2004

Parametric classification in domains of characters, numerals, punctuation, typefaces and image qualities

Osama Ahmed Khan
University of Texas-Pan American

PARAMETRIC CLASSIFICATION IN DOMAINS OF CHARACTERS, NUMERALS, PUNCTUATION, TYPEFACES AND IMAGE QUALITIES

A Thesis

by

OSAMA AHMED KHAN

Submitted to the Graduate School of the
University of Texas-Pan American
In partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

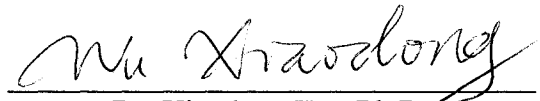
December 2004

Major Subject: Computer Science

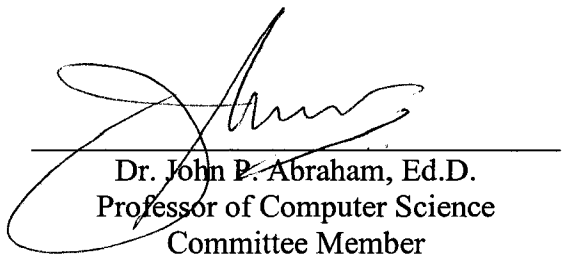
PARAMETRIC CLASSIFICATION IN DOMAINS OF CHARACTERS, NUMERALS, PUNCTUATION, TYPEFACES AND IMAGE QUALITIES

A Thesis
by
OSAMA AHMED KHAN

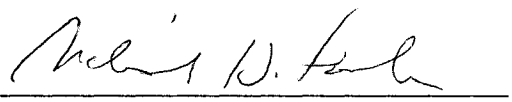
Approved as to style and content by:



Dr. Xiaodong Wu, Ph.D.
Assistant Professor of Computer Science
Chair of Committee



Dr. John P. Abraham, Ed.D.
Professor of Computer Science
Committee Member



Dr. Richard H. Fowler, Ph.D.
Professor of Computer Science
Committee Member

December 2004

ABSTRACT

Khan, Osama A., Parametric Classification In Domains Of Characters, Numerals, Punctuation, Typefaces And Image Qualities. Master of Science (CS), December, 2004, 165pp., 10 tables, 33 illustrations, references, 83 titles.

This thesis contributes to the Optical Font Recognition problem (OFR), by developing a classifier system to differentiate ten typefaces using a single English character 'e'. First, features which need to be used in the classifier system are carefully selected after a thorough typographical study of global font features and previous related experiments. These features have been modeled by multivariate normal laws in order to use parameter estimation in learning. Then, the classifier system is built up on six independent schemes, each performing typeface classification using a different method. The results have shown a remarkable performance in the field of font recognition. Finally, the classifiers have been implemented on Lowercase characters, Uppercase characters, Digits, Punctuation and also on Degraded Images.

ACKNOWLEDGMENTS

It is difficult to overstate my gratitude to my Thesis Advisor, Dr. Xiaodong Wu. With his enthusiasm, his inspiration, and his great efforts to explain things clearly and simply, he helped to make Optical Font Recognition fun for me. Throughout my thesis-writing period, he provided encouragement, sound advice, good teaching, good company, and lots of good ideas. I would have been lost without him.

Dr. John P. Abraham guided me throughout my stay here at UTPA and made me work so hard on my Thesis. He filled me with the inspiration of hard work through his real-life stories. I owe him a lot for the financial support he provided me throughout my MS program, and for the special care he took for all of us, whenever we were in need.

I had a great time with my graduate advisor, Dr. Richard H. Fowler, who was so kind and cooperative at every moment, I needed his help. I appreciate his guidance in exploring thesis projects, deciding on research goals, and revising this thesis. He has made the research group a great place to work. It was really fun taking classes, offered by him.

Lastly, and most importantly, I wish to thank my parents, who raised me, supported me, taught me, and loved me. To them I dedicate this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

It is said that writing was developed to satisfy a very strong desire, i.e. to communicate and to broadcast information in an accessible and storable form. The development of various printing technologies has increased the volume of documents enormously, and hence the need for a better organization of company documentation.

1.1 The document domain

The perception of document has advanced over recent years to incorporate all characteristics of written communication present in paper and electronic forms. The revolution in the information broadcasting domain has defined new necessities not only for reading remote documents, but also for reprocessing information they contain. These requirements presume strong mechanisms for information storage and retrieval allowing a fast and precise access to this information.

Nowadays, the majority of documents are saved, circulated, and presented on paper form, which constitutes the primary medium for books, manuals, newspapers, magazines, and business correspondence. The diminishing cost and escalating performance of hardware

All the figures in this chapter were taken from [Zra95] with the permission of the author

will, however, make the storage and delivery of documents by electronic means the predominate medium [WCW82].

Document recognition systems are required to incorporate current paper documents and such systems could help users in encoding printed documents for computer processing. Although the recovery of stored documents in electronic forms could be easily executed, reading printed documents still poses serious problems.

1.1.1 Document structures

A document may appear in various forms [Jol89, Qui89] but one is especially interested in its robust representations, which can be stored on devices or broadcasted through communication networks. Two representations of particular interest which are commonly addressed are the 'logical structure' and the 'physical structure' [Fur89, Hu94]:

- Logical structure: this identifies the document structure and content. It symbolizes the author's point of view of the document. For example, a book can be viewed as a hierarchy of logical entities (e.g. chapter, section, subsection, paragraph, etc.).
- Physical structure: this depicts the formatted form of the document that is to be transmitted to an output device. It characterizes the typographer's or typesetter's point of view of the document.

While the logical structure takes the document's revisable form, the physical structure deals with its rendable form.

1.1.2 Document manipulations

During its life, a document goes through a cycle of several steps (see Figure 1.1). From its logical view to its paper form, the document navigates through the classical stages of document production:

- The document is envisaged by its creator (author) and comes into life through publishing services. At this stage, the document is viewed by its logical structure;
- After formatting, the document is represented by its physical structure;
- The physical document is finally delivered, i.e. printed on paper or exhibited on screen;
- A reader can now capture the document.

The reverse path that allows converting the document from its paper form to the equivalent logical structure, is executed by 'Document Recognition' systems:

- The document is first scanned to generate images;
- The images are processed to produce the physical document using a document analysis process;
- The physical document is then transformed into its logical correspondent through a document understanding process.

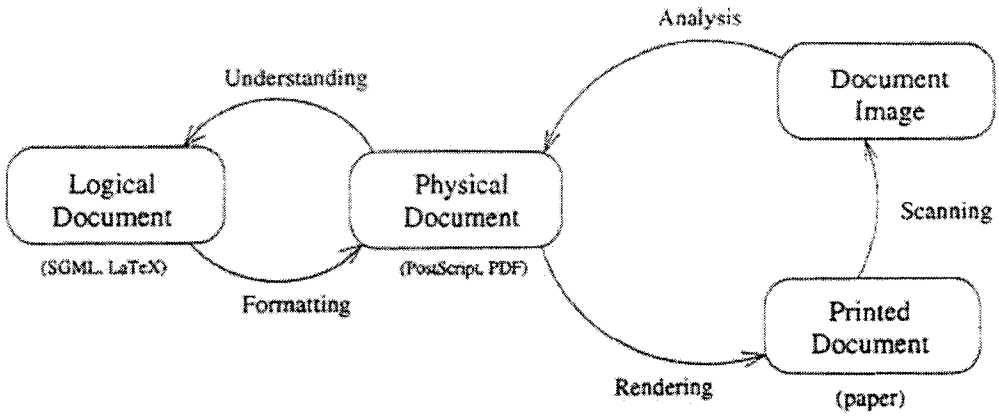


Figure 1.1. Possible manipulations applicable on documents.

1.2 Document structure analysis

'Document Structure Analysis' systems (DSAs) aim to transform into a computer-reusable form data presented on paper and intended for human understanding. The analysis includes the examination of the physical document which is stored in binary images to acquire the logical structure that portrays its content. During the last two decades, many prototype systems and business tools have undertaken the goal of DSA [WCW82, WS88, Den90, TA90, TSYC91, SLG+92, Che93, Hu94, TCB94, TCB95].

A DSA system needs to interact with diverse document layouts. Two techniques are fundamental to accomplish that goal: 'document analysis' and 'document understanding' [TA90, TSYC91, Doe93], which should be both implemented by a DSA system:

- 'Document analysis' is employed to extract the document physical structure. It breaks down the document image into several blocks, indicating coherent

components, such as text lines, headlines, graphics, etc. The extraction may be done by taking into account the document generic layout (e.g. a scientific journal layout) [Azo95].

- 'Document understanding' is used to extract logical relationships between the extracted blocks. Normally, it maps the physical structure into a logical structure taking into consideration the logical relationships between the objects in a particular document.

The two techniques require the detection of the characters that appear in the document and their typographical features (font, margins, line spacing, etc.). Optical Character Recognition (OCR) and Optical Font Recognition (OFR) are the two significant tasks in DSA systems.

While OCR has been dealt with both commercially and academically for long, the OFR problem seems having been comparatively ignored. This thesis addresses OFR and presents diverse classification methods implemented to satisfy its requirements.

1.3 Overview of results

To deal with the area of font recognition, we built up six classifiers, each based on a different scheme, and then implemented them successfully. The results showed a remarkable difference in performance of the six classifiers. Starting from Classifier 1, where we used the basic method to address the font recognition problem, we improved

the scheme every time with the change of classifier, until we finished building up the Classifier 6. From the results discussed in detail in chapter 6, it can be observed that while getting an average error rate of approximately 70% on the first two classifiers, we were able to reduce it down to approximately 7%, i.e. 10 times reduction in the error rate. The results have proved that the more the features are taken into consideration, the greater the classification performance is. Also, dealing with binary features, nearly optimal performance can be achieved. The future work includes tweaking the Classifier 6 architecture in order to boost up the performance level and achieve the best performance ever observed.

1.4 Organization of the thesis

The thesis is organized in six chapters and two appendices. It can be viewed as composed of four parts each addressing a particular issue. The first part (Chapters 2 and 3) summarize the OCR field, which has dominated scientific and commercial efforts for a long time, concentrates on the OFR problem, and sets the structure of the classifier system.

The second part (Chapter 4) outlines the field of Typeface Discrimination, especially from the font recognition point of view. It begins with an explanation of vital typographical concepts and then presents some elements allowing discriminating between typefaces and fonts. A survey of the typeface classification problem is also provided.

The third part (Chapter 5) presents the 'Bayesian Decision Theory'. We begin by considering the ideal case in which the probability structure underlying the categories is known perfectly. While this sort of situation rarely occurs in practice, it permits us to determine the optimal (Bayes) classifier against which we can compare all other classifiers. Moreover, in some problems it enables us to predict the error we will get when we generalize to novel patterns.

In the last part (Chapter 6), a first assessment of the system performances on a set of 10 fonts is then presented. The evaluation has been applied on the font models in order to learn the relevance of features in font discrimination and to estimate the theoretical performance of the classifiers. Several classification trials have also been designed in order to assess the approach practically.

Two appendices are presented at the end of the thesis. Appendix A presents some complements on typeface products and some related statistics. Appendix B shows samples of texts written with the measured fonts.

CHAPTER 2

OPTICAL CHARACTER RECOGNITION

Optical Character Recognition (OCR) concentrates on the retrieval of characters from document images. It has been into existence for more than forty years. The first products came for commercial use in the form of highly specialized machines. With the production of personal computers, OCR became more accepted, leading to a considerable enhancement in performance and a severe decrease in costs. The expensive hardware solutions have been substituted by more economical software tools.

Several methods have been anticipated to distinguish printed characters. Some of them model characters by their topological structure, others deal with character templates directly. The current OCR technology performs not only fundamental character recognition, but also segmentation and grammatical substantiation.

OCR systems can be categorized into three classes according to the font handling capabilities. While some of them disregard the text font, others involve its 'a priori' knowledge. The assessment of OCR solutions can be based on diverse criteria.

All the figures in this chapter were taken from [Zra95] with the permission of the author

After a definition of OCR in Section 2.1, the main techniques used to recognize characters are presented in Section 2.2. Section 2.4 tries to make a classification of OCR systems. Section 2.4 describes a conventional architecture of OCR systems. Section 2.5 fixes the framework of the OCR evaluation and emphasizes the limits of existing solutions.

2.1 Definition

Pattern recognition involves the assignment of a 'pattern' to one of 'n classes'. The patterns and classes are generally symbolized by vectors of measured features. The classes are stored in a collection of pre-classified vectors. In the OCR context, a class could signify one letter of the alphabet.

Let $C = \{c_1, c_2, \ldots, c_n\}$ be the character set. The aim of any OCR system is to allocate a character c_i to a candidate pattern. The system extracts a feature vector from the pattern and then passes through the set C in order to find a character c_i that offers the best 'matching' with the pattern vector. The set C is often produced from a training set of samples of each character.

Complete surveys of OCR related problems can be found in [Nag82, Pav93, Bai93, RKN93, RKN94]. Scanning, segmentation, normalization, classification as well as other OCR related issues are extensively presented in these papers.

2.2 OCR methods

The methods used to carry out OCR comprise of adaptations of pattern recognition and classical signal processing technologies. The next subsections outline the most common means of character recognition.

2.2.1 Template matching methods

Template matching was one of the foremost methods used by the OCR devices. A collection of character templates is preserved and used to identify patterns. The identification consists of finding the 'closest' matching template. Closest matching is accomplished by the computation of the minimum distance between the pattern and each class template. The distance is defined as the symmetrical difference between two templates. Other distances have also been described such as the 'Stochastic distances' [Ing89].

Many alternatives of the template matching method have been used. They are simple to design but remain heavily susceptible to the image quality (skew, noise, etc.). They also require templates for each font and presume the 'a priori' knowledge of the font used in the document.

A specific template matching method has been effectively used by Ingold [Ing89]. He came out with a special definition of templates, which considers the patter distortions. In

such a process, each character is modeled by a skeleton (obligatory pixels) and an envelope (optional pixels).

2.2.2 Structural methods

Structural methods are based on the topological structure of the character. They represent characters by structural features and their relationships. Features state the topological properties of the character such as loops, arcs, etc. A relationship could be, for example, the relative position of a feature related to another one.

Structural methods can be grouped in three categories:

- 'String matching methods' where characters are represented by feature strings. The comparison between the pattern and a class consists of computing the correlation between the corresponding strings by a distance metric. Many methods based on string matching techniques have been presented in [BB92].
- 'Syntactic methods' where each character is characterized by a sequence of features (phrase). The features compile the vocabulary of a given language. The recognition of a given pattern consists of deciding whether the phrase (describing the pattern) could be generated by the language grammar. Syntactic methods have been used by Ramesh [Ram89] and Baptista [BK88] to attend to the OCR problem.
- 'Graph-based methods' consist of a graph construction where nodes include features. The relationship between the features is represented by arcs. The

recognition consists of toning the constructed graph with other graphs representing the reference characters.

