeroSVM, although the distributions [Figure 17(b)] of correctly and incorrectly classified samples are Gaussian in shape, their measurements do overlap for almost half of the range. Therefore, it is difficult to design a rejection strategy based on the measurement of maximum confidence value.



(a)



(b)

**Figure 17. Distribution of the output on the Training set in: (a) LibSVM and in (b) HeroSVM**

## 5.2 First Rank Measurement (FRM) & First Two Ranks Measurement (FTRM)

Generally, rejection strategies can be directly applied to the classifier's outputs with probability estimations. In an M-class problem, suppose $P(x) = \{p_1(x), p_2(x), \dots, p_M(x)\}$ is the classification output vector of the given pattern $x$, with probabilities $p_i(x)$ in descending order. The decision can be based on $sgn\,(\Phi_1(x) - T_1)$, where $T_1$ is a threshold derived from the training data, and $\Phi_1(x) = p_1(x)$.

If $\Phi_1(x) \leq T_1$, the classifier rejects the pattern and does not assign it to a class (it might instead be passed to a human operator). This has the consequence that on the remaining patterns, a lower error rate can be achieved. This method uses the First Rank Measurement (FRM) [28].

Using this method, the frequency distribution according to confidence values of samples in the training set is considered and the threshold $T_1$ is determined accordingly.

However, FRM cannot distinguish between reliable and unreliable patterns with the probability distribution of erroneous samples shown in Figure 17.

To overcome this deficiency of FRM, we have designed First Two Ranks Measurement (FTRM) [93], which uses the difference between the probabilities $p_1(x)$ and $p_2(x)$ of the first two ranks as a condition of rejection. In FTRM, the measurement function is $\Phi_2(x) = \|p_1(x) - p_2(x)\|$, where $\|.\|$ can be any distance measurement, and the decision function is based on $sgn\ (\Phi_2(x) - T_2)$, where $T_2$ is a threshold derived from the training set.

However, FTRM cannot solve the problem in some cases. For example, if $\|p_1(x) - p_2(x)\|$ is relatively large compared to $T_2$, but the distance $\|p_2(x) - p_3(x)\|$ is much larger, this pattern may still be accepted, when this pattern should really have been rejected since the top two classes are closer together in terms of relative distance.

## 5.3 Linear Discriminant Analysis Measurement (LDAM)

To consider the relative difference between the measurements in the first two ranks and all the other measurements, LDAM is defined and then applied. Since rejection in classification can be considered as a two-class problem (acceptance or rejection), we apply LDA to implement rejection.

An LDA approach to the problem assumes that the conditional probability density functions of the two classes are both normally distributed. There are

$n = n_1 + n_2$ observations with d features in the training set, where $\{x_{1i}\}_{i=1}^{n_1}$ arise from class $\omega_1$ and $\{x_{2i}\}_{i=1}^{n_2}$ arise from class $\omega_2$ . Gaussian-based discrimination assumes two normal distributions: $(x \mid \omega_1) \sim N(\mu_1, \Sigma_1)$ and $(x \mid \omega_2) \sim N(\mu_2, \Sigma_2)$ . In LDA, the projection axis (discriminant vector) w for discriminating between two classes is estimated to maximize the Fisher criterion:

$$J(w) = tr((w^T S_w w)^{-1}(w^T S_B w)) \qquad (47)$$

where tr(·) denotes the trace of a matrix, $S_B$ and $S_w$ denote the between-class scatter matrix and within-class scatter matrix respectively, and $w$ is the optimal discriminant vector. For the two classes $\omega_1$ and $\omega_2$, with a priori probabilities $p_1$ and $p_2$ (it is often assumed that $p_1 = p_2 = 0.5$ ), the within-class and between-class scatter matrices can be written as:

$$S_w = p_1\Sigma_1 + p_2\Sigma_2 = \Sigma_{12}, \qquad (48)$$

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T, \qquad (49)$$

where $\Sigma_{12}$ is the average variance of the two classes. The maximum separation occurs when:

$$\begin{aligned} w &= S_w^{-1}(\mu_1 - \mu_2) = (p_1\Sigma_1 + p_2\Sigma_2)^{-1}(\mu_1 - \mu_2) \\ &= \Sigma_{12}^{-1}(\mu_1 - \mu_2). \end{aligned} \qquad (50)$$

To apply this principle to the outputs for the rejection option as a one-dimensional application, we define the two sets $G^{(1)}(x) = \{p_1(x)\}$ , and $G^{(2)}(x\ ) = \{p_2(x),\ p_3(x), \ldots,\ p_M(x)\}$. Then,

$$\mu_1 = p_1(x), \qquad (51)$$

$$\mu_2 = \frac{1}{M-1}\sum_{i=2}^{M} p_i(x), \tag{52}$$

$$\Sigma_1 = (p_1(x) - \mu_1)^2 = 0, \tag{53}$$

$$\Sigma_2 = \frac{1}{M-1}\sum_{i=2}^{M} (p_i(x) - \mu_2)^2, \tag{54}$$

and

$$\Sigma_{12} = \tfrac{1}{2}\Sigma_2. \tag{55}$$

Thus, in LDA,

$$w = \Phi_3(x) = \frac{\sum_{i=2}^{M} \| p_1(x) - p_i(x) \|}{(M-1)\cdot\Sigma_{12}}. \tag{56}$$

Then, the decision function would be based on $sgn\ (\Phi_3(x) - T_3)$, where $T_3$ is a threshold derived from the training set, and all values are scaled to [0, 1].

In summary, when compared to FRM and FTRM, LDAM should be more reliable and informative since it compares the relative difference of the measures in the first two ranks with all the other measures.

## 5.4. Experiments

The experiments on rejection measurements were conducted on different databases and with different classifiers, such as LibSVM and HeroSVM. Firstly, we used the same classifier (LibSVM) on three databases to compare the experiments: CENPARMI Hindu-Arabic Isolated Numerals Database, CENPARMI numerals database, and Isolated Numerals Database in NIST Special Database 19. Details are described in Section 5.4.1. Moreover, in order to evaluate this LDAM's efficiency, we compared the experiments with two classifiers (LibSVM and HeroSVM) on the same database: CENPARMI Hindu-Arabic Isolated Numerals Database. Section 5.4.2 will provide a description of these experiments in detail.

### 5.4.1 Experiment I

In this section, we firstly describe the distribution of samples in each database, and then we illustrate the results of the experiments on each database.

The distributions of samples in each of the three databases are given below:

The CENPARMI Hindu-Arabic Isolated Numerals database contains 18,585, 6,199, and 6,199 samples in the Training, Validation, and Test sets, respectively, with the distribution shown in Table 1 of Section 4.4.

The CENPARMI and NIST numeral databases are well-known and have been tested by researchers for over twenty years; the former consists of the handwritten ZIP codes that were extracted from USPS mailed items in the early 1980's, while the latter consists of Latin numerals that were collected in the early 1990's from 3,699 forms, on which the writers were instructed to print specific numerals in designated boxes.

The CENPARMI numeral database contains 4,000 and 2,000 samples in the Training and Test sets respectively, with equal numbers of samples per class in each set. NIST Special Database 19 consists of 344,307 samples of isolated numerals in the Training set and 58,645 samples in the test set, with the distribution shown in Table 4.

**Table 4. Distribution of samples in NIST SD 19**

| Label | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Training | 34,803 | 38,049 | 34,184 | 35,293 | 33,432 | 31,067 | 34,079 | 35,796 | 33,884 | 33,720 | 344,307 |
| Test | 5,560 | 6,655 | 5,888 | 5,819 | 5,721 | 5,539 | 5,858 | 6,097 | 5,695 | 5,813 | 58,645 |

For each database, the SVM classifier was trained on the training set, tested on the test set, and LDAM was applied as a rejection criterion with a threshold of T = 0.05. The results are shown in Table 5.

**Table 5. Results of Classification and Rejection with LDAM on Test Sets**

| Database | | CENPARMI Hindu-Arabic Numerals | CENPARMI Numerals | NIST Numerals |
|---|---|---|---|---|
| # Training samples | | 24,784 | 4,000 | 344,307 |
| # Test samples | | 6,199 | 2,000 | 58,645 |
| Results without Rejection | # Correct | 6,104 | 1,962 | 57,740 |
| | Rate (%) | 98.47% | 98.10% | 98.46% |
| | # Errors | 95 | 38 | 905 |
| | Rate (%) | 1.52% | 1.90% | 1.54% |
| Results with Rejection (T=0.05) | # Correct | 5735 | 1819 | 55,664 |
| | Rate (%) | 92.51% | 90.95% | 94.92% |
| | # Errors | 17 | 6 | 174 |
| | Rate (%) | 0.27% | 0.30% | 0.30% |
| | # Reject | 447 | 175 | 2,807 |
| | Rate (%) | 7.21% | 8.75% | 4.79% |
| | Reliability | 99.70% | 99.67% | 99.69% |

It is worth noting that for these three databases, the LibSVM classifier achieved very similar recognition rates without the rejection option, varying from 98.10% to 98.47%. This shows the consistent behavior of the SVM classifier even when trained on sets of sizes with different orders of magnitude.

Then, when the LDAM was applied for rejection, the method was most effective on the NIST database, given that out of 2807 rejected samples, 732 of them (26.08%) would have been recognition errors. For the CENPARMI Hindu-Arabic and CENPARMI numeral databases, the ratios are 17.22% and 18.29%, respectively.

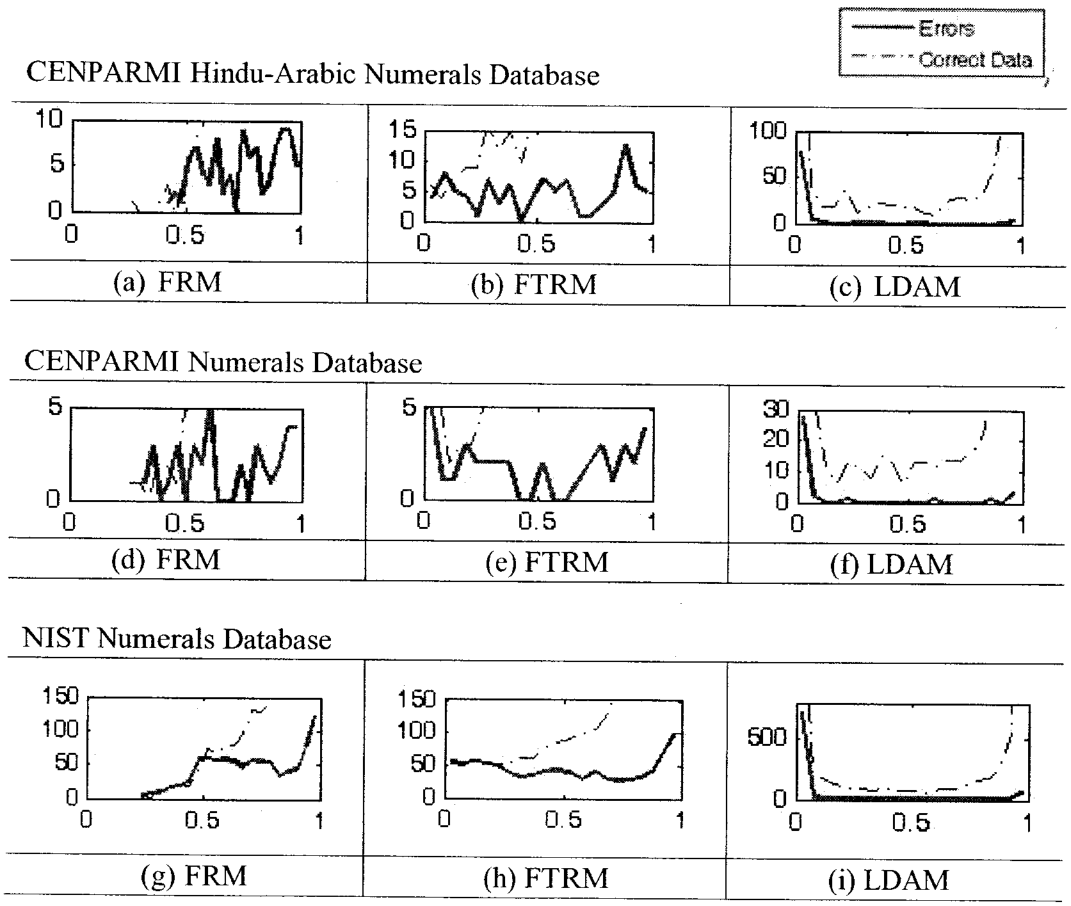
Furthermore, it is remarkable that on the three very different databases, the reliabilities achieved with the same SVM classifier and LDA rejection measurement are uniformly high, at around 99.7%. The level and consistency of these results provide solid support for the validity of the method presented in this work.