Several graph-based methods have been proposed. That of Pavlidis [Pav86], called 'Line Adjacency Graph', thins the character and concurrently labels the generated graph. Those of Baird, Kahan and Lebourgeois, which attained good results, are based on similar principles [Bai86, KPB87, Leb91].

2.2.3 Statistical methods

In statistical pattern recognition, which is based on the statistical decision theory, each pattern is measured as a single entity and is characterized by a finite dimensional vector of pattern features. Bayesian, stochastic, nearest-neighbor classifications and neural networks are four fundamental statistical methods:

- Bayesian classification: the Bayesian decision theory is based on the hypothesis that the decision is stated in probabilistic terms and that all of the relevant probability values are known beforehand [DH73]. Let $C = \{c_1, c_2, \ldots, c_n\}$ be the set of characters. The classification process consists of connecting a character c_i to a pattern defined by a feature vector 'x', so that the conditional probability of c_i given 'x', $P(c_i \mid x)$ is maximal. $P(c_i \mid x)$ is computed from the Bayes rule,

$$P(c_i \mid x) = \frac{p(x \mid c_i)\, P(c_i)}{P(x)}$$

where,

$$P(x) = \sum_{i=1}^{n} p(x \mid c_i)\, P(c_i)$$

$P(c_i)$ expresses the 'a priori probability' of the character c_i and $p(x \mid c_i)$ represents the 'conditional density function' of 'x', i.e. the probability of obtaining 'x' when its class is c_i. The discriminant function can have several forms, for e.g.:

$$D_i(x) = p(x \mid c_i)$$

In the OCR context, the described method has been used by Baird and Campigli to achieve multi-font character recognition [BF91, CCP91].

- Nearest neighbor classification: as compared to Bayesian methods, nearest neighbor classification methods, denoted k-NN, are non-parametric since they utilize all of the class training samples as prototypes for the class. These methods have been the focus of decades of research [Das88]. The 1-NN classification of an unknown vector 'x' is basically the class of the nearest prototype. The discriminant function has the form [BCG+93],

$$D_i(x) = - \min_{1 \le j \le M_i} d^2(x, x_j^{(i)})$$

where $x_j^{(i)}$ designates the feature vector from j^{th} sample of class 'i', and 'M_i' the number of training samples of class 'i'.

When $k > 1$, voting between the 'k' nearest neighbors is employed and the majority class wins. It is valuable, when the single nearest neighbor may belong to the wrong class but the majority are not.

- Stochastic classification: a handwritten character can be modeled by a stochastic process by replicating the progression of a pen writing the character. This process is time dependent. For printed characters, the temporal part is lost, but the pen movement can be easily imitated by contour tracing.

Thus, in the stochastic approach, a character is measured as a continuous signal discernible in time at different points representing some observation states. In this modeling approach, the states are depicted by transition probabilities (from one state to another) and by each one (of the states) observation probability. The classification consists of locating in the state graph the most likely path analogous to the series of observed elements within the input string.

Stochastic models have been used by Anigbogu to identify multi-font texts [Ani92]. A method based on pseudo two-dimensional hidden Markov models was also proposed by Agazzi [AK93]. It merges and optimizes character recognition and image normalization.

- Artificial neural network classification: over the last two decades, there have been considerable efforts to expand models of neural networks. They were devoted to unravel knowledge and recognition problems in a sensible time.

Two striking features of artificial neural networks are learning and generalization from training sets. The power of the model exists in the network architectures and their capacity to perform autonomous learning.

Several neural network approaches have been projected and applied in the OCR context. A neural network for an OCR system that performs well on noisy, handwritten characters, was proposed by Le Cun et al. [Cun89]. Avi-Itzhak [AIDG95] developed a multi-layered neural network approach to achieve high

accuracy recognition on multi-size and multi-font characters. He exploited a centroid-dithering training process with a low noise sensitivity normalization system. Gosselin [Gos91] applied a Boolean neural network to distinguish characters and achieved a 99.8% accuracy. D'Acierno [DSV91] applied a completely connected feed-forward neural network to perform the recognition of multi-font printed characters.

Blue et al. made an exciting comparative study of classification accuracy of four statistical and three neural networks classifiers for two image based pattern classification problems, namely fingerprint and isolated handprint digits recognition [BCG+93]. The classifiers were tested on the NIST (National Institute of Standards and Technology) databases for learning and classification. For the evaluated datasets, the best accuracy for both applications was provided by a probabilistic neural network.

2.3 Taxonomy of OCR systems

OCR systems can be classified in three categories according to their capability in font handling.

2.3.1 Mono-font OCR systems

Mono-font OCR systems identify characters of an 'a priori' known font. They have been neglected because they need a precise learning for each font, and are susceptible to the

image quality (noise, skew, etc.). An experimental mono-font OCR package, devised by Ingold [Ing89], achieved good recognition rates by means of a template matching method. Each character class has been modeled by its skeleton and envelope.

2.3.2 Multi-font OCR systems

They permit the recognition of characters from several already learned fonts. The recognition is usually preceded by a size and weight normalization stage. The normalization, which prevails over small deformations, helps OCR systems boost their qualitative performance.

'ExperVision', which uses template matching techniques, is one of the odd commercial multi-font systems. It uses templates trained from 30 samples of each character extracted from different documents [Way93].

2.3.3 Omni-font OCR systems

'Omni-font' OCR systems make generalization of the font information since they intend to distinguish characters of any font and size. They use structural methods, which do not require learning or at worst a limited learning of special characters. Furthermore, they need a contextual knowledge to differentiate between indistinct cases. For example, in order to distinguish a 'P' from a 'p', the character position in the word is required. The

distinction between '0' and 'O' or between 'l' and '1' also relies on contextual analysis. The question arises whether the word's characters are numerals or alphabetic letters.

Most of the available OCR systems claim to be 'omni-font'. The term 'omni-font' used commercially as a marketing case is not suitable for current OCR solutions since none of them can recognize the 3000 fonts accessible in the market. The more practical term poly-font was used by Baird [BF91] to denote OCR systems recognizing a large number of fonts.

2.4 OCR system architecture

Experimental and commercial OCR systems often use fusions of classification and analysis methods. They may merge structural methods with statistical ones, e.g. structural shape analysis with Bayesian statistical classification.

In practice, they do more than intrinsic isolated character recognition. Figure 2.1 shows the structural design of a typical OCR system. The document traverses five processes before being delivered to the user.

1. 'Scanning' constitutes a opening process that transforms the document from its paper form to an image form. It is generally included in the OCR software where scanning parameters (resolution, thresholds) are put to default values, or explicitly provided by users according to the document quality.

2. 'Pre-processing' is applied to the image to clean the character patterns. The objective of the pre-processing is to organize the image for the segmentation and recognition processes. It usually consists of applying a succession of image processing techniques (e.g. noise removal, skew detection and correction, character contour smoothing or thinning, etc.).

   These techniques can be applied on the whole image or on a single pattern. They may therefore be executed before and/or after segmentation and are often implicitly applied.

3. 'Segmentation' allows the extraction and locality of each character in the image. Several segmentation schemes that are based on top-down, bottom-up or mixed approaches are presented in [Azo95].

   It characterizes a crucial problem for OCR since recognition algorithms often presume isolated patterns. The segmentation overpowers the OCR performance since a segmentation fault leads automatically to several OCR errors. The segmentation of touching characters continues to pose grave problems [LSA89, Lu93].

4. 'Character recognition' signifies the main process in the system since it allocates a character class to the pattern. It uses a library which is often created by learning from training sets including models of all characters. The character modeling and classification approaches rely totally on the OCR method used.

5. 'Post-processing' is used to improve the character recognition, especially to correct spelling. It is often based on linguistic dictionaries, n-grams techniques, typographical context analysis, etc. [Ani92, Sen94].

Within commercial OCR systems, processes 2 to 5 are often transparent to users and the segmentation and classification methods used are often kept secret.

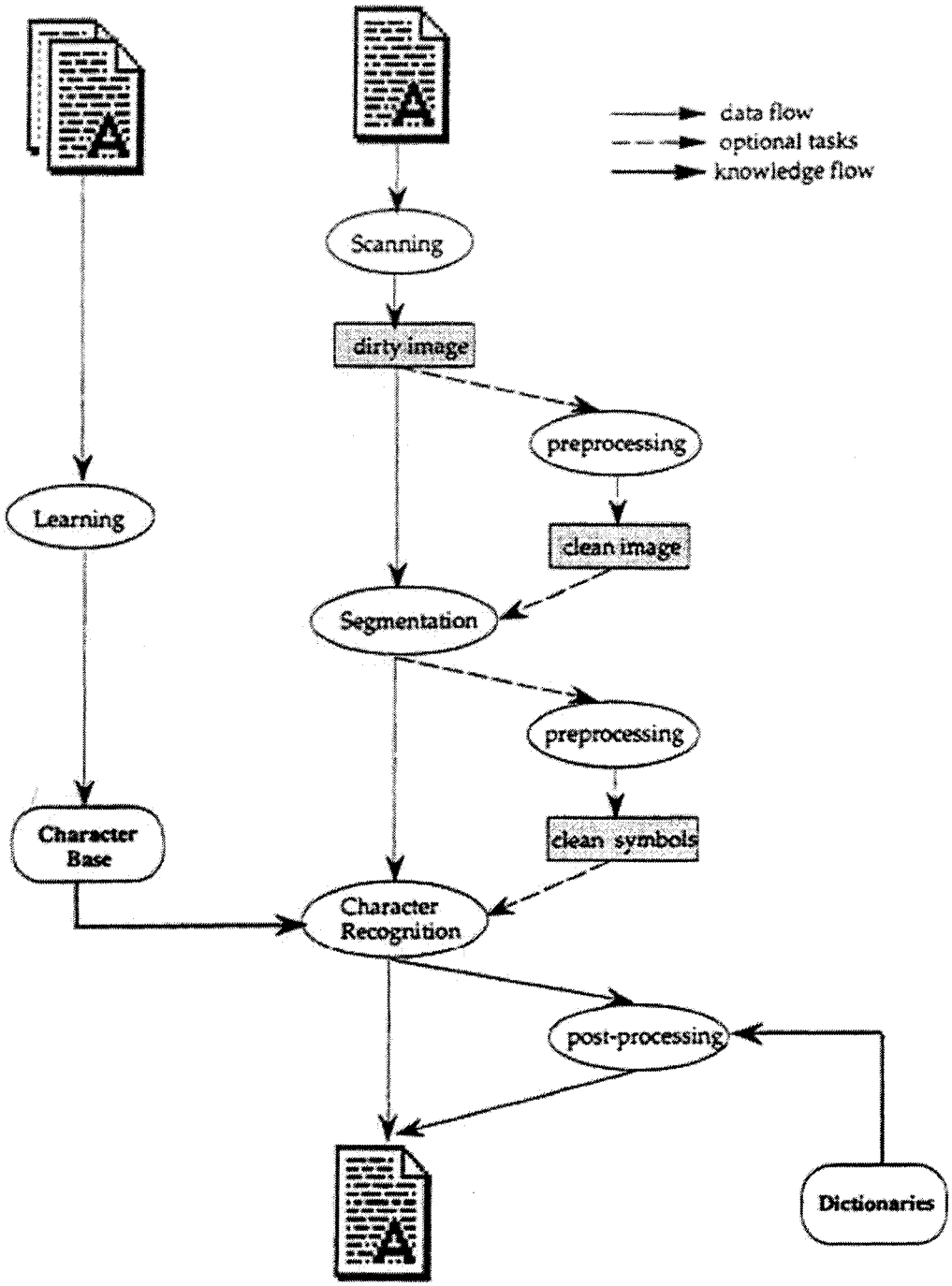


Figure 2.1. A general architecture of OCR systems.

2.5 Performance Evaluation

Many OCR solutions are available on the market and the customer choice may rely on several principles. Some useful criteria to evaluate OCR technologies is presented ahead and the results of evaluation experiments performed by various authors are discussed.

2.5.1 Performance evaluation criteria

Some of the criteria have already been presented in [KNRN93]. They are grouped in three categories according to character recognition accuracy, document analysis facilities and economical aspects.

2.5.1.1 Character recognition

While accuracy remains the principal criterion, the rest have to be considered in OCR evaluations:

- Class: is the OCR system mono-font, multi-font or omni-font?
- Accuracy: the established practice in OCR evaluation is recognition accuracy determination, i.e. the percentage of characters or words correctly recognized. Few studies were devoted to debate on the difference among several types of errors. Two factors are of significant importance: the 'rejection rate' and the 'substitution rate'. While the rejection rate might be as high as a few percentage point, a zero substitution rate is often necessary. Indeed, for unique applications

(bar code industry, postal environment) a substitution may have dramatic consequences (wrong amount, wrong destination, etc.). Unfortunately, such a distinction is rare and is missing in few published evaluations of commercial systems [Egl94, RKN93, CP93].

- Learning capabilities: OCR packages are often adjusted to identify a fixed set of characters (e.g. the Latin alphabet or only numerals). Since, for individual needs (scientific domains, such as chemistry, physics or particular alphabets), documents may include some unusual symbols (e.g. mathematical and chemical signs as well as special characters), the OCR system has to offer users with the capability to attach new signs to the class library. The addition of new classes also has to be performed through a user-friendly learning procedure.
- Image quality: the quality of the image used as input for the OCR device depends on many factors:
  - Printing device: the printing technology used to produce documents may also influence their quality. Documents produced by laser printers have an improved quality as compared to those produced by dot-matrix devices;
  - Poor quality documents: photocopying or faxing change the quality of documents and so the corresponding images;
  - Scanning: a scanning device breaks down an input page into pixels and generates a bit-mapped image that is a graphical representation of the page. The image produced is highly dependent on the page quality as well as on the scanning conditions (skew, thresholds, resolution, etc.).

Dirty images add up to a major problem for current OCR technology because even small marks can obscure vital parts of a character or switch a letter into another one (e.g. 'c' into an 'o'). If a document has been photocopied repetitively, some characters may be thinned to the point of breaking, or fattened until they bleed together. In a skewed image, characters are distorted producing classification errors.

- Multi-lingual documents: managing multi-lingual documents is a regular office task. Hence, OCR packages have to be capable of recognizing various languages, which may be 'a priori' known or not.

2.5.1.2 Document analysis

Some applications may need the recovery from the document image of more than the ASCII characters. The paragraph detection, the detection of the reading order in multi-column documents, the typeface retrieval are also needed to carry out an intelligent document recognition. Following are some criteria linked to document analysis:

- Automatic zoning: corresponds to the first step in an OCR process. It allows an automatic location of the text regions and their reading order. The system locates the text columns and identifies a distinct zone for each one, so that the generated text will be de-columnized. In addition, the system discovers non-textual regions (graphics, images, etc.) in order to exclude them from the character recognition process. Furthermore, the delimitation of some complex textual regions (e.g. tables and formulas) poses serious problems [Azo95].

- Typographical properties retrieval: much typographical information can be recovered from the document image:
  - Page formatting, which includes margins, line spacing, justification mode, etc.;
  - Typeface attributes, such as the font family, size and weight;
  - Character coordinates in the image (position, width, height, etc.).

  The use of typographical information may be very helpful in an interactive environment for OCR correction and reviewing, or at least for document reprinting [CI94].
- Open software architecture: concerns the capability of the OCR package to be incorporated into another application. The zoning may be performed by another toolkit and the OCR may be called separately on each zone under the control of a host application. Open software OCR packages are often presented as a library with an API (Application Programming Interface) to be associated with the application.

2.5.1.3 Economical criteria

Taking economical aspect into consideration, price, speed and time required to control efficiently the system are relevant criteria:

- Throughput: is measured by counting the number of characters or words correctly recognized and dividing it by the time it took to recognize the document [Egl94].

An accurate package that runs slowly can have a better throughput measurement than a fast one that makes lots of errors. Hence, accuracy is more important than speed.

- Price: plays a significant role in the OCR package choice. The price is usually relative to the software capabilities (accuracy, speed, multi-lingual options, platform, etc.).
- User interface: OCR packages are often used by non-specialists (office applications). The system control has to be as simple as possible: the user should only have to set some input, like the document language or the document quality (original, photocopied, etc.) before starting the recognition. Technical aspects such as the choice of scanning parameters (resolution, thresholds), the choice of the pre-processing technique must be placed under the system responsibility.
- Output format: the recognition results should be recovered in a familiar format to the user (ASCII text, Adobe Portable Document Format (PDF), Microsoft Rich Text Format (RTF), etc.).

Actually, there is no OCR system which assures all the presented criteria. According to their individual requirement, users may concentrate in their evaluations on some criteria more than others.

2.5.2 Experimental evaluation

Evaluations of commercial omni-font OCR systems have been completed [Egl94, CP93, RKN94]. The more thorough evaluations have been carried out by the Information Science Research Institute (ISRI) of the Las Vegas University. Based on accuracy and automatic zoning criteria, they carry out a yearly evaluation of the performances of some commercial OCR systems. In the 1994 report, eight OCR systems were tested on a data set of 500 pages. The selected documents were quite diverse with a considerable variety of typefaces and type sizes. The page quality ranges from perfect originals to several generation photocopies [RKN94]. The following criteria were considered in the evaluation:

- Accuracy: the average recognition rates range from 95.52% to 98.48% at the character level (character recognition), but drop at the word level (word recognition) to be between 89.64% and 97.31%;
- Font features: each document page was allocated to one of two groups depending on whether it contains mostly proportional or fixed pitch text. Each OCR system achieved a higher character accuracy when dealing out fixed pitch texts. Similarly, the page base was subdivided depending on whether a page contains mostly serif or sans-serif text. Higher accuracies were attained on the seriffed texts;
- Image quality: the performance depends greatly upon the quality of the document. For good quality images, the character recognition rates are all higher than 99.60%, while they vary from 84.58% to 94.83% for degraded document images

(skewed and photocopied documents). Systems were also tested on images of different resolutions (200, 300, 400 dpi) where the best recognition rates were accomplished on the 300 dpi images. This shows that recognition algorithms were tuned to that resolution;

- Automatic zoning: most of the OCR systems can not present accurate zoning on multi-column texts. The same result is also valid for complex textual structures (tables, etc.). They offer tools to label the document zones manually.

The report has also shown an accuracy progress in one year for almost all OCR systems (1993's vs. 1994's accuracy).

Most of the available OCR systems satisfy only some of the criteria of Section 2.5.1. Some of them try to maintain the text format (Omnipage, WordScan Plus), others are good in handling multi-lingual documents (Recognita Plus) but fail on dirty (photocopied) documents [Egl94].

2.6 Conclusion

This analysis has shown the limitations of the existing OCR technologies. Indeed, R. Casey stated in [CW90] that a 99% character accuracy can only be obtained by commercial OCR products if "a printed document is a fixed-pitch, typed original or clean copy, in a simple paragraph format and in a common typing font". Although a 99% accuracy seems almost perfect, the error level is annoying since there are approximately

3000 characters on a text page. Even a 99.5% success rate still produces up to twenty errors per page, involving a considerable human intervention.

Even on 'ideal' images, i.e. noise and skew free images synthesized by a computer program (e.g. generated from PostScript files), OCR systems can not achieve 100% accuracy. Indeed, some experiments on eight OCR systems have shown that the best once performs only 99.9% accuracy on 'ideal' images with very common typefaces (Times, Helvetica, Courier) and sizes (10pt, 12pt, 14pt) [RKN93].

Since they make generalization of the font and size, omni-font OCR systems are unable to recognize the fonts used in the evaluated documents. Some of them claim to detect typographical attributes such as the text style (regular, bold, italic, etc.). With the current classification system, a solution to the font recognition problem is provided, which can assist OCR systems to improve character recognition.

CHAPTER 3

OPTICAL FONT RECOGNITION

The main objective of Optical Font Recognition (OFR) is to recover from images fonts with which texts have been printed. In spite of its worthiness for both document and character recognition, OFR has often been ignored. The recognition of fonts can be completed either before or after OCR. In each approach an influential combination of both mechanisms can offer a perfect recognition of characters as well as their typographical attributes.

The approach adopted within the classifier system allows the recognition of fonts without taking into account characters appearing in that text (omni-char OFR). An intelligent character and font recognition can be accomplished by a joint approach based on the association of an omni-char OFR with a mono-font OCR.

The issue of font recognition in the document analysis context, will be introduced in Section 3.2. font and Character recognition techniques will be addressed in Sections 3.3 and 3.4. Section 3.5 shows that features allowing discriminating fonts can be extracted locally from individual characters or globally from large text entities.

All the figures in this chapter were taken from [Zra95] with the permission of the author

3.1 Introduction

Very limited data on font recognition exists since, in the optical reading perspective, the key efforts have been dedicated to character recognition, especially to omni-font OCR which does not require font identification. During the last decade, font recognition has emerged as an essential task for document analysis. The SSPR'90 working group on character recognition stated that [BHN92, pp.566]:

> "The detection of the font style, point size etc. of a text is an obvious way to improve the capabilities of text recognition algorithms. This would allow for hundreds of fonts to be used for training but retain the recognition accuracy and potential speed of a system that uses a small number of fonts. This appears to be a promising but hitherto almost neglected topic."

Font recognition is still presumed to be a complex task [BN94]. It is practical and vital in diverse domains [Mor92, ZAI92, ZI94, ZI95]:

- A simple reprint of a scanned document involves not only character recognition, but also the detection of fonts used to produce it. Indeed, a realistic reproduction of a printed page needs the identification of various typographical information such as fonts, justification mode, margins, line spacing, etc.;

- The detection of logical document structures, where the information about the font used in a word, a line, or a text block may be significant to establish its logical label;
- The knowledge of the font may also boost up character recognition. Since fonts are expressed by characters, recognition of the font provides information on the structural design of characters. The conveyed information could be used by OCR systems to identify the characters as recommended by the SSPR'90 working group;
- Interactive environments for OCR error correction and assessment may benefit from font recognition. Superposing the distinguished characters and their images appears to be a clear way to deal with OCR correction [CI94]. An accurate superposition needs the classification of the typographical attributes of the examined characters (position in the image, font, etc.).

Font recognition is generally performed to identify the font changes in documents. However, the shape, size and spacing of letters in words influence the appearance of the document more than any other single visual element [Par88]. The choice of the font and the way it is positioned on the page can improve the reader's ability to comprehend the message of the document.

In a document, diverse fonts are used in order to highlight some parts of the text so that they can be easily observed by the reader. Normally, font changes in a document may occur at particular positions (titles, indexes, references, etc.). They might be prepared by

preferring another typeface, or altering the style or the size of the same typeface, i.e., regular typeface for the running text, bold for titles, italic for references and mono-spaced typeface for program listings. In reality, one makes a small number of systematic font changes in a particular structured document.

3.2 Font recognition for document analysis

In a document, two level structures can be labeled. In the perspective of document recognition, font identification may be useful for the detection of the logical macro- and micro- structures.

3.2.1 Macro-structure recognition

The 'macro-structure' expresses the high level structure of a document down to fragments. For example, consider Figure 3.1, which shows a graphical demonstration of a scientific paper. Several high level entities can be measured. In the illustration, the entities: 'title', 'authors', 'affiliation', 'email', 'abstract', 'keywords' and 'section' can be easily distinguished.

Figure 3.1 demonstrates that the 'title' of the paper was printed using a bold and large sized typeface, however, the 'author' fragment was printed using capital letters, the 'affiliation' fragment using an italic font, and the 'running text' with a regular font. The

classification of each font can help the identification of each fragment. At this level, the examination can be based on text lines.

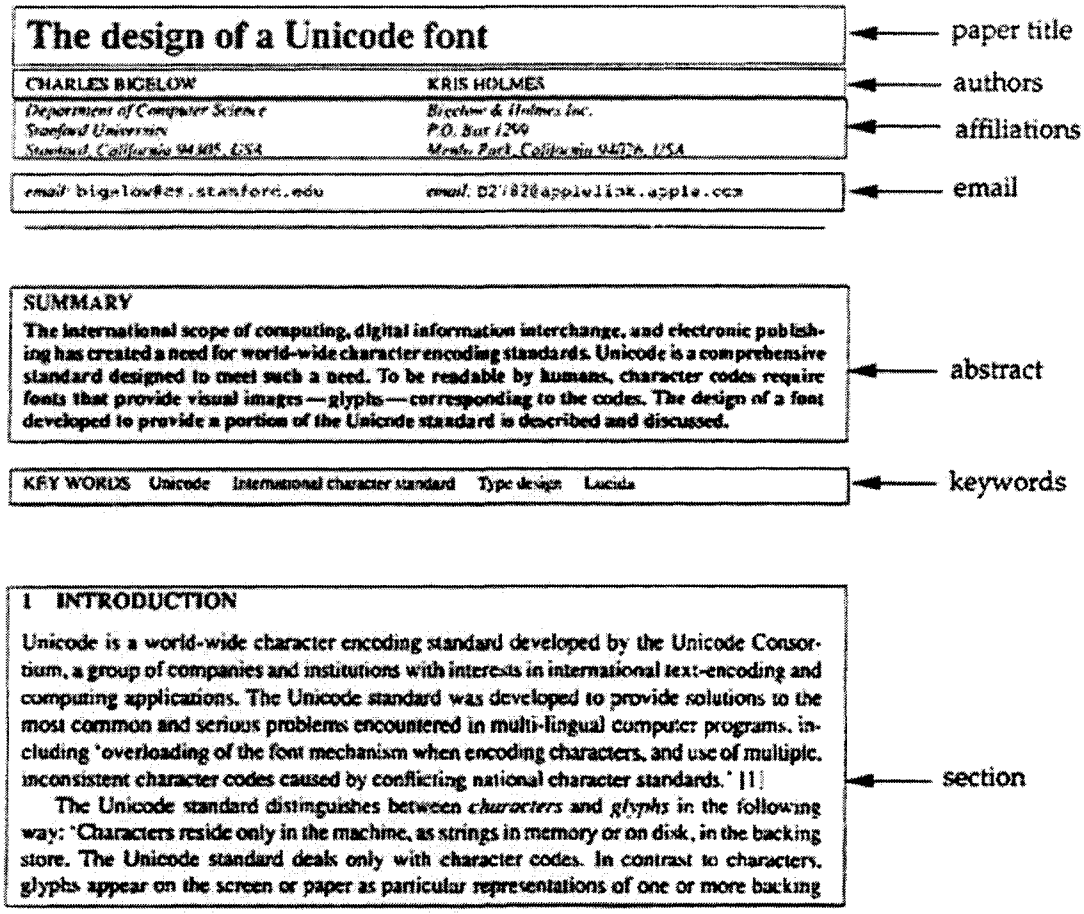


Figure 3.1. Macro logical structure.

3.2.2 Micro-structure recognition

The 'micro structure' expresses the lower structure level of each fragment down to characters. The study of bibliographical references copes with low level entities. It needs the recognition of small text regions including few words wherein the same text line,

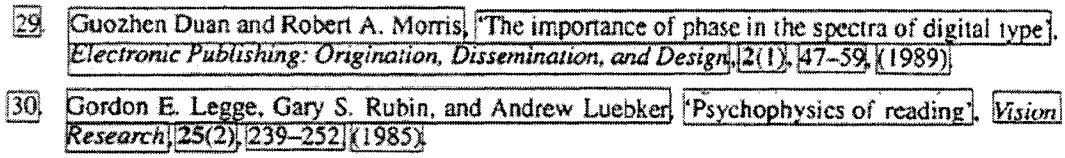
several logical entities may be found. The logical entities are discriminated by the font used to print them or by particular marks (brackets, punctuation, etc.), as shown in Figure 3.2. The examination may be based on the recognition of typographical attributes as well as on the text content. It may take advantage of an accurate font recognition pertained to single words.

Several studies have been dedicated to the analysis of both macro- and micro-structures [Che93, Hu94]. In his bibliographical references recognition system, Chenevoy [Che93] based the study on a syntactical description of the references as well as on the search of significant features directing the system. The analysis centered on the detection of special characters such as [ ] ( ) → , . : and also on the finding of typographical attributes of text lines within the considered documents. Related words are differentiated from running text words by their diverse styles: bold, italic, underlined, capitals, small caps, etc. The experiments have revealed the complexity of extracting relevant features from multi-font and noisy documents. However, the system achieved excellent outcomes when the references had stable structures.

3.2.3 Character recognition

The limitations of omni-font OCR technology were mentioned in the previous chapter. While an omni-font OCR overlooks serifs, a mono-font OCR takes a great advantage from such information. The serif presence or absence and its shape give accurate

indication of the character shape. For example, serifs can discriminate a '1' from an 'I', 'P' from a 'p', etc.


29. Guozhen Duan and Robert A. Morris, 'The importance of phase in the spectra of digital type', *Electronic Publishing: Origination, Dissemination, and Design*, 2(1), 47–59, (1989).

30. Gordon E. Legge, Gary S. Rubin, and Andrew Luebker, 'Psychophysics of reading', *Vision Research*, 25(2), 239–252, (1985).

Figure 3.2. Typographical structure of bibliographical references.

3.3 Font recognition approaches

Font recognition may be merged with character recognition according to two approaches, which varied in the manner OCR-OFR interactions are performed.

3.3.1 'A priori' font recognition

An 'a priori' font recognition approach comprises identifying the text font without any knowledge of the characters that appear in that text. The OFR can be based on attributes extracted from global properties of the text image, such as the text density, letters size, orientation and spacing. Features may then be extracted from text entities with different lengths such as words, lines or even paragraphs.