The distributions of samples according to each of the three measurement values from the experiments conducted on the three test sets in different databases are shown in Figure 18. In each graph, the horizontal axis indicates the values of each measurement (FRM, FTRM and LDAM), while the vertical axis shows the number of samples. The solid lines represent the distributions of errors, and the dotted lines represent the distributions of correctly recognized samples.

The distributions based on FRM are shown in Figure 18(a), Figure 18(d), and Figure 18(g). Although the correctly classified samples display a Gaussian distribution, the errors are distributed more evenly over ranges of confidence values (measurements), so the graphs are too flat to separate correctly and incorrectly classified samples according to FRM. When compared to FRM, FTRM [Figure 18(b), Figure 18(e), and Figure 18(h)] is more discriminating, as the range of measurements here is wider than in FRM. However, the distribution of errors in FTRM is flat as well.

LDAM is more discriminating than FRM and FTRM. This is because the errors plus correctly classified samples with low confidence values are assigned small measurements. This can be seen for all three databases in Figure 18, in which the number of errors can also drop sharply for small values of LDAM [Figure 18(c), Figure 18(f), and Figure 18(i)]. For another example, with the CENPARMI

Hindu-Arabic numerals, out of the 95 samples initially wrongly classified without rejection, 78 of them were assigned LDA measurements of less than 0.05, and would therefore be rejected with this threshold. Thus, LDAM enables a more effective reduction of potential errors with the thresholds obtained from the training set.



CENPARMI Hindu-Arabic Numerals Database

(a) FRM (b) FTRM (c) LDAM

CENPARMI Numerals Database

(d) FRM (e) FTRM (f) LDAM

NIST Numerals Database

(g) FRM (h) FTRM (i) LDAM

**Figure 18. Distributions of samples in the test sets according to the three measurements for CENPARMI Hindu-Arabic, CENPARMI, and NIST Numeral Databases**

As indicated in Table 5, after processing the three datasets by the LibSVM classifier and applying LDAM for rejection, there were 17, 6 and 174 misclassifications in the test sets of the CENPARMI Hindu-Arabic, CENPARMI, and

NIST numeral databases, respectively. All of these images are shown in Figure 19.

In Figure 19, for the Hindu-Arabic numerals, most of the misclassifications (12 out of 17) were due to the confusing styles that could be used in writing the numerals ‘2’ and ‘3’, that would be indistinguishable even to human beings. This could be due to the fact that writers in different regions/countries can write the numerals ‘2’ and ‘3’ in identical styles, and this problem appears to be more severe in this data set than the one reported in [1], where there were confusions between ‘2’ and ‘3’ in only 5 samples out of 10,000.

For the other numeral databases, some of the misclassifications are very understandable as they may have been the results of incorrect labeling during the process of data collection and preparation by humans. However, the causes of other misclassifications are far from obvious to the human eye, and are probably the result of falling on the wrong side of certain threshold(s) in the automatic recognition process. Some of the errors in the NIST database may have been caused by the mislabeling of certain samples (or by writers’ mistakes in printing the numerals indicated) in the test and training sets. Due to the immense size of the latter (344,307 samples), the effort required to verify and ensure the correct labeling of all data might have been too immense to be practical.

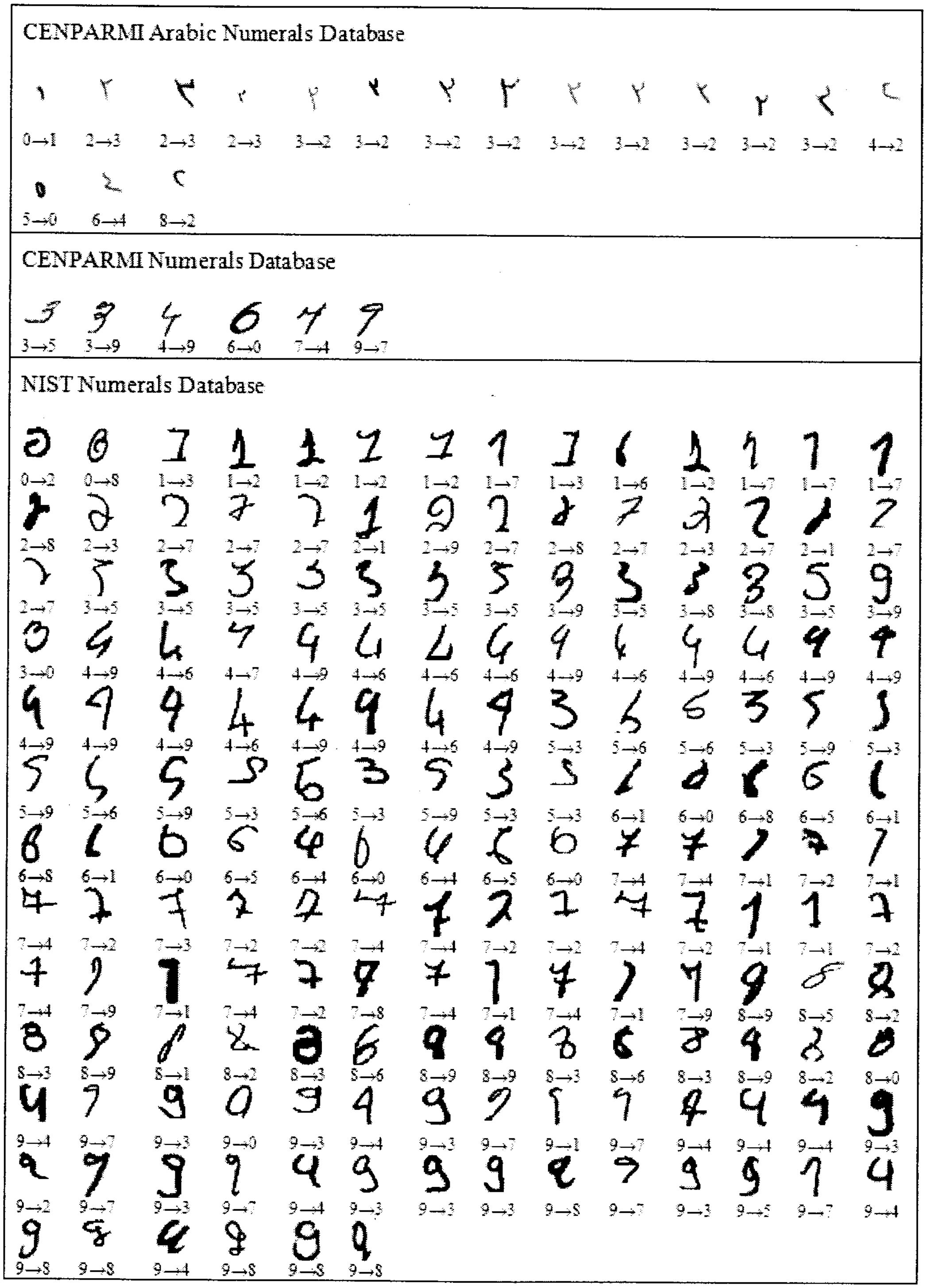


**Figure 19. Incorrectly classified samples from the three databases**

### 5.4.2 Experiment II

In the previous section, we conducted experiments on different databases with the same type of classifier. However, as mentioned before, outputs from classifiers may be different, with either posterior probabilities (LibSVM) or distances (HeroSVM). Therefore, we conducted more experiments on the same database but with different classifiers (LibSVM and HeroSVM) to compare the effectiveness of LDAM. All of the experiments in this section were designed on the CENPARMI Hindu-Arabic Isolated Numerals Database, which was described in Section 5.4.1.

In Figures 20 and 21, we show the distributions of samples for the three measurements (FRM, FTRM, and LDAM) obtained from the test sets for outputs of LibSVM and HeroSVM, respectively. The solid lines represent the distributions of errors, and the dotted lines represent the distributions of correctly recognized samples.

With LibSVM, the distributions of the training and test data are similar for FRM (the distributions of training data were also shown in Chapter 5, Figure 17(a)). Although the correctly classified samples display a Gaussian distribution, the errors are distributed almost evenly for confidence values (measurements) ranging from 0.4 to 1, which means that correctly and incorrectly classified samples cannot be distinguished based on FRM. When compared to FRM, FTRM is more discriminating, as the range of measurements in FTRM is wider than in FRM (the range is (0, 1) for FTRM). However, the distribution of errors in FTRM is rather even as well. LDAM is the most discriminating of the three measurements, because the errors together with correctly classified samples with low confidence values are assigned small measurements. In LDAM, most incorrectly classified samples (78/95) retain very low measurements (less than 0.05), which results in a high reliability (99.7%) when the threshold is set at this value.