In an 'a priori' font recognition approach, font and character recognition is achieved in two steps as illustrated in Figure 3.3:

1. Font recognition methods, called 'omni-char OFR', are applied on the text image in order to detect its font;
2. A mono-font OCR is then employed on that text using the identified font. Under the hypothesis that mono-font OCR offers superior results if the font is correct but leads to a high unrecognized rate otherwise, this recognition rate might later be used to verify or reject the results generated by font recognition.

3.3.2 'A posteriori' font recognition

An 'a posteriori' font recognition approach consists of identifying the font of a text using the knowledge of characters appearing in it. Hence the OFR can utilize features based on local properties of individual letters. However, the letter shape relies totally on the font family (Times, Helvetica, etc.) and style (roman, italic, bold), such as letters 'g' and '*g*', 'a' and '*a*', etc.

In such an approach, font and character recognition may be executed in two steps, as illustrated by Figure 3.3:

1. An omni-font OCR is applied to identify the text characters;
2. A character specific OFR algorithm, called 'mono-char OFR', is applied on each character in order to recognize its font.

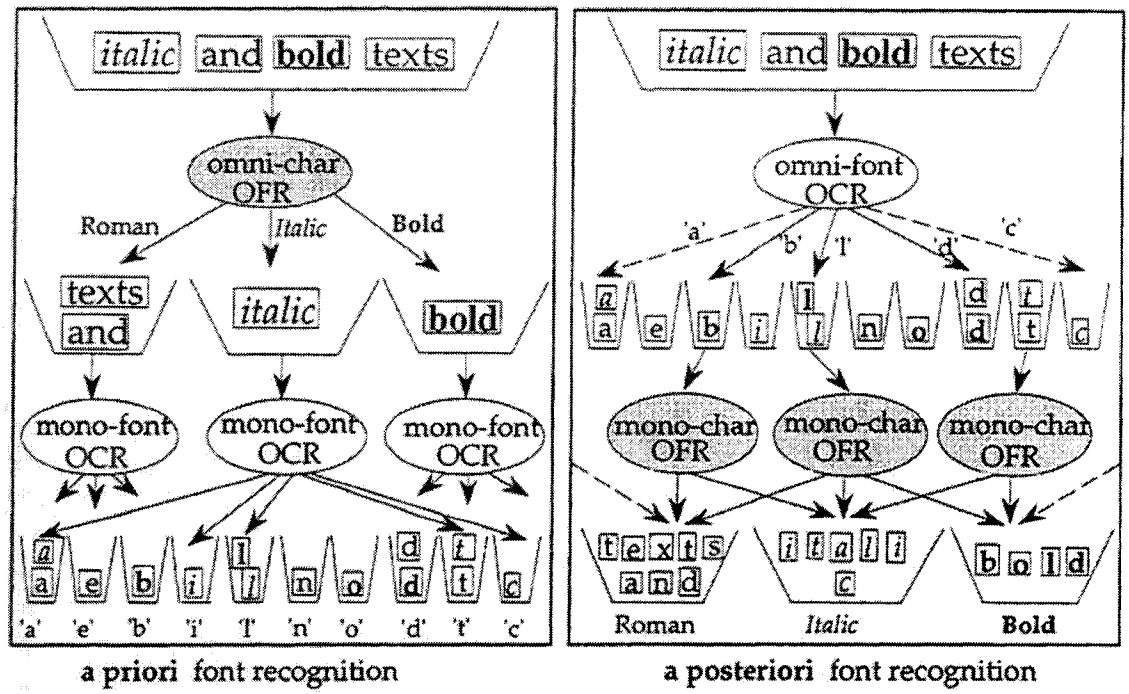


Figure 3.3. OCR-OFR combination strategies.

3.3.3 Cooperative recognition

A 'cooperative' approach that merges the two preceding ones, can also be described. It is largely based on the association between the OCR and OFR components. In such an approach, the text is recognized (font and characters) after some OFR-OCR interaction cycles. Such association will be discussed in the next section.

The perspective in which the OFR package is used proposes the implementation of one font recognition approach. The choice of a particular strategy is greatly influenced by the OCR capabilities: a perfect omni-font OCR package can be well merged with either an

omni-char OFR or a mono-char OFR, while a mono-font OCR requires an omni-char OFR package.

3.4 Font and Character Recognition

In order to boost up the performances of OCR algorithms, the current OCR technology has implemented several approaches, which are as follows:

- Conventional methods that contain post-processing using lexicons and n-grams to correct OCR spelling-errors [Dam64, TIAY90];
- Approaches that incorporate character recognition with contextual analysis, such as word shape, syntactic and statistical models of recognition devices, characters and words, local typeface homogeneity [HHS91, JSB91, BN94];
- Techniques that employ typographical constraints derived from character shapes [Sen94];
- Procedures that combine numerous OCR packages applied on the same image. The combination often uses voting algorithms [Ani92, LZ94].

In the following, a focus on the 'a priori' font recognition approach, implemented by the classifier system and used in the OSCAR-II project [Hu94, ZI92], is done. A special effort is made to describe the relation between the OFR and OCR packages in their association to distinguish characters. An intelligent character recognition system, which identifies characters and their typographical attributes, may merge a 'mono-font OCR'

with an 'omni-char OFR'. The combination is made under the hypothesis that the knowledge of the text font boost up the character recognition.

3.4.1 Omni-char font recognition

Using a base of font models from training samples, the OFR system allocates a font, from an 'a priori' known font list $\{f_1, f_2, \ldots, f_n\}$, to a text entity (word, line) symbolized by its image (see Figure 3.4). The font identification is executed without any information on the characters showing up in the text.

The system returns a list of couples $<f_i, p(f_i)>$, where,

- f_i signifies a font identifier;
- $p(f_i)$ indicates the degree of confidence for f_i, i.e. the conditional probability $p(f_i \mid \text{text})$ that the text was printed with f_i.

The list is sorted according to decreasing values of $p(f_i)$, i.e., the first element in the list is the most likely result.

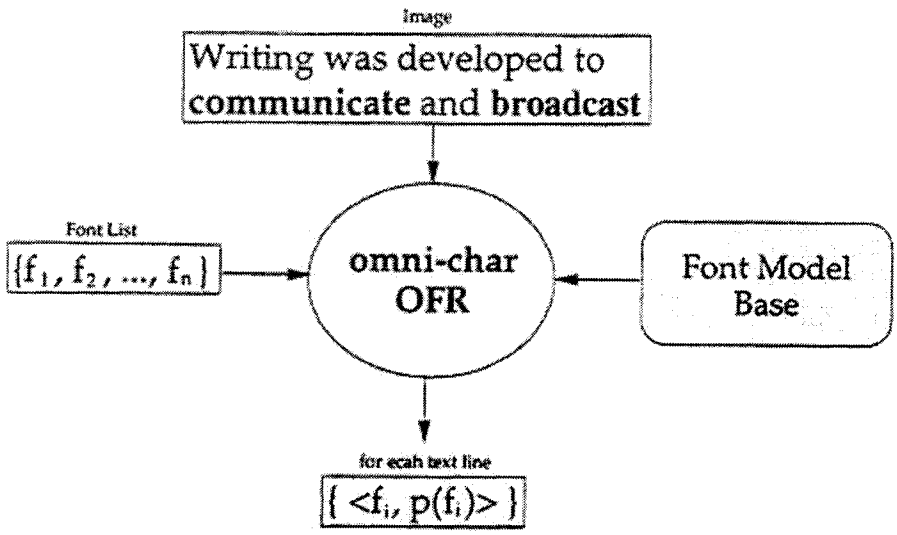


Figure 3.4. Omni-char font recognition.

3.4.2 Mono-font character recognition

Character recognition is carried out by a 'mono-font OCR' using a base of font dictionaries. Each dictionary includes character models of a given font. The system presumes the 'a priori' knowledge of the font f_j in order to use the appropriate dictionary, as shown in Figure 3.5.

For each character, the system returns a sorted list of triplets $< c_i, p(c_i), coord(c_i) >$, where:

- c_i represents a character class;
- $p(c_i)$ indicates the probability that the pattern corresponds to c_i (from the font f_j);
- $coord(c_i)$ represents the character coordinates in the image. They are useful in the correction and demonstration of the OCR results. The coordinates may vary from

one solution to another for the same bitmap. For e.g., the sign 'e' may be identified as the character 'e', or as the character 'e' with noise above. The two solutions produce different character coordinates.

The recognition can also be achieved at the word level. In this case, the characters of the same word are clustered and a confidence degree for the whole word is figured out.

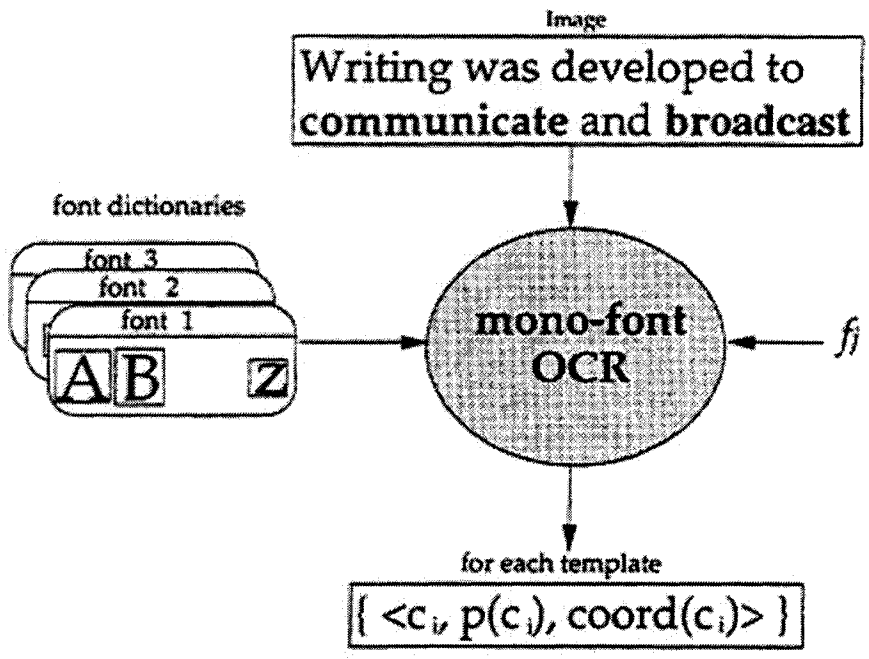


Figure 3.5. Mono-font character recognition.

3.4.3 Cooperative recognition strategy

It is assumed that:

- The OFR package can affix probabilities among an input set of fonts, as explained above, and,

- The OCR package provides superior results when the font is correct and leads to a high unrecognized rate otherwise.

In this case, a line containing strings of different fonts can be identified after a consistent sequence of some OCR-OFR steps, as illustrated in Figure 3.6.

3.4.3.1 A collaborative scenario

Considering the situation in Figure 3.6, in order to distinguish a text line of two words:

1. 'Recognition' printed with font lb-bold-12pt;
2. 'strategies' printed with font lb-regular-12pt.

It is assumed that it is known that the document contains entirely lb fonts.

In the first step, the omni-char OFR is applied on the text and only the three best candidates are selected:

{

< lb-b-12pt, 0.65 >,

< lb-r-13pt, 0.25 >,

< lb-b-13pt, 0.08 >

}

The mono-font OCR package is then applied on the image using the selected fonts. The OCR results prove that

- The first word is definitely printed with lb-b-12pt;
- The second word may be printed with lb-r-13pt, but certainly not with lb-b-13pt.

In the second step, the OFR is requested to propose other candidates for the second word only. The lb-b-12pt and lb-b-13pt are obviously excluded from the font list. Two fonts are selected with a leading one, i.e. lb-r-12pt. The preferred font is finally established by the OCR to be the superior one.

In the demonstrated approach, the OCR is used to satisfy two requirements:

1. To carry out character recognition using the knowledge on the font;
2. To validate or reject the OFR solutions. A high character recognition rate substantiates that the text is printed with the selected font, and a low rate proves that the text is printed with another font.

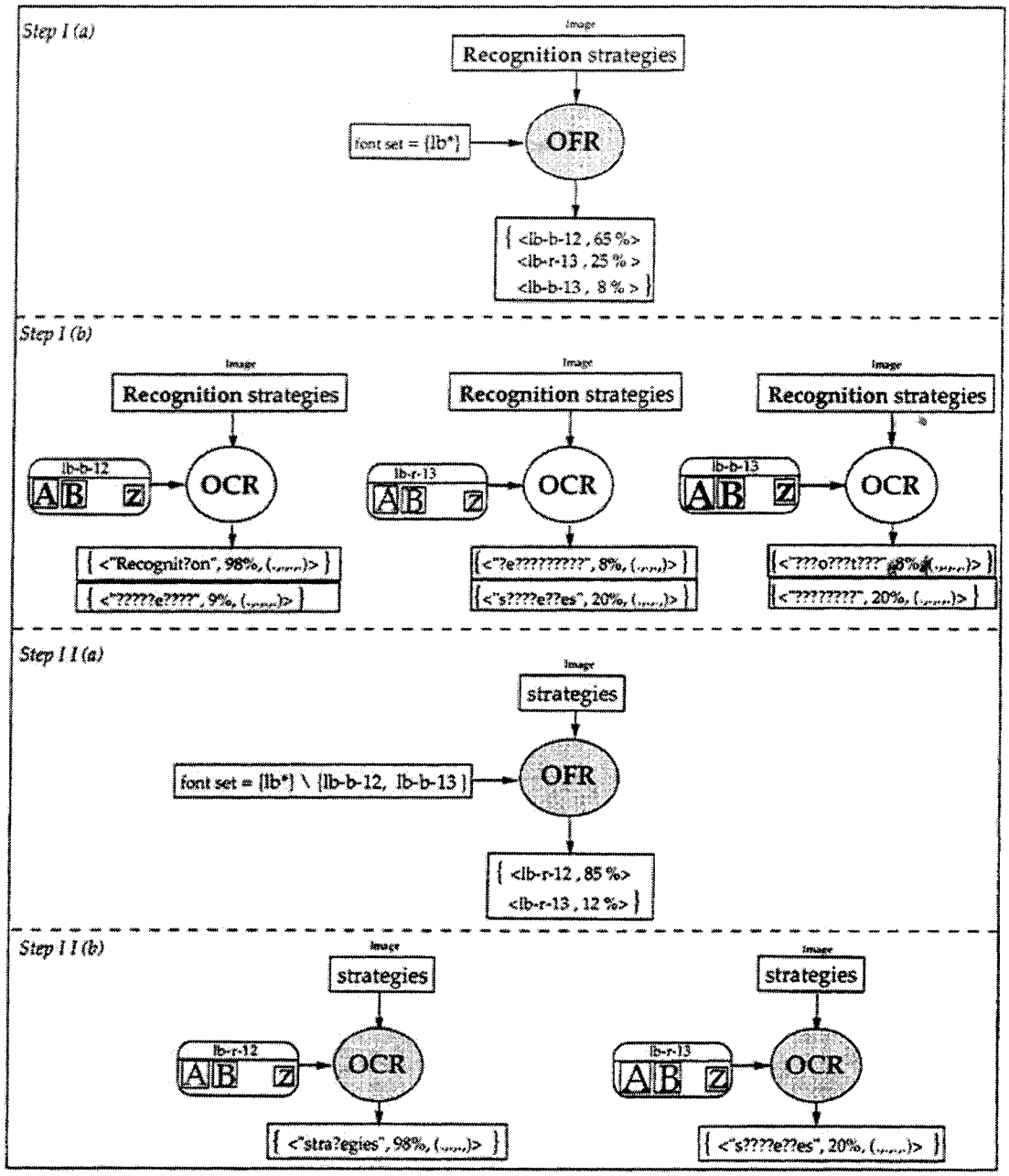


Figure 3.6. Cooperation between the OFR and OCR components to recognize words.

3.4.3.2 Output format

A font and character recognition system can produce the recognition results using a range of formats that contain both character and font information. The results can be presented as character collections or word collections. The following format, based on the font changes and focusing on word collections, may be implemented. The recognition results are given as a sequence of (font, text):

[{f, p(f); {w, p(w), coord(w)}*;}*]

The results of the preceding scenario according to the presented format are shown below:

['lb-b-r-12, 0.65, 'Recognit?on', 0.98, (10,34,50,35);

'lb-r-r-12, 0.85, 'stra?egies', 0.98, (70,34,45,34)]

Such a technique, merging an omni-char OFR with a mono-font OCR, has been investigated within the CIDRE project 'Cooperative Interactive Document Reverse Engineering' [BBI95, BBZI96].

3.5 Feature extraction for font recognition

Similar to any pattern recognition problem, font recognition is based on the extraction of a set of attributes from document images. These attributes can be worked out locally from

individual characters or globally from large text entities such as words, lines or even paragraphs. In this section, the issue of how features can be locally or globally extracted is presented.

3.5.1 Local feature extraction

In such an approach, feature extraction concentrates on the localization of the character particularities, such as the serif's shape and on the depiction of special characters like '*g*' and 'g', 'a' and '*a*', etc. For a given alphabet, only a few characters hold the most typeface character. As shown in Figure 3.7, the shapes of characters 'a', 'g' and 'w' rely greatly on the typeface, while those of characters 'z' and 'o' have roughly the same structure for all typefaces.

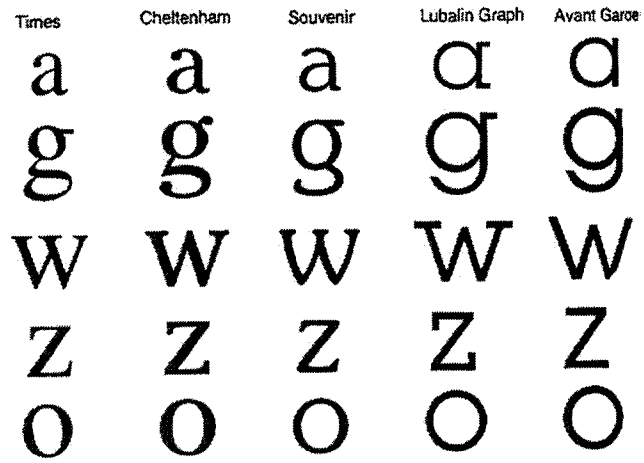


Figure 3.7. Typeface variants of characters a, g, w, z and o.

Hence, relevant features can only be sensed on a limited character set. Collier [Col91] founded a list of characters distinguishing typefaces. Following is the character list, for various categories, in a decreasing significance order (see Figure 3.8):

- Uppercase letters: Q J G W A K C R M E P S T F B N O U X Y D H Z L V I
- Lowercase letters: g a j y k t f r q w e b s c d p m u x o v h n i l z
- Numerals: 3 7 5 2 1 4 9 6 8
- Specials: & % $

Recovering these attributes from real documents requires very careful processing. Since they focus on small and particular character parts (e.g. serifs), the extraction of local features involves smooth character contours. Such contours are greatly influenced by the image condition (noise, skew, low resolution, binarization thresholds, etc.). They can, yet be provided by gray level images produced at a relatively high resolution. Additionally, the extraction of local features involves:

- The presence of the relevant characters in the text block (a, g, Q, etc.);
- The 'a priori' knowledge of the character classes. For example, it is obligatory to recognize that the examined contour is of letter 'g' in order to identify whether the loop is closed (g) or open (*g*).

If these conditions are fulfilled, then such features can be used to differentiate typefaces. The slope detection can be achieved using local features (seeking a negative angle). The weight and size recognition is, though, more difficult and even impracticable, because

characters maintain the same shape, where only a horizontal or a vertical scaling is performed.

Courier	a	g	j	y	s	%	&	$	Q	J	G	W	3	6	9	7
Times	a	g	j	y	s	%	&	$	Q	J	G	W	3	6	9	7
Palatino	a	g	j	y	s	%	&	$	Q	J	G	W	3	6	9	7
Lucida Bright	a	g	j	y	s	%	&	$	Q	J	G	W	3	6	9	7
Bookman	a	g	j	y	s	%	&	$	Q	J	G	W	3	6	9	7
New Century	a	g	j	y	s	%	&	$	Q	J	G	W	3	6	9	7
Garamond	a	g	j	y	s	%	&	$	Q	J	G	W	3	6	9	7
Souvenir	a	g	j	y	s	%	&	$	Q	J	G	W	3	6	9	7
Lubalin Graph	a	g	j	y	s	%	&	$	Q	J	G	W	3	6	9	7
Helvetica	a	g	j	y	s	%	&	$	Q	J	G	W	3	6	9	7
AvantGarde	a	g	j	y	s	%	&	$	Q	J	G	W	3	6	9	7
Franklin Gothic	a	g	j	y	s	%	&	$	Q	J	G	W	3	6	9	7

Figure 3.8. Character variants characterizing typefaces.

A local feature extraction policy fits well on 'a posteriori' font recognition approach where characters are identified using an omni-font OCR before the detection of their typographical attributes.

The extraction of local features has often been used to execute OCR. Anigbogu [Ani92] in his multi-font OCR system, classified models for characters and placed them in a tree according to certain typographical attributes (ascenders, descenders, holes, etc.). A little pre-processing is done on the text in order to choose one sample for each kind of shape (characters), which are then positioned in the tree according to their attributes. In another operation, a tree is created for each font (instantiation of a generic tree with the diverse

characters of the given font). The shape tree is lastly matched with each font tree to compute a distance, where the smallest distance defines the associated font. Font detection is primarily done to improve OCR performance by limiting the search space (font trees). This approach appears to offer superior results when the generated shape tree is complete enough, i.e., it has a sample for each character.

3.5.2 Global feature extraction

Global features can be extracted from large text entities such as words, lines, or paragraphs, without considering their content. Font changes are easily noticed, even by non-experts in typography. In an illustration in Figure 3.9; it is very easy to observe that characters of the third line are bolder than those of the first and second lines. Additionally, serifs, spacing mode and text height changes are also visible to the text reader. The recognition of the font changes presumes long enough text entities, each one homogeneously typeset.

However, it can be realized that global features are not capable of discriminating too similar typefaces such as Times and Palatino (lines 1 and 8), because they can not capture their subtle distinctions (they have visually the same weight, serifs, etc.). Merging very similar fonts is, though, very rare in real documents.

Since they signify rough properties of texts, global features may rely on the length of text (number of characters) more than on its content. They can also be tolerant of the image conditions, i.e., they can be extracted from binary images scanned at low resolutions.

Global features can be well utilized within an 'a priori' font recognition approach, since they do not concentrate on the character contents. Hence they can be performed without OCR.

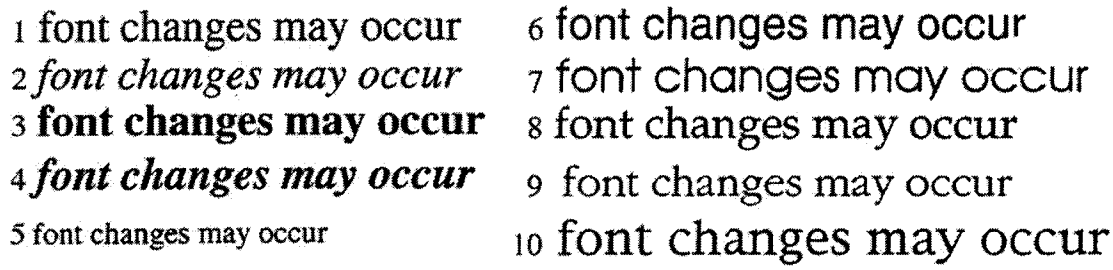


Figure 3.9. Global features discriminating fonts.

The application of global features to execute font recognition was addressed by Morris [Mor92]. He based the study of the problem of digital font recognition on the examination of Fourier amplitude spectra extracted from word images. The study was largely done to observe the applicability of human vision models to typeface discrimination, and to explore whether spectral features might be practical in typeface creation. He applied a Fourier transform to the word image and then extracted a feature vector by applying many filters to the resulting spectra. Numerous font classification experiments were devised, where a quadratic Bayesian classifier was applied on 55 fonts. The classifier achieved fantastic results with an average error rate of 6%, and a very low error rate when the classification was carried out within fonts of the same typeface.

On the other hand, important simplifications were completed: (a) the images evaluated were noise-free since they were produced by software instead of being scanned from paper documents; and (b) only one font size for the dissimilar samples was considered.

Kopec completed an attractive OFR related work, which consisted of estimating font metrics (baseline, side bearings, kerning, etc.) from text images. The estimation was exercised within a text images editor [Kop93].

3.6 Font recognition and related fields

The font recognition problem lies at the intersection of varied areas, as shown in Figure 3.10:

- Document analysis and recognition: OFR is, as discussed earlier (see Section 3.2), a significant topic in document analysis and recognition. It is also connected with other tasks such as segmentation or OCR.
- Typography: typefaces are drawn by type designers with respect to artistic and scientific concepts. Much knowledge from typography, especially Digital Typography is investigated and exploited to model fonts and to determine discrimination features.
- Image processing: since images characterize the raw material from which fonts and characters are extracted, some aspects of image processing are considered;

- Pattern recognition: fonts are modeled by feature descriptors (extracted from images) permitting their discrimination. The font modeling and the classification process are based on a statistical pattern recognition approach.

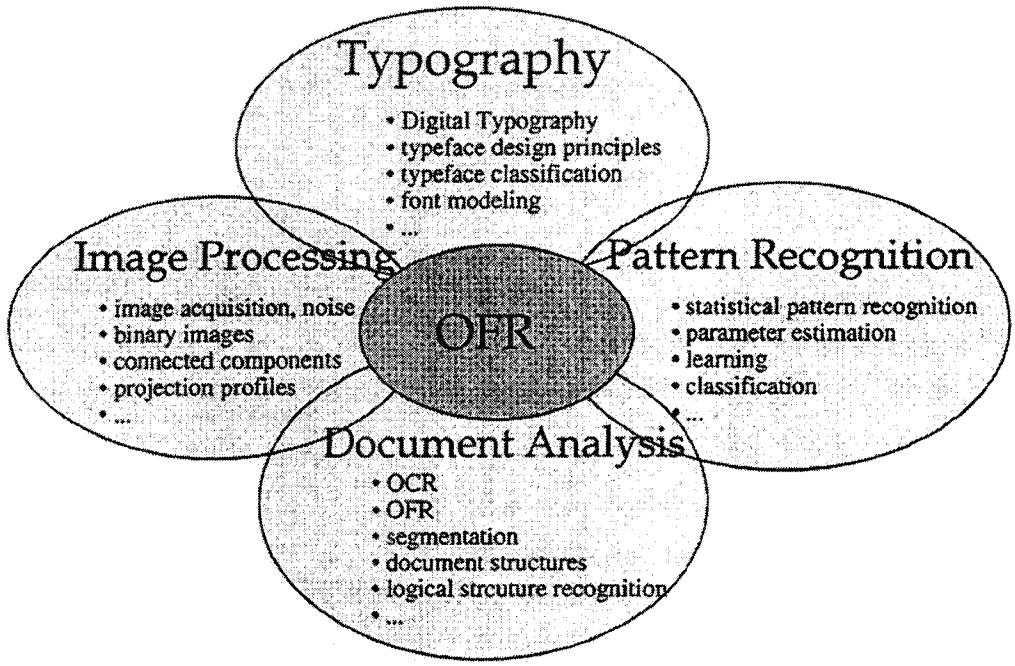


Figure 3.10. Font recognition and its interaction with other fields.

3.7 Conclusion

It has been proved in this chapter that OFR is valuable, not only for document analysis, but also for character recognition. Despite of its significance, OFR has been ignored by the optical reading community. Through the classifier system, the OFR problem in the context of document analysis and recognition will be addressed.

CHAPTER 4

TYPEFACE DISCRIMINATION

This chapter discusses the issue of font specification and classification from two points of view: typeface design and font recognition. Typefaces are largely recognized by their writing style, serif shape, the relative proportion of the x-height in the font size and the spacing between characters. Fonts belonging to the same typeface are differentiated by their slope, weight and size.

Section 4.1 details some features differentiating typefaces and fonts. Section 4.2 summarizes typeface classification and presents statistics measured by P. Karow for several typeface measurements computed from a typeface base.

4.1 Typeface and font discrimination

Due to their abstract design, typefaces require instantiations into fonts in order to be depicted. A font conveys:

- The typeface style that discriminates it from other typefaces;
- Its intrinsic character that distinguishes it from other instantiations of the same typeface.

All the figures in this chapter were taken from [Zra95] with the permission of the author

Hence, fonts can be denoted at two levels: the typeface level (inter-typeface) and the font level (intra-typeface).

4.1.1 Typeface identification

Some general type design elements which offer the stylistic character of a type face permit distinguishing one typeface from another. In this section, four elements: writing style, serifs, x-height and character spacing will be focused on.

4.1.1.1 Writing style

Type designers can take on diverse styles in their character drawings. Generally two rough styles can be easily differentiated: the cursive style simulating handwriting and the typesetter style related to typesetting machines. Cursive writing is often illustrated by words formed of connected characters (as is done with handwriting), while typesetter styles always write characters separately.

4.1.1.2 Serifs

Serifs are small strokes at the end of the character main strokes. According to their presence or absence, serifs make a distinction between two typeface families: the seriffed family and the sans-serif family. Shortly, serifs are the most evident feature classifying typefaces. As well as decorative purposes finishing the main strokes of characters, serifs

have the effect of providing horizontal flow between characters highlighting the line of reading [Lun92].

The majority of seriffed typefaces have strokes with clearly dissimilar thicknesses, while sans-serif ones have only a distinct apparent weight of stroke as shown by Figure 4.1. The figure also illustrates that serifs have a variety of shapes and dimensions. A serif may be coved, square, square coved, thin line, exaggerated or triangular. The stem ends of sans-serif typefaces also have several shapes: they may be square normal, square perpendicular, flattered or rounded. A detailed categorization of serifs is given in [Bau91].