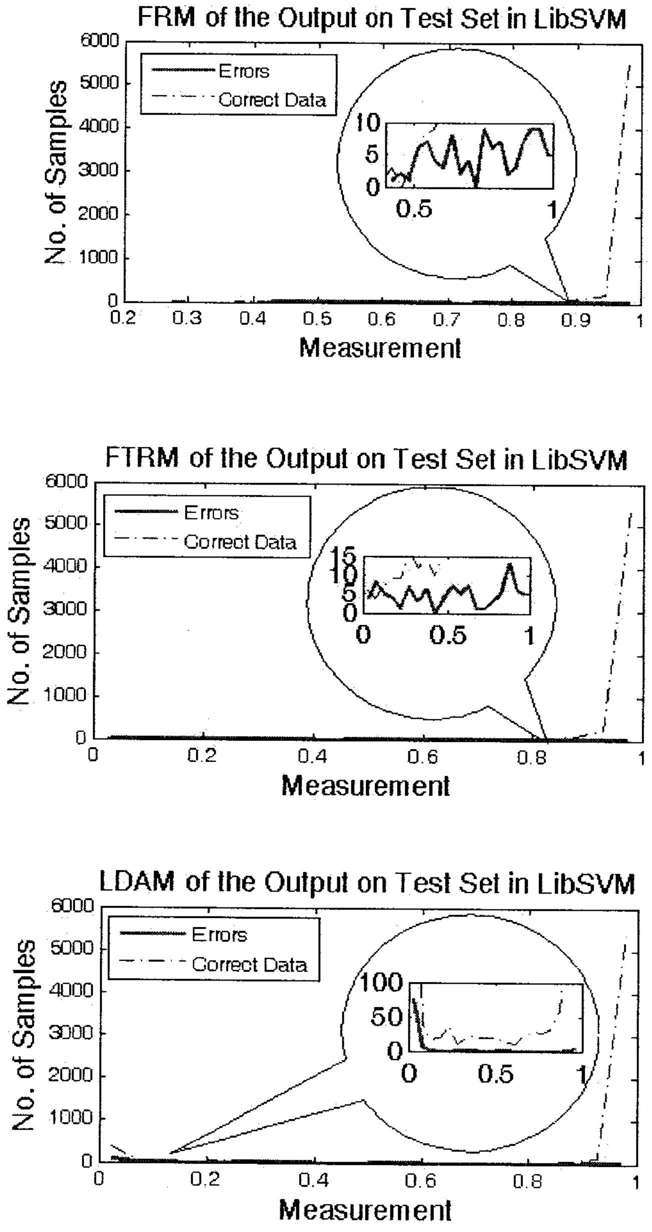


**Figure 20. Distributions of the three measurements on the Test Set with LibSVM**

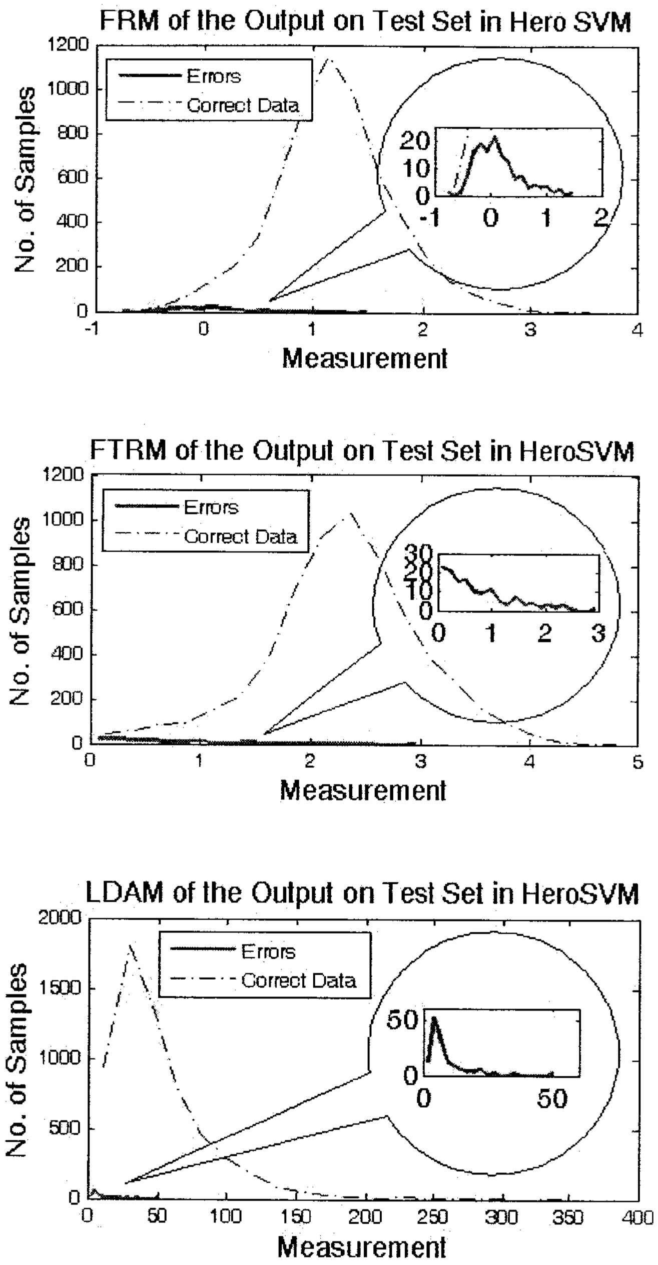
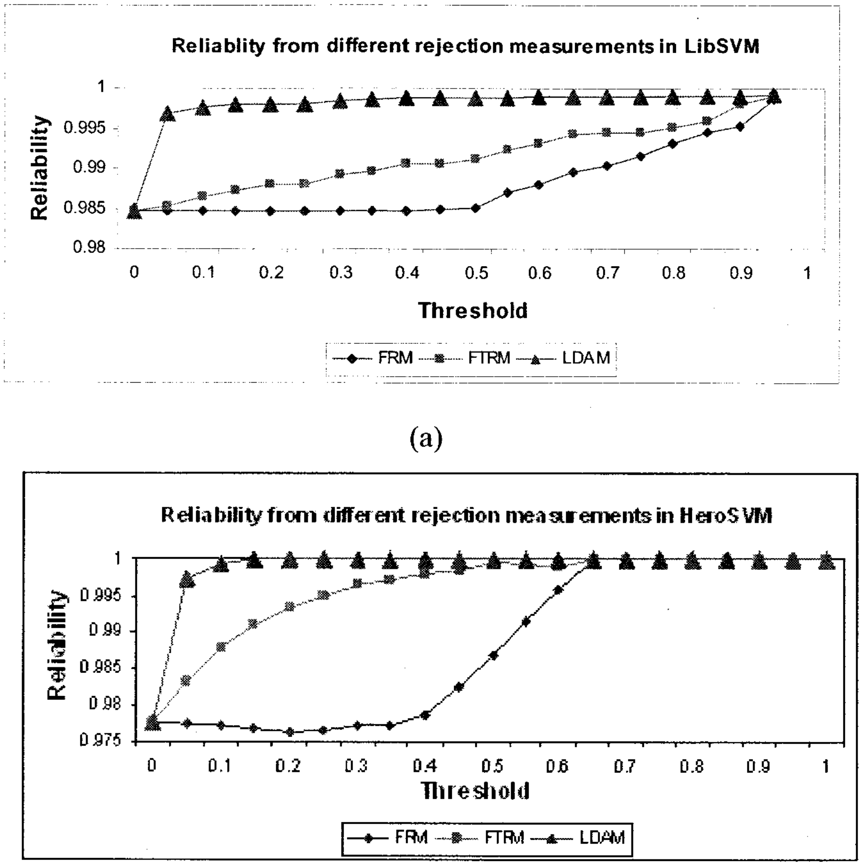


**Figure 21. Distributions of the three measurements on the Test Set with HeroSVM**

In Figure 21 (which shows the results for HeroSVM), it can be observed that for FRM, both the correctly and incorrectly classified samples display Gaussian distributions but with overlapping ranges of measurement values, which means that correctly and incorrectly classified samples cannot be adequately distinguished. When compared to FRM, FTRM is more discriminating, as the peaks of the distributions are located farther apart (at 2.35 and 0.08, respectively). However, the distribution of errors in FTRM is even as well. LDAM is more discriminating than FRM and FTRM because, similar to the distributions in LibSVM, most errors are assigned small measurements, and the distribution of errors decreases sharply from the peak. With LDAM, most incorrectly classified samples (111/139) have very low measurements, from 0 to 10 out of a total range from 0 to 351, and this yields a high reliability of 99.51%.

These experimental results show that LDAM enables the rejection of samples classified with low reliability when the thresholds are obtained from the training set for both LibSVM and HeroSVM.

The performances using different thresholds with the various measurements on the CENPARMI's Hindu-Arabic Numeral test set are shown in Figure 22. As illustrated, when the threshold $T_3$ is set to 0.05 in LibSVM, the reliability increases to 99.69% with LDAM, while the reliabilities with FRM and FTRM are 98.48% and 98.52%, respectively. Similarly, when the threshold $T_3$ is set to 0.01 with HeroSVM, the reliability increases to 98.05% with LDAM, while the reliabilities with FRM and FTRM are 97.76% and 97.87%, respectively. These results show that LDAM is the most effective measurement for obtaining reliable results from both LibSVM and HeroSVM when applied to the CENPARMI's Hindu-Arabic Numeral Database.



(a)

(b)

**Figure 22. Reliability with different thresholds used on the three measurements in: (a) LibSVM (b) HeroSVM**

When the reliabilities of LDAM on LibSVM and HeroSVM are 99.70% and 99.51%, respectively, there are 17 and 28 errors (out of 6199 samples) respectively, which are shown in Figure 23. Both LibSVM and HeroSVM yielded some common errors in recognition. As can be seen, these errors are reasonable since even human beings would have difficulty in recognizing them, or to distinguish between samples of "2" and "3" written in the same styles for Hindu-Arabic numerals.

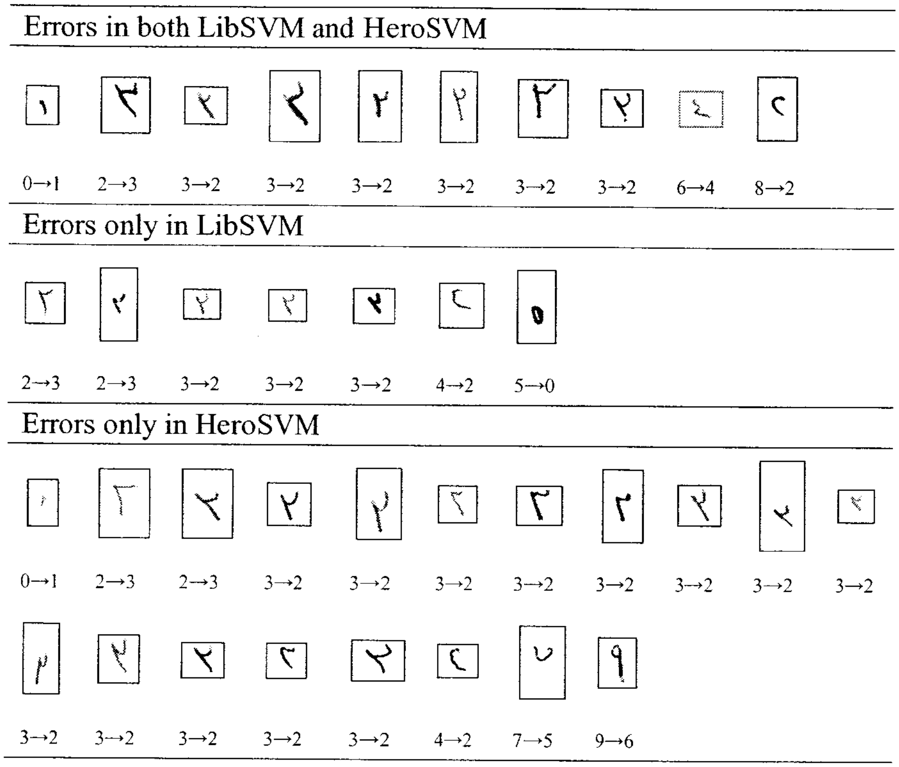


**Figure 23. Incorrectly classified images in LibSVM and HeroSVM with LDAM**

## 5.5 Conclusion

The rejection option is very useful for preventing misclassifications, which is important in applications which require high reliabilities. We designed a novel rejection criterion using the LDA Measurement (LDAM), which relies on the principle of Linear Discriminant Analysis and considers relationships among the probabilities in each output vector. It was implemented to reject the data with unreliable classification results which were produced by the supervised learning method.

The design of this LDAM incorporates information about the relationships among the probabilities or distances in the output vector of each pattern. This measurement was applied to process the training and test sets of three databases of very different sizes and on different classifiers. The recognition results indicate that a very consistent and high level of reliability can be achieved. At the same time, we compared LDAM with other measurements such as First Rank Measurement (FRM) and First Two Ranks Measurement (FTRM). The results indicate that LDAM achieved a higher reliability than the other measurements when a small threshold was set [45]. Finally, we conducted more experiments on the same database (CENPARMI's Hindu-Arabic Numeral Database) but with different classifiers (LibSVM and HeroSVM). The results show that LDAM is the most effective measurement for obtaining a high reliability with these classifiers.

# Chapter 6

# Error Categorization

In this chapter, we categorize errors and design target-oriented strategies for verification, which is applied to the patterns rejected by the supervised learning method. In general, a verifier can precisely evaluate the results produced by the classification stage to compensate for its weakness. We thereby analyze errors in the Training Set in Section 6.1, divide these errors into four categories, and figure out the corresponding strategies in Section 6.2. The experiments and error analyses after verification are also described based on different strategies in Section 6.3, and the conclusion is presented in Section 6.4.

In Pattern Recognition and Machine Learning, error analysis is vital to enhance the recognition system's performance. In fact, error analysis is not a new term in the manufacturing industry, such as in the manufacturing of electronics. It is an important discipline used in the development of new products and for the improvement of existing products. In the manufacturing industry, error analysis is the process of collecting and analyzing data to determine the cause of a failure. Similarly, we analyze errors in Pattern Recognition and Machine Learning, and accordingly define the strategies to correct errors belonging to different categories. These strategies should have the capability of being transferred to different applications so that the cost of instability in a learning system can be reduced.

## 6.1 Error Analysis in the Training Procedure

We analyzed the data in the Training Set in order to define the error categories and determine the strategies in verification. Rather than reviewing a database with 2D images, we can apply error analysis to any pattern recognition system. Thus, instead of visually reviewing the images, we first analyzed the data based on their statistical distributions.

Firstly, we analyzed the data based on their performance with Principal Component Analysis (PCA) [71]. PCA is used in each class to investigate and understand the distributions of the data in the feature space. PCA involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables named principal components. Its operation can be thought of as revealing the internal structure of the data in a way which best explains the variance in the data. If a multivariate dataset is visualized as a set of coordinates in a high-dimensional data space (e.g., 1 axis per variable), then PCA supplies the user with a lower-dimensional picture, which is a "shadow" of this object when viewed from its (arguably) most informative viewpoint. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible.

PCA in each class of the supervised learning method is shown in Figure 24. In each graph, the horizontal axis indicates the first principal component, while the vertical axis shows the second principal component.

Accordingly, although the distributions of each class are not uniform, they more or less are Gaussian in shape except for Classes 1 and 2. The distribution of Class 1 has a shape like a crescent moon due to size normalization, but it has one centre. Moreover, the distribution of Class 2 has more than one centre, and it seems that the data in Class 2 may have multi-variation (more than one sub-class) within the class. We should apply unsupervised learning in Class 2 to cluster the samples into sub-classes. This analysis can be proven by visually analyzing the data in the database. This theory exactly matches the 2D patterns in the database. As mentioned before (in Figure 4 in Chapter 1), the data in Class 2 have different shapes. Details will be illustrated in Chapter 7.

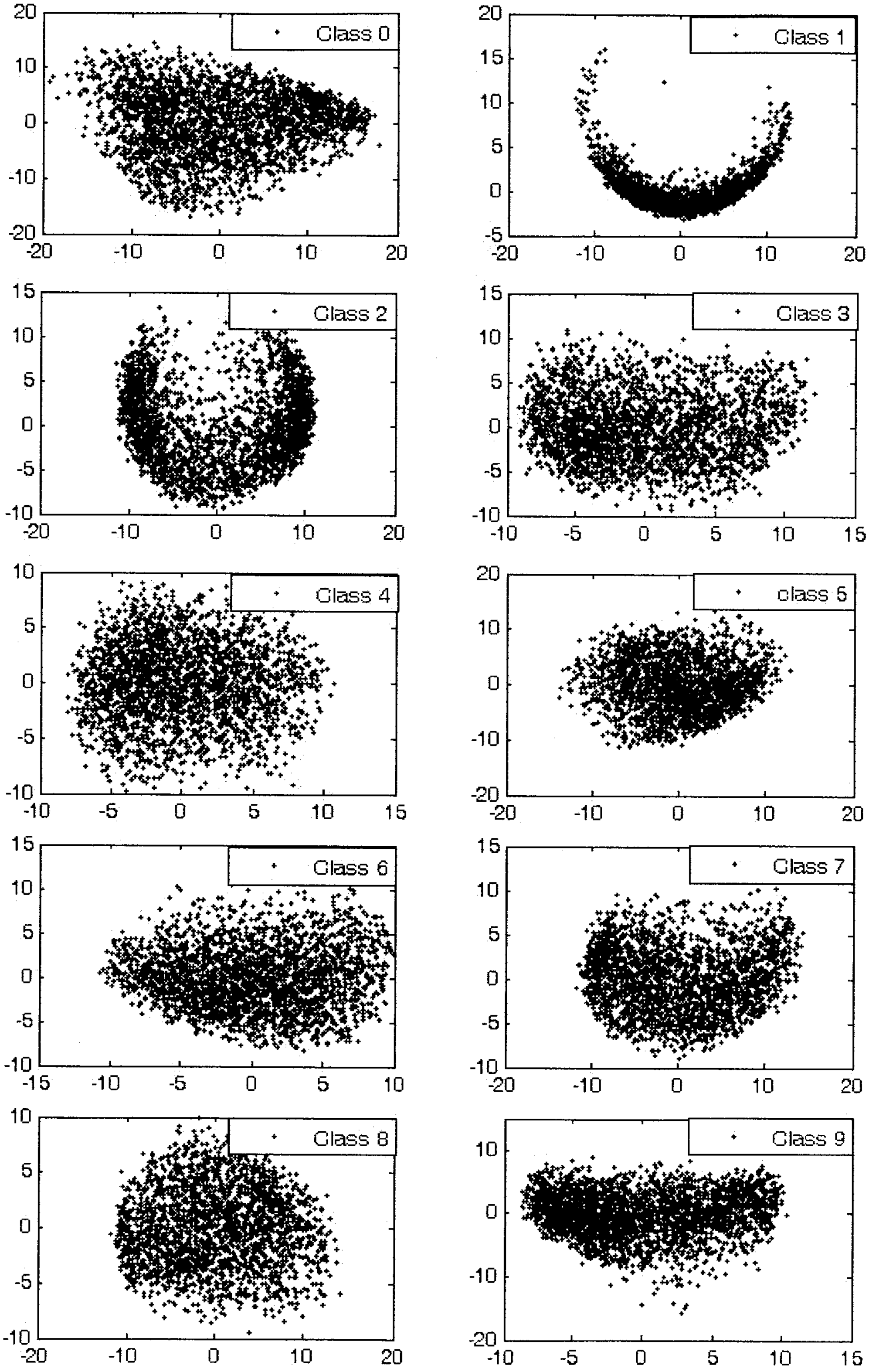


**Figure 24. PCA in each class of the supervised learning method**

Since the amount of training data is huge, it is impossible to analyze all the data one by one. Hence, let us analyze all the errors in the Training Set so that we can analyze/categorize them. In this system, a total of 195 errors occurred in the Training Set. The confusion matrix is shown in Table 6.

**Table 6. Error confusion matrix in the Training Set**

| | | Output | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Truth Label | 0 | 2,647 | 5 | | | | 1 | | | | |
| | 1 | 8 | 2,456 | | | | | 1 | | | |
| | 2 | | | 2,542 | 111 | 1 | | | | | 1 |
| | 3 | | | 64 | 2,503 | | | | | | |
| | 4 | | | | | 2,447 | | | | | |
| | 5 | | | | | | 2,253 | | | | |
| | 6 | | | | | | | 2,477 | | | 1 |
| | 7 | | | | 1 | | | | 2,338 | | |
| | 8 | | | | | | | | | 2,321 | |
| | 9 | | | | | 1 | | | | | 2,800 |

There were 175 (89.74%) errors between classes consisting of numerals 2 and 3, 13(6.67%) errors between Class 0 and Class 1, and 7(3.59%) errors among the remaining classes. Therefore, the errors in this HAHNR could be divided into three main Groups: 1) errors between Class "2" and Class "3"; 2) errors between Class "0" and Class "1"; and 3) errors not belonging to Groups I and II. Since the errors in 3) could be mislabeled by human operators or could be misclassified as shapes that are similar to the predicted classes, we accordingly divided this category into two. The first category has errors mislabeled by human operators, and the second category has errors misclassified due to similar shapes to the predicted classes. These errors were matched to the error categories in [98].

In [98], misclassification errors are categorized into four types based on the costs: (I) Error cost conditional on time of classification, (II) Error cost conditional on individual case, (III) Error cost conditional on feature value, and (IV) Error cost conditional on classification of other cases. The adaptation of these categories to the HAHNR database will be described in the following section.

## 6.2 Error Categories Based on the Cost of Misclassification

It is obvious from our research (in Section 6.1) that a certain type of error may be conditional on the circumstances, and thereby we should not assume that the errors have a fixed cost. In the following subsections, we will describe errors in our adapted categories due to different costs and figure out strategies for different categories.

### 6.2.1 Error Cost Conditional on Time of Classification

In a time-series application, the cost of a classification error is dependent on the timing [98]. Without proper timing, some confusing shapes may appear such that even a human being could have difficulties to tell them apart [94]. These errors can be corrected if the training/testing could be given a sufficient amount of time.

In handwriting recognition, if we can learn about a writer or about his/her writing style with a sufficient amount of time before or during the recognition process, some confusing shapes could be classified correctly. In general, most writers may keep a consistency in their writings, while the confusing shapes may be written by other writers. Some samples in Classes 2 and 3 in the HAHNR database could be distinguished with writers who had consistent writing styles. Since we will describe the entire procedure of verification between Classes 2 and 3 in Chapter 7, which matches this error category, the verification procedure will only be briefly summarized in this subsection.

In the HAHNR database used in our study, many errors occurred among classes “2” and “3” in the training procedure, and they may originally have had confusing shapes. Samples from the first six writers in the database were chosen for numerals 2 and 3, which are illustrated in Figure 25. Samples in the same column are written by the same writer. This figure shows that samples with bounding boxes have shapes that can be confused between the two classes.

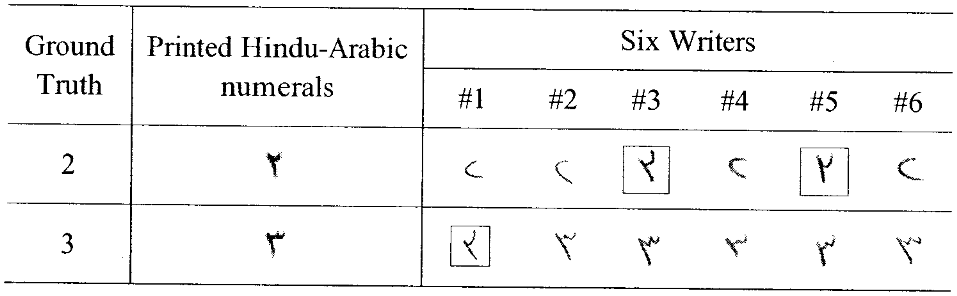

Ground Truth
Printed Hindu-Arabic numerals
Six Writers
#1
#2
#3
#4
#5
#6
2
٢
3
٣

**Figure 25. Some samples of handwritten Hindu-Arabic numerals “2” and “3”**

When collecting enough samples in a certain class by a writer over a period of time, this person’s writing style in this class can be learned and applied to the predicted class in the testing procedure. However, since this HAHNR is based on an off-line handwriting database, and tracking and learning each writer’s writing style is difficult, strategies to retrieve the timing property in this database need to be found and examined.

Some writer information was recorded during the data collection for the Isolated Hindu-Arabic Numeral Database at CENPARMI [4]. An ID was assigned to each writer, and this enabled us to design an unsupervised learning (clustering) process that makes use of the writing style information to validate the recognition results. As a result, two writing styles in each class (either Class 2 or 3) were identified. All writers with the combination of their writing styles in Classes 2 and 3 could be divided into four groups. These four Combined Writing Styles (CWS) are shown in Table 7. Once a person's writing style was unknown (or "too-difficult-to-detect"), this style was assigned to a Case of Rejection. Therefore, the persons' writing styles could be automatically learned or detected before the recognition process, and a pair-wise verification between Classes 2 and 3 could be effectively implemented on the samples in these two classes. For example, if the sample ' ' originated from a writer with CWS I, then it would belong to Class 3, whereas if the writer has CWS II, then it would belong to Class 2. Details can be found in Chapter 7.

**Table 7. Combined Writing Styles (CWS) for Classes 2 and 3**

| | Class 2 | Class 3 |
|---|---|---|
| CWS I | | |
| CWS II | | |
| CWS III | | |
| Case of Rejection | Unknown | Unknown |

In conclusion, errors costs conditional on time of classification are misclassifications due to lack of data in the instance level. Hence, if we are able to trace the data based on timing or retrieve the timing property in applications, error costs conditional on time of classification could be corrected. We allotted more time in the collection of CWS and recorded the writer's information in this case, thereby reducing the error costs.

### 6.2.2 Error Cost Conditional on Individual Case

The cost of a classification error may depend on the nature of the particular case [98]. These errors may have confusing natures [94], and due to the nature of these individual cases, the classification results are predictable and can be compared against the ground truth in the database.

In handwriting recognition, quite possibly the source of these errors comes from the mistakes made by human operators. When the human operators label the ground truth on a huge amount of data, they may make mistakes.

In this HAHNR, since it is a time-consuming process to match and verify all the labels to the ground truth in the original collected documents, only errors with high confidence values in the rejection measurement were verified by human operators. With regards to this verification, some mislabeled samples are shown in Figure 26.

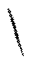
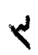
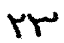
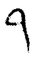
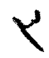
| Samples |  |  | |  | |  |  |
|---|---|---|---|---|---|---|---|
| Mislabel → Ground Truth | 2→1 | 2→3 | 3→2 | 3→23 | 3→7 | 4→9 | 9→3 |

**Figure 26. Some Mislabeled Samples in the Database**

In conclusion, error costs conditional on individual case are due to misclassifications by human operators. It is possible to tolerate these errors in the training procedure with SVM. When we train the system, we set a suitable penalty parameter of the error term in SVM that can tolerate these errors. However, if the classifier cannot tolerate a certain percentage of errors, then the errors in this category should be detected and removed/corrected from the training set in the database.

### 6.2.3 Error Cost Conditional on Feature Value

The cost of making a classification error with a particular case may depend on the value of one or more features in the case [98]. Although some samples can be easily recognized by human beings, quite a few errors may occur in machine recognition [94]. These errors should be corrected when we extract or combine different feature sets.

In handwriting recognition, a large number of errors may occur between two classes in the training procedure, but they may not necessarily have confusing shapes.

In this HAHNR, errors between numerals 0 and 1 were due to size normalization in pre-processing. Errors between these two classes in the training procedure are shown in Figure 27.

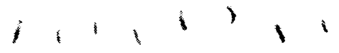
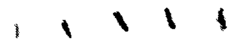
| Ground Truth | Printed Hindu-Arabic numerals | Errors in the Training Set |
|---|---|---|
| 0 | ٠ |  |
| 1 | ١ |  |

**Figure 27. Errors between numerals "0" and "1" in the Training Set**

In conclusion, error costs conditional on feature value are misclassifications due to data in the feature level. Size normalization may cause some samples in these two classes (Figure 27) to become confusing in shape, so extracting different feature sets without size normalization should be implemented to improve the recognition rate. Accordingly, we trained and tested the samples of numerals in Classes 0 and 1 with their heights and widths. Once samples were recognized as 0 or 1 in the rejection process during testing, we verified the recognition results with these new feature sets (global features) on the original samples without size normalization.

### 6.2.4 Error Cost Conditional on Classification of Other Cases

The cost of making a classification error with one case may depend on whether errors have been made with other cases [98]. Some errors in this category occur because of the poor quality of images, and the recognizer cannot classify them accurately. Other errors occur in a few particular cases due to their similarities to other classes, but these errors only occur sparingly.

In handwriting recognition, errors that occur outside of the first three error cost categories can be grouped to this category.

In this HAHNR database, most images have a good quality. Only mislabeling and errors that occur outside of the errors between numerals 0 and 1, and between numerals 2 and 3 should belong to this category. All of the errors found in this category of the training procedure are shown in Figure 28. All of the printed Hindu-Arabic numerals can be seen in Figure 3 of Chapter 1.

| Samples | | | | |
|---|---|---|---|---|
| Mislabel → Ground Truth | 4→2 | 5→0 | 6→1 | 9→6 |

**Figure 28. Errors in Classification of other cases in the Training Set**

In conclusion, error costs conditional on classification of other cases are misclassifications due to random factors. These errors had very similar shapes to the samples belonging to other classes. These errors only occurred rarely and could not be categorized. Every recognition system may produce some errors in this category. Since these errors may not mislead the classifier significantly, the penalty parameter of the error term in SVM can tolerate these errors. Thus, we can keep them in the training procedure.