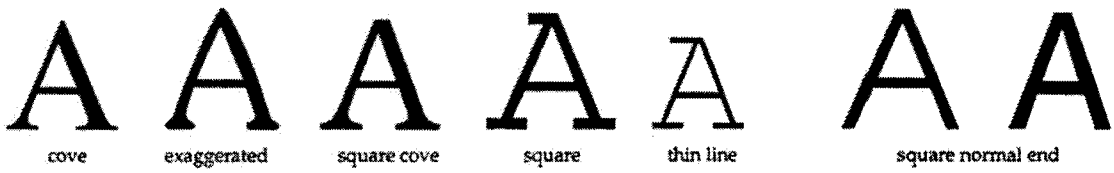


Figure 4.1. Serif shapes and stroke variations.

4.1.1.3 x-height

Although all typefaces have a regular baseline, the proportion of x-height, ascender and descender height fluctuates extremely from face to face. The most noticeable distinction between typefaces at the same point size is the relative proportion of the x-height. A typeface with a small x-height in a given body emerges with the same size as a font with relatively big x-height but with a smaller body. Hence, the size and width of texts differ, for the same point size, from one typeface to another as demonstrated by Figure 4.2.

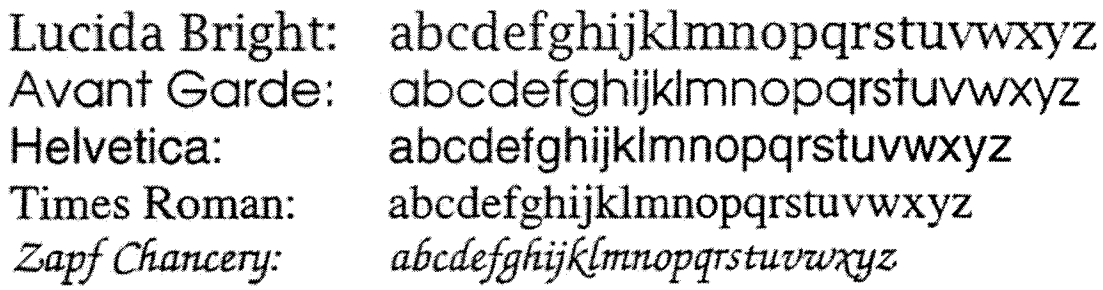


Figure 4.2. Typefaces in the same size (12 pt), but with different heights of characters.

4.1.1.4 Inter-character spacing

The expansion of industrial and mechanical hot-metal typesetting created a new problem for the designers as to how to systematize type production in order to get a unified character spacing.

The most significant model is the one-unit system of the conventional typewriter. Each individual character, whether a narrow 'i' or a wide 'w', has a unified character width. However, for the early composing machines, special unit systems were launched that resulted in a relatively high quality systematic character spacing (see Figure 4.3(a)).

The inter-character spacing depends both on the technology implemented and on the type quality desired. Normally, there exist three spacing classes:

1. Fixed spacing: each individual character takes up the same horizontal space regardless of its shape. This spacing class has been enforced by the technology state of typewriters permitting the handling of only one space. The renowned Courier typeface is a fixed spaced one. In order to have a visually balanced

typeface, the Courier designer reduced or stretched out character components. For e.g., the serifs of the character 'I' have been expanded to fill the available space;

2. Proportional spacing: each character takes up different amounts of space depending on the character shape. It is shown in Figure 4.3(a) that the 'i' of the mono-spaced font takes the same space as the 'n', while it is clearly narrower in the proportional font. Characters are devised and created such that the character position, shape and width assure a correct spacing between all character combinations.

A spacing vector is linked to each font, signifying the space value between each character and the following ones;

3. Proportional spacing with kerning: a spacing value is set for each character pair by means of a kerning table. The spacing values, stored in the kerning table, are optimized to acquire normal visual spacing. Figure 4.3(b) shows the difference between proportional spacing with and without kerning.

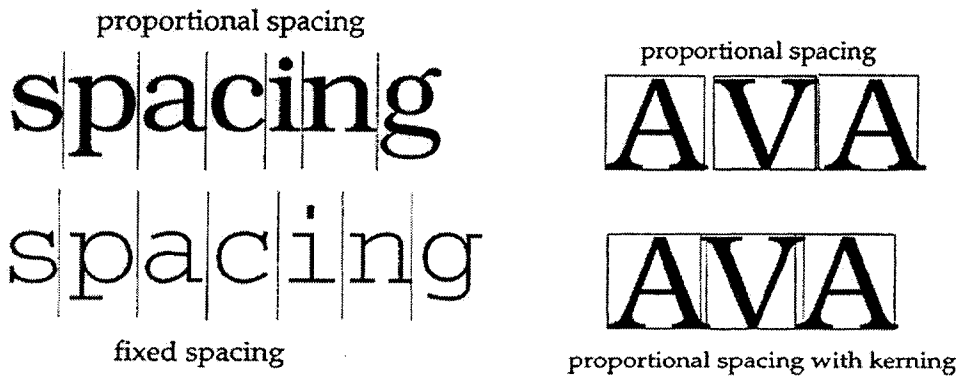


Figure 4.3. (a) Width system for characters. (b) normal vs. kerning.

Typographers claim that fixed-width fonts are harder to read than proportional-width ones. However, studies have revealed that there is a little distinction in reading rates between these fonts at normal reading size [MBHL91, MBH93, LEB94]. Morris et al. have made a quantitative study of type quality and text readability, from the human vision modeling point of view [MBHL91, MBH93]. They made a variety of deformations to character spacing and shapes, calculated classical reading rates, and concluded that loose letter spacing alone does not make fonts easier to read; instead, shape or other font quality factors do.

To conclude, it can be stated that four factors discriminating typefaces have been presented: the writing style, the x-height proportion in the font body, serifs and the spacing mode. Identifying these elements is of immense help in a font recognition system.

4.1.2 Font specification

Within the same typeface, fonts can be differentiated largely by four features: the slope, weight, width and size. These features are discussed in detail ahead.

4.1.2.1 Slope

The font slope represents the character's incline, which is actually the transposition of the handwriting incline in the printed character domain, becoming cursive due to writing speed. Fonts can be presented with a variety of slopes:

- 'Upright', which corresponds to the set of upright characters (uppercase, lowercase) without any style distinction. Upright characters may be:
    - 'Roman', which characterizes the normal upright form of a typeface, generally used for running text;
    - 'Script', which symbolizes upright form with cursive characters simulating handwriting;
- 'Slanted', which is said for each character slanted to the right. The following are differentiated:
    - 'Oblique', which is regularly used for sans-serif typefaces, where oblique characters may look like the roman shapes simple slanted;
    - 'Italic', which is often linked to seriffed typefaces. Italic is a cursive form portrayed by variant glyphs such as '*a*', '*g*', '*f*' and angle of slope, and generally by appearing lighter and narrower than the roman style.

4.1.2.2 Weight

The weight of a character is conveyed by the thickness of its strokes. For a particular sketch, the weight is determined by decreasing or increasing stroke widths. The majority of the bold sans-serif fonts are generated by building an overall thickening, where glyphs are preserved from one weight to another. Creating a successful seriffed bold font is not that simple, since only the strokes which are thick in the normal weight can be made substantially heavier. Hence, sans-serif typefaces can be devised with an abundance of variant weights, but many seriffed typefaces exist in a limited range of weights.

Normally, there are just four weights of a typeface, which are regular, medium, bold and extra-bold. Nevertheless, for some renowned sans-serif typefaces (Helvetica, Gill, Univers, etc.), more weights may be labeled such as ultra-light, light, demi-bold, heavy and ultra-bold. In fact, the practice of many weights is often limited to professional publishing.

4.1.2.3 Width

The font width expresses the amount of expansion or contraction with respect to the normal width in the font family. Normally, reduced fonts are used in marginal notes. A number of typesetting systems can even squeeze fonts automatically to fit a given measure. A few typefaces have distinct designs with a variety of widths such as Helvetica and Helvetica-Narrow.

4.1.2.4 Size

On rendering (printing or displaying), a typeface must be represented by a given size. Font copies at precise sizes have to be produced and stored in order to be employed, depending on the technology used. Present technology allows the production of fonts regardless of the size, since typefaces are stored as contour descriptions.

Besides considering the general characteristics of a new typeface, the designer needs to choose the range of sizes the typeface is most likely to be placed in. The choice often

comes down to a division between small composition sizes (for newspaper classified advertisements), text composition sizes (8 pt to 14 pt covers newspaper, book, etc.) and display (conventionally, sizes above 14 pt, used to set individual words and headlines rather than continuous text) [Lun92]. Figure 4.4 illustrates some font weights and slopes and a variety of sizes available on the Macintosh system.

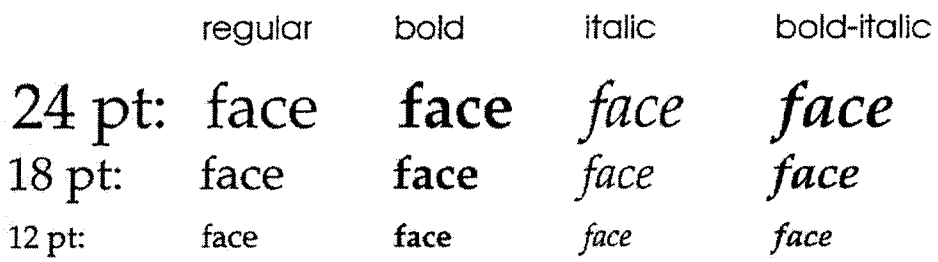


Figure 4.4. Various font weights, slopes and sizes of the Palatino typeface.

4.1.2.5 Other shapes

Several other shape variants may be present, for e.g., 'small caps' shape is in regular use, in which lowercase letters are characterized as capitals with reduced height. With 'outline' shapes the inner parts of the strokes are empty. For display purposes, 'shaded' shapes are presented where characters emerge as three-dimensional.

4.2 Typeface classification

Document style is conveyed through the page layout and characters inside. From centuries, page formatting has relied on current formats, justification modes, interline spacing and on characters used, where each element affects the others. Characters, in

relation to the epoch fashion and with the technological evolution (paper nature, printing process, etc.), have obtained proportions (wide or tight, thin or thick, etc.), modified the contrast of their strokes, sloped or straightened up their rounded letter axis and changed their serifs.

These features may be used as a starting point to character classification into families, which leads to their stylistic recognition. This categorization can be done from two points of view: the design point of view and the recognition point of view.

4.2.1 Classification from the design point of view

Font classification was roughly abandoned and did not follow the swift development of digital typography with a rapid production of new fonts. The taxonomy, done by experts in typography, remains manual and fundamentally based on historical features.

4.2.1.1 Thibaudeau's classification

The first effort in type classification was made by Francis Thibaudeau in the 1920's [DR85]. He detailed four classes according to the serif shapes: (1) the 'Elzevirs' with their triangular serifs, (2) the 'Didot' with upright serifs, (3) the Egyptiennes with rectangular serifs and (4) the Antique without serifs.

4.2.1.2 DIN's classification

The A.TYP.I (Association TYPographique Internationale) implemented the DIN's classification differentiating typeface classes according to a multitude of design criteria such as historical use, geographic or linguistic origin, direction of writing, the way a spoken language is written and various others. The classification standard basically addressed Latin typefaces and grouped them into nine groups [Kar94a].

This classification has its foundation in historical criteria and on delicate differences in shapes of special characters such as 'e' or small parts such as bowl axis, which are not simply visible to common users. This classification is now becoming outdated because it does not follow intended criteria to categorize new typefaces. In reality, with the great help from computers, a fast increase of Latin typefaces and the emergence of typefaces for other scripts such as Arabic, Cyrillic and Chinese, can be observed.

4.2.1.3 AFII's Classification

In the 1990's, the AFII (Association of Font Interchange International) projected a truly advanced classification system which was comprised of a typeface design grouping scheme, with three-level hierarchical structure. Typefaces which are alike in appearance or have features that would allow them to be replaced for each other, were clustered together.

This classification too considers various scripts such as Latin, Arabic, Chinese and Cyrillic. It has an open design permitting new fonts to be linked to one of the predefined classes. A few font designs could be linked to more than one group, but the responsibility lies in the hand of the font designer to choose the group which offers the best appearance for substitution. A detailed presentation of the AFII's classification model is provided in [Kar94a].

4.2.1.4 The PANOSE typeface classification system

The PANOSE classification system [Bau91] allocates an eight digit number to a typeface, which explains its principal visual characteristics. Each digit symbolizes a visual feature such as serif shape, contrast levels, x-height proportions and stroke variations.

Unlike other typeface classification systems, PANOSE focuses on physical measurements of the type, rather than on any subjective historical or artistic analysis. However, examination of this classification scheme has discovered that:

- Classification features allow distinguishing typefaces but not fonts with size and style specifications (regular, bold, roman, italic, etc.);
- Some features exist on only a limited set of characters such as 'g', 'Q' or 'O'.

All these classification systems reflect on typefaces from the design point of view where all features are 'a priori' known by the type classifier.

4.2.1.5 Typeface statistics

Peter Karow [Kar93] did a statistical study of about 1795 out of 3000 hand-digitized typefaces stored in the 'Ikarus' format [Kar94b]. The statistics were computed from measurements corresponding to some font metrics. From this set, 1049 fonts were seriffed and 485 sans-serif. The measurements have shown that (see Appendix A):

- Sans-serif fonts have bigger x-heights than seriffed ones but smaller ascenders and descenders;
- Italic characters have slightly bigger ascenders and descenders than upright ones;
- Sans-serif typefaces have a low contrast (ratio of horizontal to vertical stroke widths) of about 82%, while seriffed ones have high contrast with a ratio value of 50%.
- Seriffed glyphs are relatively wider than sans serif ones: 75% vs. 63% of the body size. This is mainly due to serifs which take up an important place in the character width;
- Sans-serif typefaces are simpler than seriffed ones concerning contour elements (corner points, straight lines, inflection points, curves, etc.). However, a sans-serif 'm' is always more complex than a seriffed 'l'.

In conclusion, it can be stated that these measurements can largely distinguish seriffed from sans-serif fonts.

4.2.2 Classification from the OFR point of view

From the OFR point of view, fonts may be classified according to criteria resulting from the following considerations:

- Fonts have to be identified from document images of diverse qualities. In fact, according to image conditions (resolution, noise, skew, etc.), some features of font design (e.g., the serif forms and the smoothness of character contours) may be lost;
- In reality, only a few well-known fonts are used in documents.