## 6.3 Experiments and Results

All of the experiments in this Chapter are still based on the supervised learning method with an SVM. Two parameters (c, $\gamma$) were determined in supervised learning, where c is the penalty parameter of the error term and $\gamma$ is the kernel parameter. Since the error costs conditional on individual case and the error costs conditional on classification of other cases could be tolerated with a certain value of c, it had to be greater than zero in this study. These parameters were optimally chosen by cross-validation via a parallel grid search in the training set, and these optimal parameter values were then applied on the test set.

As mentioned before, the recognition rate achieved in supervised learning was 98.47% on the test set without verification, and there were 95 errors out of 6199 samples. When we applied LDAM in Chapter 5 [44] with an optimal threshold (T = 0.05), the recognition rate, error rate, and reliability were 92.40%, 0.27%, and 99.70%, respectively, and 17 errors could not be rejected due to the high confidence values in LDAM. Although the reliability is very high (99.70%), the recognition rate (92.40%) is quite low compared with 98.47%. Thus, verification on rejected patterns should be applied so that the recognition rate can be improved while maintaining high reliability. In a total of 447 rejected samples, 342 were recognized as class 2 or 3, and 40 were recognized as class 0 or 1. We kept the recognition results of the other rejected samples.

After verification with sub-classifiers and comparing the ground truth data of the original documents (details of this procedure can be found in Chapter 7), the recognition rate increased to 99.05%, and the number of errors decreased from 95 to 59, and almost 38% of previously misclassified samples could now be correctly recognized with this verification procedure (Table 8). It was difficult to have such an improvement when the number of errors was relatively small. Moreover, this database involved writers from different countries/regions, so they may have written "2" and "3" with identical shapes. When the writer's information was undetected due to the small number of samples, errors may have occurred. The errors were more severe in this data set than those reported in [1] collected from writers in the same country, where there were confusions between '2' and '3' in only 5 samples out of 10,000.

**Table 8. Comparison of the performances without rejection**

| | [4] | Chapter 5 [44] | New Method |
|---|---|---|---|
| Rate (%) | 93.60 | 98.47 | **99.05** |
| # Correct | 5802 | 6104 | **6140** |
| Rate (%) | 6.40 | 1.53 | **0.95** |
| # Errors | 397 | 95 | **59** |

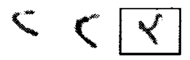
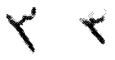
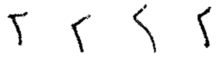
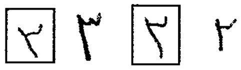
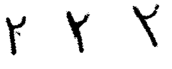
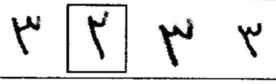
For some people whose combined writing styles in classes 2 and 3 were difficult to be determined or were unknown, we could reject them. In this case, the recognition rate, error rate, rejection rate, and reliability became 97.89%, 0.63%, 1.48%, and 99.28%, respectively, as shown in Table 9. While applying the rejection measurement based on LDAM alone, when the error rate was kept at 0.63%, the recognition rate, rejection rate, and reliability were 96.98%, 2.11%, and 99.08%, respectively, as shown in Table 9.

**Table 9. Comparison of the performances with rejection**

| | Chapter 5 [44] | New Method |
|---|---|---|
| Rate (%) | 96.98 | **97.89** |
| # Correct | 6012 | **6068** |
| Rate (%) | 0.63 | 0.63 |
| # Errors | 39 | 39 |
| Rate (%) | 2.11 | 1.48 |
| # Reject. | 131 | 92 |
| Reliability % | 99.08 | **99.28** |

All errors between classes 0 and 1 with rejection were corrected after the verification with the new feature set and its sub-classifier. However, there were still 36 errors between classes 2 and 3 after verification. These two main sources of errors arose due to the fact that some writers used contradictory styles that resulted in 2's and 3's being indistinguishable (Table 10), and there was difficulty in clustering the data accurately into sub-classes.

**Table 10. Errors due to inconsistencies in the writer's writing styles**

| Writer's ID | Writing in Class 2 | Writing in Class 3 |
|---|---|---|
| 63 |  |  |
| 176 |  |  |
| 106 |  |  |

Ten samples were mislabeled in the test set, and they are shown Figure 29. There were 13 errors with high confidence values in LDAM, and they could not be verified as shown in Figure 30.

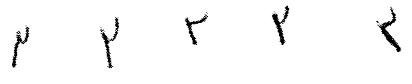
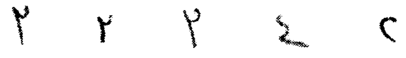
| Samples |  |
|---|---|
| Mislabel → Ground Truth | 3→2 3→2 3→2 3→2 3→2 |
| Samples |  |
| Mislabel → Ground Truth | 3→2 3→2 3→2 6→4 8→2 |

**Figure 29. Samples Mislabeled in the Test Set**

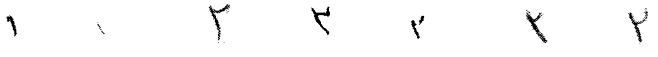
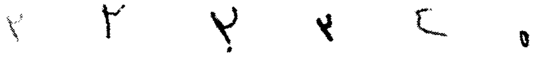
| Samples |  |
|---|---|
| Ground Truth → Output | 0→1 1→0 2→3 2→3 2→3 3→2 3→2 |
| Samples |  |
| Ground Truth → Output | 3→2 3→2 3→2 3→2 4→2 5→0 |

**Figure 30. Incorrectly classified images with high confidence values in LDAM**

## 6.4 Conclusion

In pattern recognition, error minimization should be the target of most applications. Errors should be categorized based on their features and categorization strategies should be implemented. Therefore, verification based on error categories should be designed and applied after recognition.

We have summarized errors based on Turney's research and adapted them by dividing errors in HAHNR into four categories, based on the different costs of misclassification errors. Accordingly, we studied their characteristics and analyzed the reasons for the errors in each category. Moreover, the methodologies for the detection of each error category and their corresponding categorization strategies were proposed. In order to validate this study, we matched these error categories to a recognition application, and designed a verification procedure after recognition, based on each error category.

As a result of our verification procedure, the recognition results improved significantly. Without rejections, the final recognition rate improved to 99.05%, and almost 38% of the classification errors were eliminated by using verification. When the rejection measurement was applied, the recognition rate, error rate and reliability were 96.98%, 0.63%, and 99.08%, respectively. We also assessed the verification process by holding the error rate constant at 0.63% and found that the recognition rate and reliability increased to 97.89% and 99.28%, respectively.

# Chapter 7

# Verification Based on Unsupervised Learning

In this chapter, we propose the Writing Style Verification (WSV) module based on unsupervised learning on the test set. This verification module matches the error cost conditional on time of classification, mentioned in Chapter 6. This work stems from the idea of context-based disambiguation of classification results for pairs of classes which partially overlap. The specific problem faced in this chapter is the disambiguation of classification results for Hindu-Arabic handwritten numerals which are classified as "2" or "3". The contextual information used in this case is the writing style. We define a Confusing Pair (CP) of clusters and a Writing Style (WS) and devise a methodology to automatically detect a CP and WS with unsupervised learning in Section 7.2. The experiments and error analysis based on writing style verification are described in Section 7.3.

## 7.1 Problem in Writing Styles

Unlike the supervised learning method, the unsupervised learning method with test data can be used to "teach" some extra information about the predicted sample. When we trace a writer's writing on a timing axis, the ambiguous shapes can be classified correctly. For example, in Figure 31 below, when we see all the writing samples through a timing axis, Numeral 3 in Writer I's writing may be confused with Numeral 2 in Writer II's writing. However, when we trace one writer's writing, e.g. Writer I, and find a lot of the written shape " ‹ " which is obviously Numeral 2, we can correctly classify the other ambiguous shapes in Writer I's writing as Numeral 3. We assume that a writer would not confuse him/herself by writing samples of two different classes with the same shape or style. This means that writers could be grouped based on their writing styles, and this prior knowledge about writers could result in more accurate recognition performances. Thus, if we can find a way to trace writers' writings automatically in machine learning, then the ambiguous shapes can be classified correctly.

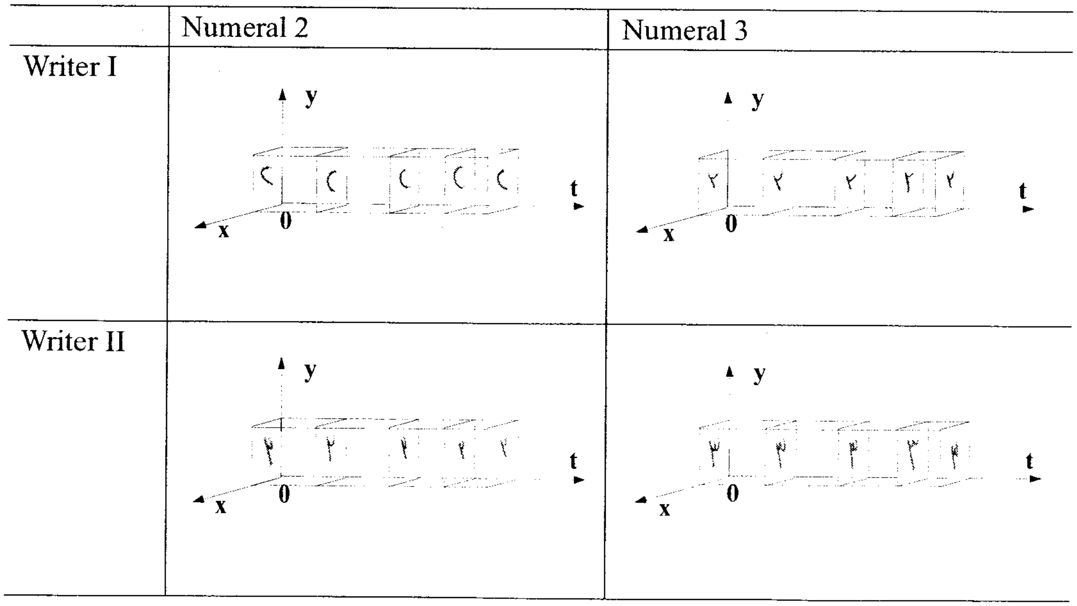


**Figure 31. Different writers' writings traced on a timing axis**

Most researchers have adapted their systems for each writer during the training procedure. Had they known that writers can be grouped according to their writing styles, it is not necessary for their systems to learn the style writer by writer during testing procedure. Instead, writing styles should have been categorized and this knowledge should have been applied to correctly classify the ambiguous shapes encountered.

It was possible to implement this process by recording some writer information during the data collection process. As mentioned in Chapter 6, this was the case for the Isolated Hindu-Arabic Numeral Database at CENPARMI, in which an ID was assigned to each writer. This enabled us to design an unsupervised learning (clustering) process that makes use of the Writing Styles (WS) Information to validate the recognition results.

## 7.2 Writing Styles Design

In this study, we will apply an unsupervised learning (clustering) process within each of the two confusing classes (2's and 3's), and the number of clusters will be determined automatically. Clusters of different classes containing samples with very similar shapes will form a confusing pair (CP). Accordingly, we can define the writing styles based on the clusters in each class. All the writers will be assigned to a group with a known writing style or a group with an unknown writing style. Then, when we know the writing style of a sample, this sample will be assigned to the correct class. The next three subsections include the definitions of Confusing Pairs (CP) and Writing Styles (WS) (Section 7.2.1), an explanation of how CP and WS are detected (Section 7.2.2) and the process of finding them in HAHNR (Section 7.2.3).

### 7.2.1 Definitions of Confusing Pairs & Writing Styles

For a classification problem with the two classes $W_i (i = 1,2)$, only the samples close to the decision boundary of their class may be confused with the data from the other class. We propose to identify these confusing samples through the unsupervised learning (clustering) process.

Suppose that for $i = 1, 2$, the data from class $W_i$ is divided into $k_i$ clusters (sub-classes) $\{W_i^j\}$, each with centre $c_i^j$, where $j = 1, 2, \ldots, k_i$. The distance between any two clusters is defined as the Euclidean distance between their centres. For $i = 1, 2$, we define the smallest intraclass distance between clusters in $W_i$ as $IAD_i$ $= min\ d(W_i^m, W_i^n)$ for all m, n in $\{1,2, \ldots, k_i\}$, m≠n.

We then determine the pair of clusters $W_1^{ii}$ in $W_1$ and $W_2^{jj}$ in $W_2$ (with $1 \leq ii \leq k_1$ , $1 \leq jj \leq k_2$ ) such that $d(W_1^{ii}, W_2^{jj})$ $= \min d(W_1^m, W_2^n)$ , for $1 \leq m \leq k_1$, $1 \leq n \leq k_2$.

If this minimum interclass distance $d(W_1^{ii}, W_2^{jj})$ is smaller than the minimum intraclass distance $IAD_i$ for i = 1, 2, then $W_1^{ii}$ and $W_2^{jj}$ are considered to be a confusing pair (CP) of clusters. This is reasonable because if the distance between two clusters from different classes is smaller than the distances between clusters of each class, then it will be difficult for a classifier to distinguish between the former two clusters.

On the other hand, if clusters $W_1^m$and $W_2^n$ do not form a confusing pair, then they can be considered together as a consistent style of writing a pair of numerals such as 2's and 3's, and this is denoted by WS. In the following section, we will describe the procedure for identifying a confusing pair (CP) of clusters.

### 7.2.2 CP Search and WS Detection with Unsupervised Learning

In order to search for a CP and a WS, we apply the well-known K-means clustering method to each class iteratively, until a CP is located or a stopping criterion is satisfied. Initially, each class is divided into two clusters ($k_1 = k_2 = 2$), and we search for a CP. As this is based on the minimum interclass distance, the number of such pairs should be either 0 or 1. We search until the CP is found or until all clusters have been considered. If no CP is found in the search, then the search is repeated with the number of clusters increased by 1 (from 2 to 3, and from 3 to 4, etc.)for one class. This process can continue until a pre-defined criterion (such as the maximum number of iterations) is satisfied.

Once the CP is found, the consistent writing styles (WS) can be determined from the consistent pairs. The statistical results of each writer's writings in each sub-class (a sub-class is a cluster of a class) are then used to assign the writer to a WS, after which his/her writings of ambiguous shapes can be assigned to the correct classes. This process is described below.

### 7.2.3 The Process of Finding CP and WS in HAHNR

Since most recognition errors in this HAHNR are due to confusions between samples in the Classes 2 and 3, we search for a CP and determine the WS for these two classes. Initially, the parameters are $k_1 = k_2 = 2$ from the two classes, as described in subsection 7.2.2. If a CP is found in this search, then the number of WS will become $k_1 \times k_2 - 1 = 3$. With two clusters in each class, the distances between each pair of centres are shown in Table 10, where Sub-class 21 (SC21) denotes cluster 1 of class 2, etc.

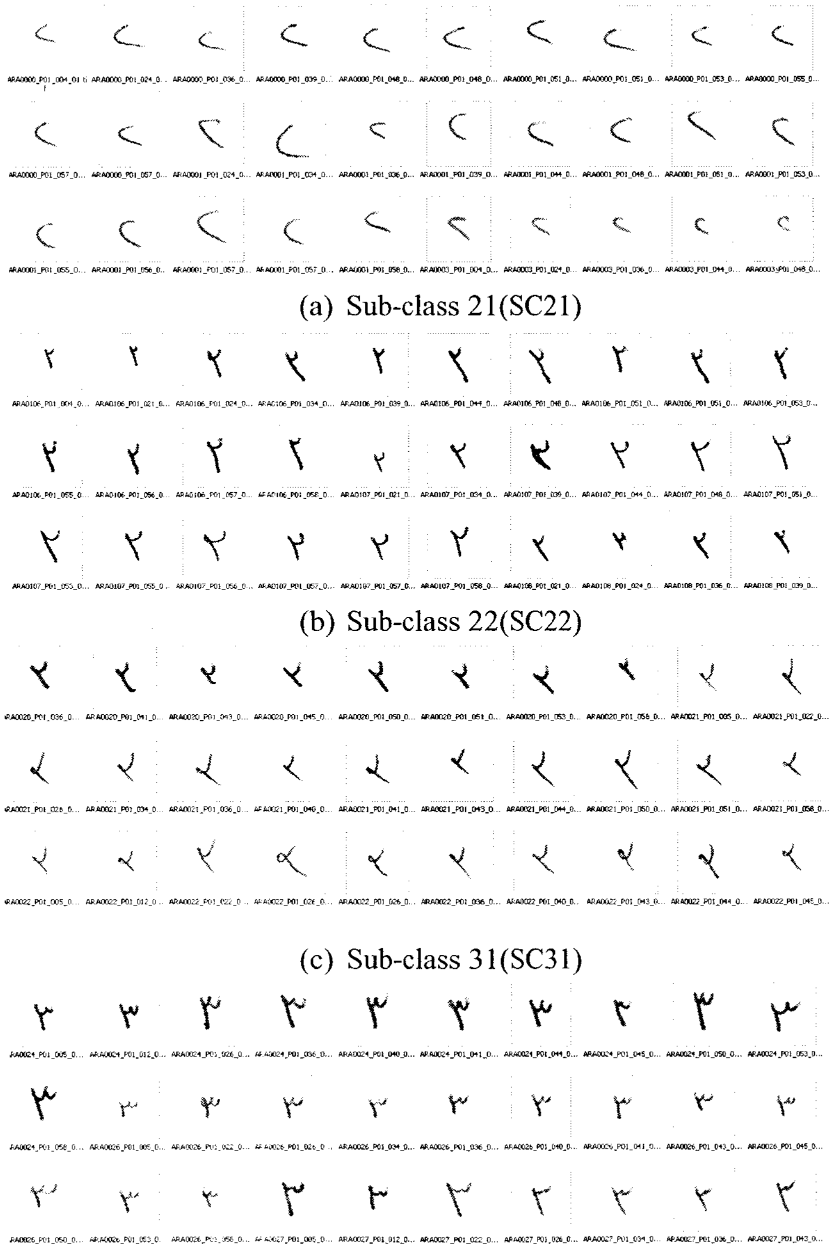
**Table 10. Distances between pairs of centres in Classes 2 and 3**

| Sub-class (SC) | 21 | 22 | 31 | 32 |
|---|---|---|---|---|
| 21 | 0 | 6.62 | 6.46 | 7.56 |
| 22 | 6.62 | 0 | **2.66** | 3.63 |
| 31 | 6.46 | 2.66 | 0 | 4.42 |
| 32 | 7.56 | 3.63 | 4.42 | 0 |

In this case, the distance $d(SC22, SC31) = 2.66$ is the minimum interclass distance and it is also smaller than the two intraclass distances of classes 2 and 3, such as $d(SC21, SC22) = 6.62$, and $d(SC31, SC32) = 4.42$. So, in this case, SC22 and SC31 form a CP, and the search stops.

From our experiments, some randomly selected samples in each sub-class are shown in Figure 32. It is obvious that the samples in SC22 and SC31 form a CP. In this case, we can then categorize the writing of 2's and 3's into three valid combined writing styles (CWS) by eliminating the confusing combination of (SC22, SC31) with the assumption that a writer would not write 2's and 3's in almost identical shapes.

Table 7 in Chapter 6 lists examples of all three resulting CWS. The cases for rejection arise when the writing styles cannot be determined due to insufficient samples from writers, or when ambiguous styles are used by one writer in two classes. These patterns are then rejected.



(a) Sub-class 21(SC21)

(b) Sub-class 22(SC22)

(c) Sub-class 31(SC31)

(d) Sub-class 32(SC32)

**Figure 32. Samples from four Sub-classes**

It follows that a major issue in HAHNR would be to distinguish between Class 3 in CWS I and Class 2 in CWS II. This issue could be resolved if the writer's CWS is known. For example, if the sample ' ' originates from a writer with CWS I, then it belongs to Class 3, whereas if the writer has CWS II, then it belongs to Class 2. This means that it is important to determine the CWS of a writer.

## 7.3 Experiments and Error Analysis

Since the result of this verification module is a part of the experiments in Chapter 6, details can be found in Section 6.3, and in this section, we only summarize the performance for this Writing Style Verification (WSV).

Experiments with and without WSV were conducted on the same CENPARMI Hindu-Arabic Isolated Numerals Database. The results of the proposed method are compared with those of the algorithms presented in [44]. After applying the rejection measurement based on LDAM, where the error rate was 0.71%, the recognition rate increased from 96.87% to 97.81% with the implementation of WSV while almost identical reliabilities were achieved, as shown in Table 11. Without rejections, the recognition rate increased from 98.61% to 98.97% for the present method, and over 25% of previous wrongly classified samples could now be correctly recognized with WSV.

The two main sources of errors arose due to the fact that some writers used contradictory styles that resulted in 2's and 3's being indistinguishable, and there was difficulty in clustering this data accurately into sub-classes.

**Table 11. Performance comparisons of methodologies with and without WSV**

| | With Rejection | | Without Rejection | | |
|---|---|---|---|---|---|
| | [44] | Proposed Method | [4] | [44] | Proposed Method |
| #Correct | 6005 | 6063 | 5802 | 6103 | 6135 |
| Rate (%) | (96.87) | (97.81) | (93.60) | (98.61) | (98.97) |
| # Error | 44 | 44 | 397 (6.40) | 86 (1.39) | 64 |
| Rate (%) | (0.71) | (0.71) | | | (1.03) |
| # Rejection | 150 | 92 | - | - | - |
| Rate (%) | (2.42) | (1.48) | | | |

## 7.4 Conclusion

Since there is a high degree of confusion in shapes between Classes 2 and 3 in HAHNR, most errors in any recognition system for HAHNR have been found to occur in these two classes. In this research, we designed a verification system that could detect and correctly recognize the confusing pairs with the writing style information based on the rejections from a supervised learning process. In this verification, an unsupervised learning in the test procedure helped to retrieve the hidden context information, and it helped to correct the errors with confusing shapes [46].

While this approach was motivated by and applied to the problem of Hindu-Arabic numeral recognition, it could also be adapted for other pattern recognition contexts that require the distinction between classes of highly similar patterns [47].

# Chapter 8

# Conclusions & Future Work

In this chapter, we summarize the contributions of this thesis (Section 8.1) with some concluding remarks and address some possible future research directions (Section 8.2). In this thesis, many efforts have been devoted to improving the learnability of a pattern recognition system in the classification/prediction and verification process. From a practical perspective, this approach was motivated by and applied to the problem of handwritten Hindu-Arabic numeral recognition.

Our methodologies could be adapted for other pattern recognition or machine learning contexts that require the distinction between classes of highly similar patterns.

## 8.1 Summary

In pattern recognition, error minimization should be the target of most applications. In order to work toward a task-oriented model of learning, reduce errors, and improve the management of interactions between the learning process and pattern recognition, we designed a novel semi-supervised learning model. This model was built with the intention of defining the boundaries among classes, including the design of an effective rejection measurement, and we utilized multiple verification modules based on different error categories. These modules included a Writing Style Verification (WSV) process to retrieve the information that could not be retrieved in supervised learning. These samples which had been rejected by Linear Discriminant Analysis Measurement (LDAM) in supervised learning, were verified by WSV. In conclusion, this thesis presents some beneficial solutions to the problem of pattern recognition and has five main contributions:

1) By simulating a human being's learning and cognition, we designed a novel Semi-Supervised Learning (SSL) system with a rejection option which broadens the definition of a standard SSL. Formerly, unlabeled data have only been used as complementary data to modeling. In this study, unsupervised learning has been applied with unlabeled data to retrieve extra information (patterns' spatial properties). Thus, semi-supervised learning should be a learning procedure that we should apply not only to generate models with the labeled data but also with extra information obtained/retrieved from the unlabeled data.

2) In addition to the object information, the context information (task constraints) should be helpful in handwriting recognition. Accordingly, retrieval of context information should be considered to disambiguate the confusing shapes between two overlapping classes. When researchers work on isolated offline handwriting recognition, a prior knowledge is always ignored or limited in its

usage. Beyond the importance of context information, knowledge of how to automatically extract contextual information should be taken into consideration [34]. Thus, we worked on context knowledge retrieval in this study so that we could categorize the confusing shapes based on the retrieved context information, which is a writer's writing style.

3) Error minimization and rejection obligation are two strategies used to achieve a high reliability while maintaining a high recognition rate. Hence, in supervised learning, we designed a recognition system with some state-of-the-art technologies. We performed noise removal, grayscale normalization, and size normalization in image pre-processing. Gradient features were extracted from the processed images, and we applied SVM with a Radial Basis Function (RBF) kernel to do the classification. Moreover, based on a theoretical analysis of the trade-off in the error, rejection, and recognition rates of a classifier system, we successfully designed a novel rejection criterion using the Linear Discriminant Analysis Measurement (LDAM) to evaluate the results from classification. The LDAM relies on the principle of LDA and considers the confidence values of the classifier outputs and the relations between them. The LDAM was implemented to reject the data with unreliable classification results which were produced by supervised learning and to potentially reduce the errors. As a result, it represents a more comprehensive measurement than the traditional rejection measurements.

4) Verifications based on categorized errors compensate for the classifier's weakness. In this real-life application, before discussing the development of a system to reduce errors and achieve a high reliability, we should also study misclassified data and find ways of preventing their occurrences. Therefore, we analyzed and categorized the errors in the training procedure so that we could understand the reasons for the errors and design target-oriented verifiers in the testing procedure. In these categories, one effective strategy based on the writers' writing styles with unsupervised learning on the test set was successfully designed.