The typographical study has discovered seven criteria that can be used to distinguish fonts (some of them are detailed in [Rub88, pg.18]). They are either discrete or continuous:

1. Serifs: does the font have serifs or not? If serifs exist, what are their shapes? In reality, a rough differentiation between seriffed and sans-serif fonts may be sufficient for document analysis purposes;
2. Spacing: do letters have a fixed spacing or a proportional one?
3. Writing style: does the fonts have a typesetter or cursive style?
4. Slope: are characters upright or sloped?
5. Weight: what is the font boldness comparative to other fonts from the same family; is it light, regular, bold or black?
6. Size: what is the nominal font size?
7. x-height: what is the proportion of the x-height in the body size?

While some of these criteria permit the distinguishing of typefaces (serifs, writing, style, x-height), others are applicable to fonts (slope, weight, size).

4.3 Conclusion

None of the conventional typeface classification standards has resolved the problem from the recognition point of view. A few objective criteria that may be used to model typefaces and fonts (weight, x-height, serifs, etc.) have been discussed in this chapter. The majority of these criteria resulted from personal observations, from a typographical study and from the statistical analysis of typeface measurements made by Karow. In the presented model a font is recognized by its typeface, weight, width and size. Nearly all of the font distinguishing criteria can be extracted easily from document images. As a matter of fact, the current classifier system is based on the design of features simulating these criteria.

CHAPTER 5

BAYESIAN DECISION THEORY

5.1 Introduction

Bayesian decision theory is a fundamental statistical approach to the problem of pattern classification. This approach is based on quantifying the tradeoffs between various classification decisions using probability and the costs that accompany such decisions. It makes the assumption that the decision problem is posed in probabilistic terms, and that all of the relevant probability values are known. In this chapter, the fundamentals of this theory are detailed and it is shown how it can be viewed as being simply a formalization of common-sense procedures.

Although a quite general, abstract development of Bayesian decision theory is given in Section 2.2, but the discussion begins with a specific example. Consider the hypothetical problem of designing a classifier to separate two kinds of fish: sea bass and salmon. Suppose that an observer watching fish arrive along the conveyer belt finds it hard to predict what type will emerge next and that the sequence of types of fish appears to be random. In decision-theoretic terminology it can be said that as each fish emerges, nature is in one or the other of the two possible states: either the fish is a sea bass or the fish is a salmon. Let 'w' denote the 'state of nature', with '$w = w_1$' for sea bass and '$w = w_2$' for

salmon. Because the state of nature is so unpredictable, 'w' can be considered to be a variable that must be described probabilistically.

If the catch produced as much sea bass as salmon, it can be said that the next fish is equally likely to be sea bass or salmon. More generally, it is assumed that there is some 'a priori probability' (or simply prior) $P(w_1)$ that the next fish is sea bass, and some prior probability $P(w_2)$ that it is salmon. If it can be assumed that there are no other types of fish relevant here, then $P(w_1)$ and $P(w_2)$ sum to one. These prior probabilities reflect the prior knowledge of how likely is it to get a sea bass or salmon before the fish actually appears. It might, for instance, depend upon the time of year or the choice of fishing area.

Suppose for a moment that it is forced to make a decision about the type of fish that will appear next without being allowed to see it. For the moment, it should be assumed that any incorrect classification entails the same cost or consequence, and that the only information allowed to use is the value of the prior probabilities. If a decision must be made with so little information, it seems logical to use the following decision rule: Decide 'w_1' if $P(w_1) > P(w_2)$; otherwise decide 'w_2'.

This rule makes sense if just one fish needs to be judged, but if many fish need to be judged, using this rule repeatedly may seem a bit strange. After all, always the same decision would be made, even though it is known that both types of fish will appear. How well it works depends upon the values of the prior probabilities. If $P(w_1)$ is very much greater than $P(w_2)$, there is only a fifty-fifty chance of being right. In general, the

probability of error is the smaller of $P(w_1)$ and $P(w_2)$, and it will be seen later that under these conditions no other decision rule can yield a larger probability of being right.

In most circumstances it is not forced to make decisions with so little information. In this example, a lightness measurement 'x' might be used to improve the classifier. Different fish will yield different lightness readings, and this variability can be expressed in probabilistic terms. Consider 'x' to be a continuous random variable whose distribution depends on the state of nature and is expressed as p(x | w). This is the 'class-conditional probability density' function, the probability density function for 'x' given that the state of nature is 'w'. It is also sometimes called state-conditional probability density. Then the difference between $p(x \mid w_1)$ and $p(x \mid w_2)$ describes the difference in lightness between populations of sea bass and salmon (Figure 5.1).

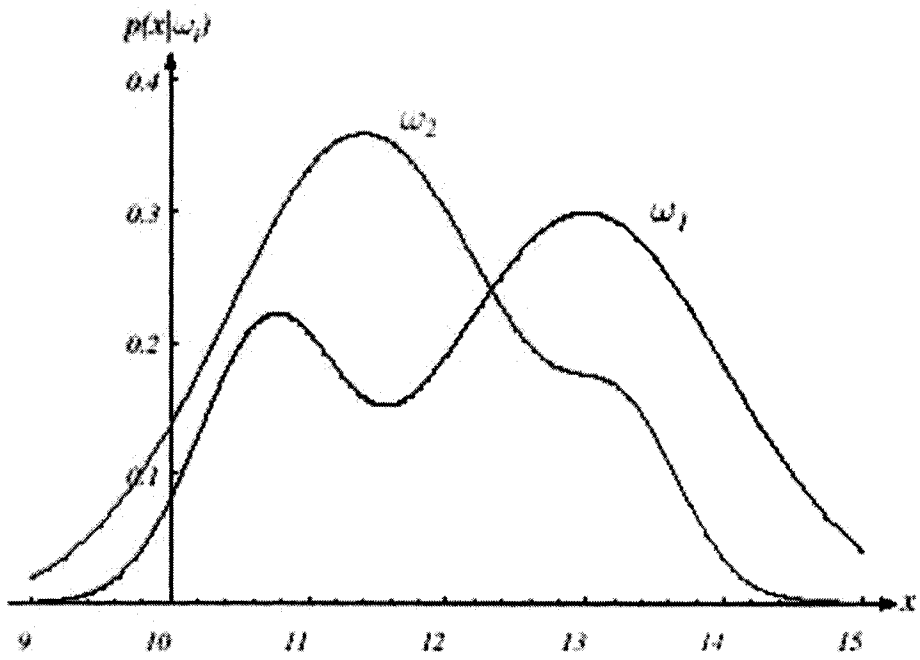


Figure 5.1. Hypothetical class-conditional probability density functions show the probability density of measuring a particular feature value 'x' given the pattern is

in category w_i. If 'x' represents the lightness of a fish, the two curves might describe the difference in lightness of populations of two types of fish. Density functions are normalized, and thus the area under each curve is 1.0.

Suppose that both the prior probabilities $P(w_j)$ and the conditional densities $p(x \mid w_j)$ for j = 1,2, are known. Suppose further that the lightness of a fish is measured and that its value is found to be 'x'. How does this measurement influence the attitude concerning the true state of nature, i.e., the category of the fish? It is noted first that the (joint) probability density of finding a pattern, that is in category 'w_j' and has feature value 'x', can be written in two ways: $p(w_j, x) = P(w_j \mid x)\, p(x) = p(x \mid w_j)\, P(w_j)$. Rearranging these leads to the answer to the question, which is called Bayes formula:

$$P(w_j \mid x) = \frac{p(x \mid w_j)P(w_j)}{p(x)} \tag{1}$$

where in this case of two categories,

$$p(x) = \sum_{j=1}^{2} p(x \mid wj)\, P(wj)$$

Bayes formula can be expressed informally in English by saying that,

$$\text{posterior} = \frac{\text{likelihood} * \text{prior}}{\text{evidence}}$$

Bayes formula shows that by observing the value of 'x', the prior probability $P(w_j)$ can be converted to the 'a posteriori' probability (or posterior) $P(w_j \mid x)$; the probability of the state of nature being w_j given that feature value 'x' has been measured. $p(x \mid w_j)$ is called the 'likelihood' of 'w_j' with respect to 'x', a term chosen to indicate that, other things being equal, the category 'w_j' for which $p(x \mid w_j)$ is large is more 'likely' to be the true

category. Notice that it is the product of the likelihood and the prior probability that is most important in determining the posterior probability; the 'evidence' factor, p(x), can be viewed as merely a scale factor that guarantees that the posterior probabilities sum to one, as all good probabilities must. The variation of $P(w_j \mid x)$ with 'x' is illustrated in Figure 5.2 for the case $P(w_1) = 2/3$ and $P(w_2) = 1/3$.

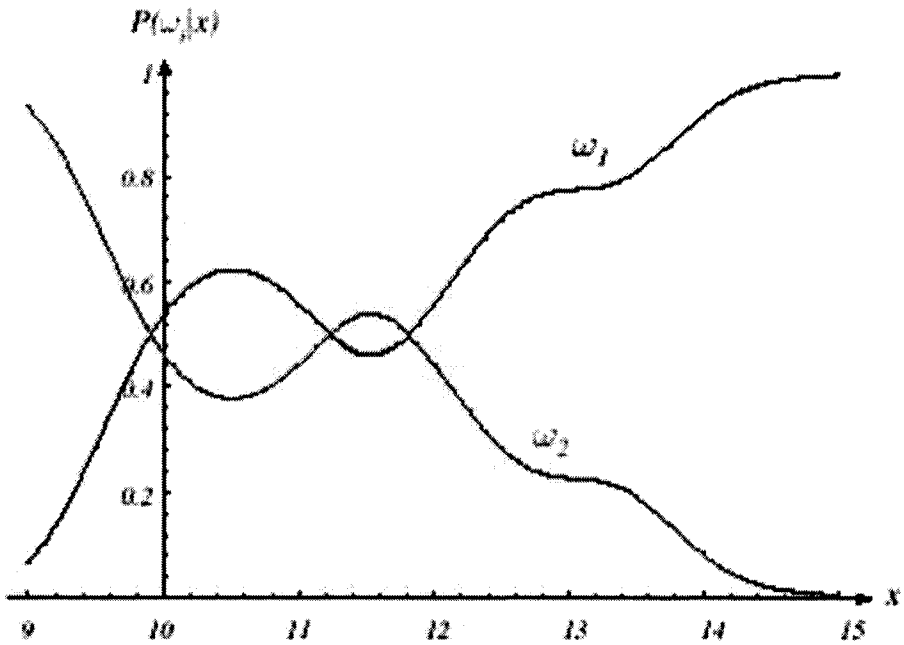


Figure 5.2. Posterior probabilities for the particular priors $P(w_1) = 2/3$ and $P(w_2) = 1/3$ for the class-conditional probability densities shown in Figure 5.1. Thus, in this case, given that a pattern is measured to have feature value x = 14, the probability it is in category w_2 is roughly 0.08, and that it is in w_1 is 0.92. At every 'x', the posteriors sum to 1.0.

Consider an observation 'x' for which $P(w_1 \mid x)$ is greater than $P(w_2 \mid x)$, it will naturally be decided that the true state of nature is 'w_1'. Conversely, if $P(w_2 \mid x)$ is greater than

$P(w_1 \mid x)$, decision will go in favor of w_2. To justify this decision procedure, calculate the probability of error whenever a decision is made. Whenever a particular 'x' is observed, the probability of error is given by,

$$P(\text{error} \mid x) = \begin{cases} P(w_1 \mid x) \text{ if } w_2 \text{ is decided} \\ P(w_2 \mid x) \text{ if } w_1 \text{ is decided} \end{cases} \quad (2)$$

Clearly, for a given 'x', the probability of error can be minimized by deciding 'w_1' if $P(w_1 \mid x) > P(w_2 \mid x)$ and 'w_2' otherwise. Of course, exactly the same value of 'x' may never be observed twice. Will this rule minimize the average probability of error? Yes, because the average probability of error is given by,

$$P(\text{error}) = \int_{-\infty}^{\infty} P(error, x)dx = \int_{-\infty}^{\infty} P(error \mid x)p(x)dx$$

and if for every 'x' it is ensured that $P(\text{error} \mid x)$ is as small as possible, then the integral must be as small as possible. Thus, the following 'Bayes decision rule' for minimizing the probability of error has been justified:

Decide w_1 if $P(w_1 \mid x) > P(w_2 \mid x)$; otherwise decide w_2

Under this rule Eq. 2 becomes,

$$P(\text{error} \mid x) = \min [P(w_1 \mid x), P(w_2 \mid x)]$$

This form of the decision rule emphasizes the role of the posterior probabilities. By using Eq. 1, the rule can instead be expressed in terms of the conditional and prior probabilities. First note that the 'evidence', $p(x)$, in Eq. 1 is unimportant as far as making a decision is concerned. It is basically just a scale factor that states how frequently a pattern with feature value 'x' will actually be measured; as mentioned earlier, its presence in Eq. 1 assures that $P(w_1 \mid x) + P(w_2 \mid x) = 1$. By eliminating this scale factor, the following completely equivalent decision rule is obtained:

Decide w_1 if $p(x \mid w_1) P(w_1) > p(x \mid w_2) P(w_2)$; otherwise decide w_2.

Some additional insight can be obtained by considering a few special cases. If for some 'x', $p(x \mid w_1) = p(x \mid w_2)$, then that particular observation gives no information about the state of nature; in this case, the decision hinges entirely on the prior probabilities. On the other hand, if $P(w_1) = P(w_2)$, then the states of nature are equally probable; in this case the decision is based entirely on the likelihoods $p(x \mid w_j)$. In general, both of these factors are important in making a decision, and the 'Bayes decision rule' combines them to achieve the minimum probability of error.

5.2 Bayesian decision theory - continuous features

The ideas considered earlier will be formalized in this section, and will be generalized in four different ways:

- By allowing the use of more than one feature.
- By allowing more than two states of nature.
- By allowing actions other than merely deciding the state of nature.
- By introducing a loss function more general than the probability of error.

These generalizations and their attendant notational complexities should not obscure the central points illustrated in the simple example. Allowing the use of more than one feature merely requires replacing the scalar 'x' by the 'feature vector: **x**', where '**x**' is in a d-dimensional Euclidean space R^d, called the 'feature space'. Allowing more than two states of nature provides with a useful generalization for a small notational expense.

Allowing actions other than classification primarily allows the possibility of rejection, i.e., of refusing to make a decision in close cases; this is a useful option if being indecisive is not too costly. Formally, the 'loss function' states exactly how costly each action is, and is used to convert a probability determination into a decision. Cost functions relate to situations in which some kinds of classification mistakes are costlier than others, although the simplest case, where all errors are equally costly, is often discussed. With this as an introduction, the more formal treatment is discussed ahead.