5) The designed OCR engines were applied to Hindu-Arabic handwritten numerals, and they achieved a high recognition performance. The final recognition rate was increased to 99.05%, significantly higher than the performance (93.60%) of [4] on the same database. The number of errors decreased from 95 (in supervised learning) to 59, and almost 38% of previously misclassified samples could now be correctly recognized with this verification. When the rejection option was applied, the recognition rate, error rate and reliability were 97.89%, 0.63%, and 99.28%, respectively.

## 8.2 Future Research

While the method presented in this thesis has been implemented for handwritten numeral recognition, it is really much more general in nature and can be applied to most pattern recognition contexts (e.g. signature recognition, fingerprint recognition, face recognition, bioinformatics, etc.). Although several models and measurements have been proposed, the work is far from finished, and future research may include the following challenging problems:

1) We should keep on conducting research on human learning and cognition so that we can guide or enlighten the research in machine learning and pattern recognition.
2) Although the database on which we conducted experiments represents a large number of samples, the diversity of writings, and even the writers' I.D.'s, etc., more information should be recorded during the data collection process. For instance, if the nationality which reflects the spatial factors of writers is recorded during the data collection process, it could be easier to define the people's writing styles accordingly.
3) In this thesis, we conducted experiments on gradient features and used an SVM classifier. In the future, these factors (e.g. choosing different features and classifiers, applying a multiple classifier system, etc.) can be taken into consideration to potentially improve the system's performance.
4) We applied the methodology of this thesis to semi-supervised learning so that we could reject the data with unreliable classification results produced by supervised learning. In the future, we can apply the LDAM rejection method to training procedures or to multi-classifier systems in which the measurement level outputs are generated. In addition, we may design more effective measurements to evaluate the outputs from classifiers.

5) Errors in this thesis have been grouped into four categories. In the future, we could incorporate other error categories or design new strategies based on these error categories.

6) In this thesis, we automatically retrieve some extra information from the database (writers' writing styles) by unsupervised learning, which indirectly reflects the spatial factor of the database. In the future, we should discover and retrieve more knowledge or information from the databases in order to classify/predict patterns more accurately.

7) While this approach was motivated by and applied to the problem of Hindu-Arabic numeral recognition, it could also be adapted for other pattern recognition contexts that may require the distinction between classes of highly similar patterns.

# References

[1] S. Abdleazeem and E. El-Sherif, "Arabic handwritten digit recognition," *International Journal on Document Analysis and Recognition*, vol. 11, no. 3, 2008, pp. 127-141.

[2] E., Amsel, R. Langer, and L. Loutzenhiser, "Do lawyers reason differently from psychologists? A comparative design for studying expertise," *Complex problem solving: Principles and mechanisms*, R. J. Sternberg & P. A. Frensch, Eds., Hillsdale, USA: Lawrence Erlbaum Associates, 1991, pp. 223-250.

[3] Y. Al-Ohali, M. Cheriet, and C.Y. Suen, "Database for recognition of handwritten Arabic cheques," in *Proc. of the 7th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2000)*, Amsterdam, the Netherlands, 2000, pp. 601–606.

[4] H. Alamri, J. Sadri, C. Y. Suen, and N. Nobile, "A novel comprehensive database for Arabic off-Line handwriting recognition," in *Proc. of 11th International Conference on Frontiers in Handwriting Recognition (IWFHR 2008)*, Montreal, Canada, 2008, pp. 664-669.

[5] H. Alamri, C. L. He, and C. Y. Suen, "A new approach for segmentation and recognition of Arabic handwritten touching numeral pairs," *Computer Analysis of Images and Patterns*, X. Jiang and N. Petkov, Eds.: CAIP 2009, vol. 5702, Berlin/Heidelberg, Germany: Springer-Verlag, 2009, pp. 165–172.

[6] S. Alma'adeed, D. Elliman, and C. A. Higgins, "A database for Arabic handwritten text recognition research," in *Proc. 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2002)*, Niagara-on-the-Lake, Canada, 2002, pp. 485–589.

[7] Plato, R. D. Archer-Hind, (ed. and tr.), *The Timaeus of Plato (1888)*, London, United Kingdom: McMillan & Co., 1888, Salem, USA: Ayers Co. Publishers, repr. 1988.

[8] G. R. Ball and S. N. Srihari, "Semi-supervised learning for handwriting

recognition", in *Proc. of 10th International Conference on Document Analysis and Recognition (ICDAR 2009)*, Barcelona, Spain, 2009, pp. 26-30.

[9] M. S. Bazarra, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms,* New York, USA: John Wiley & Sons, Inc., 1992.

[10] P. N. Belhumeur, J. Hespanda, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, 1997, pp. 711–720.

[11] R. Bertolami, M. Zimmermann, and H. Bunke, "Rejection strategies for offline handwritten text line recognition," *Pattern Recognition Letters*, vol. 27, no. 16, 2006, pp. 2005–2012.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning*, Berlin/Heidelberg, Germany: *Springer-Verlag*, 2006.

[13] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. of the 5th Annual ACM Workshop on Computational Learning Theory*, D. Haussler, Ed., Pittsburgh, USA: *ACM Press,* 1992, pp. 144 – 152.

[14] A. Brakensiek and G. Rigoll, "Handwritten address recognition using hidden Markov models," *Reading and Learning: adaptive content recognition*, vol. 2956, Berlin/Heidelberg, Germany: *Springer-Verlag*, 2004, pp. 103–122.

[15] A. 15, R. Sabourin, E. Lethelier, F. Bortolozzi, and C. Y. Suen, "Improvement handwritten numeral string recognition by slant normalization and contextual information," in *Proc. of 7th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2000)*, Amsterdam, the Netherlands, 2000, pp. 323 – 332.

[16] S. Carey and E. Bartlett, "Acquiring a single new word," in *Proc. of the Stanford Child Language Conference*, vol. 15, Stanford, USA, 1978, pp. 17-29. (Republished in Papers and Reports on Child Language Development, vol. 15, 1978, 17-29.)

[17] V. Castelli and T. Cover, "The exponential value of labeled samples," *Pattern Recognition Letters*, vol. 16, no. 1, 1995, pp. 105–111.

[18] C.-C. Chang and C.-J. Lin, LIBSVM: a library for support vector machines, 2001. [Online]. Available: LIBSVM – A library for Support Vector Machines, http://www.csie.ntu.edu.tw/~cjlin/libsvm. [Accessed: Sep. 3, 2007].

[19] H. Cecotti and A. Belaid, "Rejection strategy for convolutional neural network by adaptive topology applied to handwritten digits recognition," in *Proc. of the 8th International Conference on Document Analysis and Recognition (ICDAR 2005)*, Seoul, Korea, 2005, pp. 765–769.

[20] M. Cheriet, N. Kharma, C.-L. Liu, and C.Y. Suen, *Character Recognition Systems*, New York, USA: John Wiley & Sons, Inc., 2007.

[21] C. K. Chow, "An optimum character recognition system using decision functions," *IRE Trans. Electronics Computers*, vol. EC-6, no. 4, 1957, pp. 247 – 254.

[22] C.K. Chow, "On optimum recognition error and reject tradeoff," *IEEE Trans. Information Theory*, vol. 26, no. 1, 1970, pp. 40–46.

[23] S. Connell and A. K. Jain, "Writer adaptation for online handwriting recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, 2002, pp. 329 – 346.

[24] C. Cortes and V. N. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, 1995, pp. 273 – 297.

[25] O. Chapelle, B. Schoelkopf, and A. Zien, *Semi-Supervised Learning*, Cambridge, USA: MIT Press, 2006.

[26] W. R. Dillon and L. Schiffman, "Appropriateness of linear discriminant and multinomial classification analysis in marketing research," *Journal of Marketing Research*, vol. 15, 1978, pp. 103–112.

[27] J. X. Dong, A. Krzyzak, and C. Y. Suen, "A fast SVM training algorithm," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 3, 2003, pp. 367-384.

[28] J. X. Dong, A. Krzyzak, and C.Y. Suen, "Fast SVM training algorithm with decomposition on very large datasets," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, 2005, pp. 603–618.

[29] P. Dreuw, D. Rybach, C. Gollan, and H. Ney, "Writer adaptive training and writing variant model refinement for offline Arabic handwriting recognition," in *Proc. of 10th International Conference Document Analysis and Recognition (ICDAR 2009)*, Barcelona, Spain, 2009, pp. 21-25.

[30] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, 2nd ed., New York, USA: *John Wiley & Sons, Inc.*, 2000.

[31] C. Elkan, "The foundations of cost-sensitive learning," in *Proc. of 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)*, Seattle, USA, 2001, pp. 973-978.

[32] R.A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, 1936, pp. 179–188.

[33] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., New York, USA: John Wiley & Sons, Inc., 1987.

[34] J. Franklin, "The representation of context: Ideas from Artificial Intelligence," *Law, Probability and Risk,* vol. 2, no. 3, 2003, pp. 191-199.

[35] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed., Boston, USA: *Academic Press*, 1990.

[36] G. Fumera, F. Roli, and G. Vernazza, "A method for error rejection in multiple classifier systems," in *Proc. of the 11th International Conference on Image Analysis & Processing*, Palermo, Italy, 2001, pp. 454–458.

[37] G. Fumera and F. Roli, "Analysis of error-reject trade-off in linearly combined multiple classifiers," *Pattern Recognition*, vol. 37, no. 6, 2004, pp. 1245–1265.

[38] T.-F. Gao and C.-L. Liu, "High accuracy handwritten Chinese character recognition using LDA-based compound distances," *Pattern Recognition*, vol. 41, no. 11, 2008, pp. 3442–3451.

[39] N. Gorski, "Optimizing error-reject trade off in recognition systems," in *Proc. of the 4th International Conference on Document Analysis and Recognition (ICDAR 1997)*, vol. 2, Ulm, Germany, 1997, pp. 1092–1096.

[40] V. Gunes, M. Menard, P. Loonis, and S. Petit-Renaud, "Combination, cooperation and selection of classifiers: A state of the art," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 8, 2003, pp. 1303 – 1324.

[41] P. J. Haghighi, N. Nobile, C. L. He, and C. Y. Suen, "A new large-scale multi-purpose handwritten Farsi database," *Image Analysis and Recognition*, M. Kamel & A. Campilho Eds.: ICIAR 2009, vol. 5627, Berlin/Heidelberg, Germany: Springer-Verlag, 2009, pp. 278 – 286.

[42] C. L. He, P. Zhang, J. X. Dong, C. Y. Suen, and T. D. Bui, "The role of size normalization on the recognition rate of handwritten numerals," in *Proc. of IAPR TC3 Workshop of 8th International Conference on Document Analysis and Recognition (ICDAR 2005): Neural Networks and Learning in Document Analysis and Recognition*, Seoul, Korea, 2005, pp.8 – 12.

[43] C. L. He and C. Y. Suen, "A hybrid multiple classifier system of unconstrained handwritten numeral recognition," *Pattern Recognition and Image Analysis*, vol. 17, No. 4, 2007, pp. 608-611.

[44] C. L. He, L. Lam, and C. Y. Suen, "A novel rejection measurement in handwritten numeral Recognition Based on Linear Discriminant Analysis," in *Proc. 10th International Conference on Document Analysis and Recognition (ICDAR 2009)*, Barcelona, Spain, 2009, pp. 451-455.

[45] C. L. He, L. Lam, and C. Y. Suen, "Optimization of rejection parameters to enhance reliability in handwriting recognition," *Handbook of Pattern*

*Recognition and Computer Vision*, vol. 4, Chapter 3.4, C. H. Chen ed., 2010, pp. 377 – 395.

[46] C. L. He, L. Lam, and C. Y. Suen, "Automatic discrimination between confusing classes with writing styles verification in Arabic handwritten numeral recognition," in *Proc. of 20th International Conference on Pattern Recognition (ICPR 2010)*, Istanbul, Turkey, 2010, to be published.

[47] C. L. He and C. Y. Suen, "Error reduction based on error categorization in Arabic handwritten numeral recognition," in *Proc. of 15th International Conference on Frontiers in Handwriting Recognition (ICFHR 2010)*, Kolkata, India, 2010, to be published.

[48] R. Healey, "Database Management Systems," in *Geographic Information Systems: Principles and Applications*, D. Maguire, M. F. Goodchild, and D. Rhind eds., London, England: Longman Scientific & Technical, 1991.

[49] J. Holt, *How Children Learn,* New York, USA: Delta/Seymour Lawrence, 1983.

[50] Z. Huang, K. Ding, L. Jin, and X. Gao, "Writer adaptive online handwriting recognition using Incremental Linear Discriminant Analysis," in *Proc. of 10th International Conference on Document Analysis and Recognition (ICDAR 2009)*, Barcelona, Spain, 2009, pp. 91 – 95.

[51] R. Johnson and T. Zhang, "On the effectiveness of laplacian normalization for graph semi-supervised learning," *Journal of Machine Learning Research,* vol. 8, 2007, pp. 1489–1517.

[52] E. Kavallieratou, N. Fakotakis, and G. Kokkinakis, "Slant estimation algorithm for OCR systems," *Pattern Recognition*, vol. 34, no. 12, 2001, pp. 2515 – 2522.

[53] K. Koutroumbas and S. Theodoridis, *Pattern Recognition*, 4th ed., Boston, USA: Academic Press, 2008.

[54] A. L. Koerich, "Rejection strategies for handwritten word recognition," in *Proc. of the 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2004)*, Tokyo, Japan, 2004, pp. 479–484.

[55] H. Kuhn and A. Tucker, "Nonlinear programming," in *Proc. of 2nd Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, USA: University of California Press, 1951, pp. 481 – 492.

[56] L. Lam, "Classifier combinations: Implementations and theoretical issues," *Multiple Classifier Systems, 1st International Workshop (MCS 2000),* Cagliari, Italy, 2000, pp. 77–86.

[57] F. Lauer, C. Y. Suen, and G. Bloch, "A trainable feature extractor for handwritten digit recognition," *Pattern Recognition*, vol. 40, no. 6, 2007, pp.

1816 – 1824.

[58] Y. LeCun and C. Cortes, The MNIST Database of handwritten digits, 1998. [Online]. Available: MNIST handwritten digit database, http://yann.lecun.com/exdb/mnist/. [Accessed: Apr. 8, 2006]

[59] Lin R. S., M. H. Yang, and S. E. Levinson, "Object tracking using incremental Fisher discriminant analysis," in *Proc. of 17th International Conference on Pattern Recognition (ICPR 2004)*, Cambridge, UK, 2004, pp. 23–26.

[60] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: investigation of normalization and feature extraction techniques," *Pattern Recognition*, vol. 37, no. 2, 2004, pp. 265-279.

[61] C.-L. Liu and C. Y. Suen, "A new benchmark on the recognition of handwritten Bangla and Farsi numeral characters," *Pattern Recognition*, vol. 42, no. 12, 2009, pp. 3287-3295.

[62] N. Lund, *Attention and Pattern Recognition*, East Sussex, UK: Routledge, 2001.

[63] B. Maeireizo, D. Litman, and R. Hwa, "Co-training for predicting emotions with spoken dialogue data," in *The Companion Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL),* no. 28, Barcelona, Spain, 2004.

[64] K. J. Malmberg, "Recognition memory: A review of the critical findings and an integrated theory for relating them," *Cognitive Psychology*, vol. 57, no. 4, 2008, pp. 335-384.

[65] D. Mareschal and P. C. Quinn, "Categorization in infancy," *Trends in Cognitive Sciences*, vol. 5, no. 10, 2001, pp. 443-450.

[66] J. Mercer, "Functions of positive and negative type and their connection with the theory of integral equations," in *Philos. Trans. Roy. Soc. London,* vol. A, no. 209, 1909, pp. 415 – 446.

[67] A. Nosary, L. Heutte, and T. Paquet, "Unsupervised writer adaption applied to handwritten text recognition," *Pattern Recognition*, vol. 37, no. 2, 2004, pp. 385 – 388.

[68] J. J. de Oliveira Jr., L. R. Veloso, J. M. de Carvalho, "Interpolation/decimation scheme applied to size normalization of characters images," in *Proc. of the 15th International Conference Pattern Recognition (ICPR 2000)*, Barcelona, Spain, vol. 2, 2000, pp. 577 – 580.

[69] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 9, no. 1, 1979, pp. 62–66.

[70] U. Pal, T. Wakabayashi, and F. Kimura, "Comparative study of Devangari handwritten character recognition using different feature and classifiers," in *Proc. of the 10th International Conference on Document Analysis and Recognition (ICDAR 2009)*, Barcelona, Spain, 2009, pp. 1111-1115.

[71] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, no. 6, 1901, 559–572.

[72] C. A. Perez, C. M. Held, and P. R. Mollinger, "Handwritten digit recognition based on prototypes created by Euclidean distance," in *Proc. of the 1999 International Conference on Information Intelligence and Systems*, Bethesda, USA, 1999, pp. 320 – 323.

[73] M. Pechwitz, S. S. Maddouri, V. Margner, N. Ellouze, and H. Amiri,, "IFN/ENIT - database of handwritten Arabic words," in *Proc. of 7th Colloque International Francophone sur l'Ecrit et le Document (CIFED 2002)*, Hammamet, Tunisia, 2002, pp. 129–136.

[74] J. F. Pitrelli, and M. P. Perrone, "Confidence-scoring post-processing for off-line handwritten-character recognition verification," in *Proc. of 7th International Conference on Document Analysis and Recognition (ICDAR 2003)*, vol. I, Edinburgh, Scotland, 2003, pp. 278–282.

[75] F. R. Rahman and M. C. Fairhurst, "A new hybrid approach in combining multiple experts to recognise handwritten numerals," *Pattern Recognition Letters*, vol. 18, no. 8, 1997, pp. 781 -790.

[76] A. V. Reed, "Error-correcting strategies and human interaction with computer systems," in *Proc. of the 1982 conference on Human factors in computing systems*, Gaithersburg, USA, 1982, pp. 236 – 238.

[77] E. Riloff, J. Wiebe, and T. Wilson, "Learning subjective nouns using extraction pattern bootstrapping," in *Proc. of the 7th Conference on Natural Language Learning (CoNLL-2003)*, vol. 4, Edmonton, Canada, 2003, pp. 25-32.

[78] A. El. Sagheer, N. Tsuruta, and R.-I. Taniguchi, "Arabic lip-reading system: A combination of Hypercolumn Neural Network Model with Hidden Markov Model," *Artificial Intelligence and Soft Computing (ASC 2004)*, Marbella, Spain, 2004, 9.1–9.3.

[79] M. W. Sagheer, C. L. He, N. Nobile, and C. Y. Suen, "A new large Urdu database for off-Line handwriting recognition," *Image Analysis and Processing*, P. Foggia, C. Sansone, and M. Vento, Eds.: ICIAP 2009, vol. 5716, Berlin/Heidelberg, Germany: Springer-Verlag, 2009, pp. 538–546.

[80] M. W. Sagheer, C. L. He, N. Nobile, and C. Y. Suen, "Holistic Urdu handwritten word recognition using Support Vector Machine," in *Proc. of 20th International Conference on Pattern Recognition (ICPR 2010)*, Istanbul,

Turkey, 2010, to be published.

[81] M. W. Sagheer, N. Nobile, C. L. He, and C. Y. Suen, "A novel handwritten Urdu word spotting based on connected components analysis," in *Proc. of 20th International Conference on Pattern Recognition (ICPR 2010)*, Istanbul, Turkey, 2010, to be published.

[82] B. Scholkopf, S. Mika, C. Burges, P. Knirsch, K.-R. Muller, G. Ratsch, and A. J. Smola, "Input space vs. feature space in kernel-based methods," *IEEE Trans. Neural Networks*, vol. 10, no. 5, 1999, pp. 1000-1017.

[83] P. G. Schyns, "Diagnostic recognition: task constraints, object information, and their interactions," *Cognition*, vol. 67, no. 1, 1998, pp. 147-179.

[84] H. J. Scudder, "Probability of error of some adaptive pattern-recognition machines," *IEEE Trans. Information Theory*, vol. 11, no. 3, 1965, pp. 363–371.

[85] A. Senior and K. Nathan, "Writer adaption of HMM handwriting recognition system," in *Proc. of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1997)*, vol. 2, Washington D.C., USA, 1997, pp. 1447 – 1450.

[86] M. I. Shah, J. Sadri, C. Y. Suen, and N. Nobile, "A new multipurpose comprehensive database for handwritten Dari recognition," in *Proc. of 11th International Conference on Frontiers in Handwriting Recognition (ICFHR 2008)*, Montreal, Canada, 2008, pp. 635-640.

[87] M. I. Shah, C. L. He, N. Nobile, and C. Y. Suen, "The first multi-purpose handwritten database for Pashto handwritten recognition research," in *Proc. of 14th Conference of the International Graphonomics Society*, Dijon, France, 2009, pp. 157-161.

[88] S. Shekhar, P. Zhang, Y. Huang, and R. R. Vatsavai, "Trends in Spatial Data Mining," *Data Mining: Next Generation Challenges and Future Directions*, H. Kargupta and A. Joshi, K. Sivakumar, and Y. Yesha, Eds. Cambridge, USA: AAAI/MIT Press, 2003.

[89] M. Shi, Y. Fujisawa, T. Wakabayashi, and F. Kimura, "Handwritten numeral recognition using gradient and curvature of gray scale image," *Pattern Recognition*, vol. 35, no. 10, 2002, pp. 2051 – 2059.

[90] F. Solimanpour, J. Sadri, and C. Y. Suen, "Standard databases for recognition of handwritten digits, numerical strings, legal amounts, letters and dates in Farsi Language," in *Proc. of the 10th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2006)*, La Baule, France, 2006, pp. 3–7.

[91] G. Srikantan, D.-S. Lee, and J. T. Favata, "Comparison of normalization methods for character recognition," in *Proc. of 3rd International Conference*

*Document Analysis and Recognition (ICDAR 1995)*, Montreal, Canada, 1995, pp. 719 – 722.

[92] J. Subranhmonia, K. Nathan, and M. Perrone, "Writer dependent recognition of on-line unconstrained handwriting," in *Proc. of the 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP1996)*, vol. 6, Atlanta, USA, 1996, pp. 34788 – 3481.

[93] C. Y. Suen, C. Nadal, R. Legault, T. A. Mai, and L. Lam, "Computer recognition of unconstrained handwritten numerals," in *Proc. IEEE*, vol. 80, no. 7, 1992, pp. 1162–1180.

[94] C. Y. Suen and J. Tan, "Analysis of errors of handwritten digits made by a multitude of classifiers," *Pattern Recognition Letters*, vol. 26, no. 3, 2005, pp. 369-379.

[95] D. L. Swets and J. Weng, "Hierarchical discriminant analysis for image retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, 1999, pp. 386–401.

[96] F. Tang and H. Tao, "Fast linear discriminant analysis using binary bases," *Pattern Recognition Letters*, vol. 28, no.16, 2007, pp. 2209–2218.

[97] N. S. Thompson and F. Tonneau (Eds), "Perspectives in Ethology: Evolution, culture, and behavior", in *Series: Perspectives in Ethology*, vol. 13, New York, USA: Kluwer Academic/Plenum Publishers, 2001.

[98] P. D. Turney, "Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm," *Journal of Artificial Intelligence Research*, vol. 2, 1995, pp. 369-409.

[99] P. D. Turney, "Types of cost in inductive concept learning," in *Proc. of Workshop on Cost-Sensitive Learning at the 7th International Conference on Machine Learning (WCSL at ICML-2000),* California, USA, 2000, pp. 15-21.

[100] N. G. Ushakov, "Unimodal distribution," in *Encyclopaedia of Mathematics*, Michiel Hazewinkel Ed., Berlin/Heidelberg, Germany: Springer-Verlag, 2001.

[101] V. Vapnik and A. Lerner, "Pattern recognition using generalized portrait method," *Automation and Remote Control*, vol. 24, 1963, pp. 774–780.

[102] V. Vapnik, *The Natural of Statistical Learning Theory*, Berlin/Heidelberg, Germany: Springer-Verlag, 1995.

[103] V. Vapnik, *Statistical Learning Theory*, New York, USA: John Wiley & Sons, Inc., 1998.

[104] A. Vinciarelli and S. Bengio, "Writer adaption techniques in HMM based off-line cursive script recognition," *Pattern Recognition Letters*, vol. 23, no. 8,

2002, pp. 905 – 916.

[105] V. Vuori, "Clustering writing styles with a self-organizing map," in *Proc.of the 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2002)*, Niagara-on-the-Lake, Canada, 2002, pp. 345-350.

[106] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research*, vol. 5, 2004, pp. 975–1005.

[107] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proc. of the 33rd Annual Meeting of the Association for Computational Linguistics*, Cambridge, USA, 1995, pp. 189–196.

[108] P. Zhang, T. D. Bui, and C. Y. Suen, "A novel cascade ensemble classifier system with a high recognition performance on handwritten digits," *Pattern Recognition*, vol. 40, no. 12, 2007, pp. 3415–3429.

[109] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, 1984, pp. 236 – 239.

[110] B. Zhao, Y. Liu, and S. W. Xia, "Support vector machines and its application in handwritten numeral recognition," in *Proc. of 15th International Conference on Pattern Recognition (ICPR 2000)*, Barcelona, Spain, vol. 2, 2000, pp. 720 – 723.

[111] X. Zhu and A. Goldberg, *Introduction to Semi-Supervised Learning*, San Rafael, USA: Morgan & Claypool Publishers, 2009.