Let $\{w_1,.....,w_c\}$ be the finite set of 'c' states of nature (categories) and let $\{\alpha_1,.....,\alpha_a\}$ be the finite set of 'a' possible actions. The loss function $\lambda(\alpha_i \mid w_j)$ describes the loss incurred for taking action α_i when the state of nature is w_j. Let the feature vector $\mathbf{x}$ be a d-component vector-valued random variable and let $p(\mathbf{x} \mid w_j)$ be the state-conditional probability density function for '**x**', with the probability density function for '**x**' conditioned on w_j being the true state of nature. As before, $P(w_j)$ describes the prior probability that nature is in state w_j. Then the posterior probability $P(w_j \mid \mathbf{x})$ can be computed from $p(\mathbf{x} \mid w_j)$ by Bayes formula:

$$P(w_j \mid \mathbf{x}) = \frac{p(x \mid wj)P(wj)}{p(x)}$$

where the evidence is now,

$$p(\mathbf{x}) = \sum_{j=1}^{c} p(x \mid wj)P(wj)$$

Suppose that a particular '**x**' is observed and that taking action α_i is contemplated. If the true state of nature is w_j, by definition the loss $\lambda(\alpha_i \mid w_j)$ will be incurred. Because $P(w_j \mid$

x) is the probability that the true state of nature is w_j, the expected loss associated with taking action α_i is merely,

$$R(\alpha_i \mid \mathbf{x}) = \sum_{j=1}^{c} \lambda(\alpha_i \mid w_j)\, P(w_j \mid \mathbf{x})$$

In decision-theoretic terminology, an expected loss is called a 'risk', and $R(\alpha_i \mid \mathbf{x})$ is called the 'conditional risk'. Whenever a particular observation '**x**' is encountered, the expected loss can be minimized, by selecting the action that minimizes the conditional risk. It can now be shown that this 'Bayes decision procedure' actually provides the optimal performance.

Stated formally, the problem is to find a decision rule against $P(w_j)$ that minimizes the overall risk. A general 'decision rule' is a function $\alpha(\mathbf{x})$ that decides which action to take for every possible observation. To be more specific, for every '**x**' the 'decision function' $\alpha(\mathbf{x})$ assumes one of the 'a' values: $\alpha_1,\ldots..,\alpha_a$. The overall risk 'R' is the expected loss associated with a given decision rule. Because $R(\alpha_i \mid \mathbf{x})$ is the action, the overall risk is given by,

$$R = \int R(\alpha(x) \mid x) p(x) dx$$

where d**x** is the notation for a d-space volume element and where the integral extends over the entire feature space. Clearly, if $\alpha(\mathbf{x})$ is chosen so that $R(\alpha_i \mid \mathbf{x})$ is as small as possible for every '**x**', then the overall risk will be minimized. This justifies the following statement of the 'Bayes decision rule': To minimize the overall risk, compute the conditional risk,

$$R(\alpha_i \mid \mathbf{x}) = \sum_{j=1}^{c} \lambda(\alpha_i \mid w_j)\, P(w_j \mid \mathbf{x}) \qquad (3)$$

for i = 1,….,a and then select the action α_i for which $R(\alpha_i \mid \mathbf{x})$ is minimum. The resulting minimum overall risk is called the 'Bayes risk', denoted R^*, and is the best performance that can be achieved.

5.2.1 Two-category classification

Consider these results when applied to the special case of two-category classification problems. Here action α_i corresponds to deciding that the true state of nature is w_1, and action α_2 corresponds to deciding that it is w_2. For notational simplicity, let $\lambda_{ij} = \lambda(\alpha_i \mid w_j)$ be the loss incurred for deciding w_i when the true state of nature is w_j. If the conditional risk is stated, given by Eq.3, then the following equations are obtained:

$$R(\alpha_1 \mid \mathbf{x}) = \lambda_{11}\, P(w_1 \mid \mathbf{x}) + \lambda_{12}\, P(w_2 \mid \mathbf{x})$$

$$R(\alpha_2 \mid \mathbf{x}) = \lambda_{21}\, P(w_1 \mid \mathbf{x}) + \lambda_{22}\, P(w_2 \mid \mathbf{x})$$

There are a variety of ways of expressing the minimum-risk decision rule, each having its own minor advantages. The fundamental rule is to decide w_1 if $R(\alpha_1 \mid \mathbf{x}) < R(\alpha_2 \mid \mathbf{x})$. In terms of the posterior probabilities, w_1 is decided if,

$$(\lambda_{21} - \lambda_{11})\, P(w_1 \mid \mathbf{x}) > (\lambda_{12} - \lambda_{22})\, P(w_2 \mid \mathbf{x})$$

Ordinarily, the loss incurred for making an error is greater than the loss incurred for being correct, and both of the factors $(\lambda_{21} - \lambda_{11})$ and $(\lambda_{12} - \lambda_{22})$ are positive. Thus in practice, the decision is generally determined by the more likely state of nature, although the posterior probabilities must be scaled by the loss differences. By employing Bayes formula, the

posterior probabilities can be replaced by the prior probabilities and the conditional densities. This results in the equivalent rule, to decide w_1 if,

$$(\lambda_{21} - \lambda_{11})\, p(\mathbf{x} \mid w_1)\, P(w_1) > (\lambda_{12} - \lambda_{22})\, p(\mathbf{x} \mid w_2)\, P(w_2)$$

and otherwise decide w_2.

Another alternative, which follows at once under the reasonable assumption that $\lambda_{21} > \lambda_{11}$, is to decide w_1 if,

$$\frac{p(x \mid w_1)}{p(x \mid w_2)} > \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}} \frac{P(w_2)}{P(w_1)}$$

This form of the decision rule focuses on the **x**-dependence of the probability densities. $p(\mathbf{x} \mid w_j)$ can be considered a function of w_j (i.e., the likelihood function) and then the 'likelihood ratio' $p(x \mid w_1) / p(x \mid w_2)$ can be formed. Thus the Bayes decision rule can be interpreted as calling for deciding w_1 if the likelihood ratio exceeds a threshold value that is independent of the observation '**x**'.

5.3 Classifiers, discriminant functions, and decision surfaces

5.3.1 The multicategory case

There are many different ways to represent pattern classifiers. One of the most useful is in terms of a set of 'discriminant functions' $g_i(x)$, i = 1,….,c. The classifier is said to assign a feature vector '**x**' to class w_i if,

$$g_i(x) > g_j(x) \qquad \text{for all } j \neq i$$

Thus, the classifier is viewed as a network or machine that computes 'c' discriminant functions and selects the category corresponding to the largest discriminant. A network representation of a classifier is illustrated in Figure 5.3.

A Bayes classifier is easily and naturally represented in this way. For this general case with risks, let $g_i(x) = -R(\alpha_i \mid x)$, because the maximum discriminant function will then correspond to the minimum conditional risk. For the minimum-error-rate case, things can be simplified further by taking $g_i(x) = P(w_i \mid x)$, so that the maximum discriminant function corresponds to the maximum posterior probability.

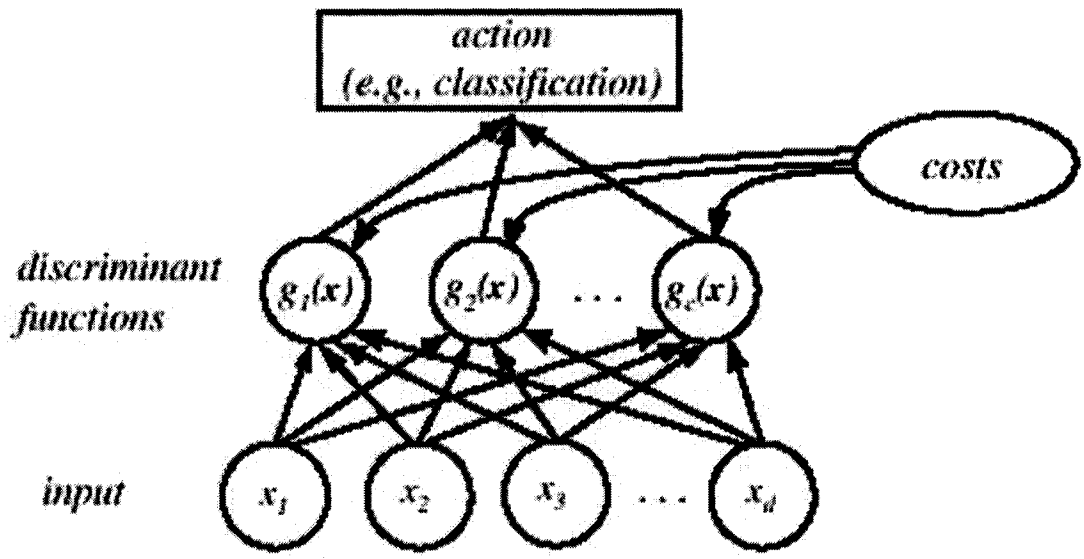


Figure 5.3. The functional structure of a general statistical pattern classifier which includes 'd' inputs and 'c' discriminant functions $g_i(x)$. A subsequent step determines which of the discriminant values is the maximum, and categorizes the input pattern accordingly. The arrows show the direction of the flow of

information, though frequently the arrows are omitted when the direction of flow is self-evident.

Clearly, the choice of discriminant functions is not unique. All the discriminant functions can always be multiplied by the same positive constant or can be shifted by the same additive constant without influencing the decision. More generally, if every $g_i(\mathbf{x})$ is replaced by $f(g_i(\mathbf{x}))$, where f(.) is a monotonically increasing function, the resulting classification is unchanged. This observation can lead to significant analytical and computational simplifications. In particular, for minimum-error-rate classification, any of the following choices gives identical classification results, but, some can be much simpler to understand or to compute than others:

$$g_i(x) = P(w_i \mid x) = \frac{p(x \mid w_i)P(w_i)}{\sum_{j=1}^{c} p(x \mid w_i)P(w_j)} \tag{4}$$

$$g_i(x) = p(x \mid w_i)P(w_i) \tag{5}$$

$$g_i(x) = \ln p(x \mid w_i) + \ln P(w_i) \tag{6}$$

where ln denotes natural logarithm.

Even though the discriminant functions can be written in a variety of forms, the decision rules are equivalent. The effect of any decision rule is to divide the feature space into 'c' decision regions, $R_1,\ldots,R_c$. If $g_i(x) > g_j(x)$ for all $j \neq i$, then '$\mathbf{x}$' is in R_i, and the decision rule calls to assign '$\mathbf{x}$' to w_i. The regions are separated by 'decision boundaries',

surfaces in feature space where ties occur among the largest discriminant functions (Figure 5.4).

5.3.2 The two-category case

While the two-category case is just a special instance of the multicategory case, it has traditionally received separate treatment. Indeed, a classifier that places a pattern in one of only two categories has a special name; a 'dichotomizer'. Instead of using two discriminant functions g_1 and g_2 and assigning '**x**' to w_1 if $g_1 > g_2$, it is more common to define a single discriminant function,

$$g(x) \equiv g_1(x) - g_2(x)$$

and to use the following decision rule: Decide w_1 if $g(\mathbf{x}) > 0$; otherwise decide w_2. Thus, a dichotomizer can be viewed as a machine that computes a single discriminant function $g(\mathbf{x})$, and classifies '**x**' according to the algebraic sign of the result. Of the various forms in which the minimum-error-rate discriminant function can be written, the following two (derived from Eqs. 4 and 6) are particularly convenient:

$$g(x) = P(w_1 \mid x) - P(w_2 \mid x)$$

$$g(x) = \ln \frac{p(x \mid w_1)}{p(x \mid w_2)} + \ln \frac{P(w_1)}{P(w_2)} \tag{7}$$

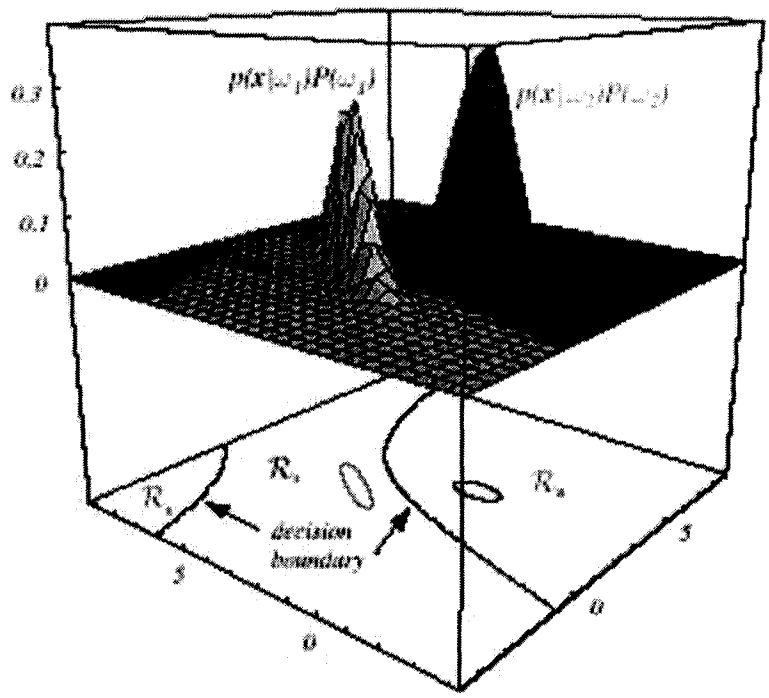


Figure 5.4. In this two-dimensional two-category classifier, the probability densities are Gaussian, the decision boundary consists of two hyperbolas, and thus the decision region R_2 is not simply connected. The ellipses mark where the density is 1/e times that at the peak of the distribution.

5.4 The normal density

The structure of a Bayes classifier is determined by the conditional densities $p(\mathbf{x} \mid w_i)$ as well as by the prior probabilities $P(w_i)$. Of the various density functions that have been investigated, none has received more attention than the multivariate normal or Gaussian density. To a large extent this attention is due to its analytical tractability. However, the multivariate normal density is also an appropriate model for an important situation, namely, the case where the feature vectors '**x**' for a given class w_i are continuous-valued,

randomly corrupted versions of a single typical or prototype vector $\boldsymbol{\mu}_i$. In this section a brief exposition of the multivariate normal density is provided, focusing on the properties of greatest interest for classification problems.

First, recall the definition of the 'expected value' of a scalar function f(x), defined for some density p(x):

$$\in [f(x)] \equiv \int_{-\infty}^{\infty} f(x)p(x)dx$$

If the values of the feature 'x' are restricted to points in a discrete set D, summation must be done over all samples as,

$$\in [f(x)] = \sum_{x \in D} f(x)P(x)$$

where P(x) is the probability mass at 'x'. Calculations of expected values by these and analogous equations defined in higher dimensions, may occasionally be needed.

5.4.1 Univariate density

The continuous univariate normal or Gaussian density is defined as,

$$\mathrm{p(x)} = \frac{1}{\sqrt{2\lambda}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma} \right)^2 \right] \tag{8}$$

for which the 'expected value' of 'x' (an average, here taken over the feature space) is,

$$\mu \equiv \in [x] = \int_{-\infty}^{\infty} xp(x)dx$$

and where the expected squared deviation or 'variance' is,

$$\sigma^2 \equiv \in \left[(x-\mu)^2 \right] = \int_{-\infty}^{\infty} (x-\mu)^2 p(x)dx$$

The univariate normal density is completely specified by two parameters: its mean 'μ' and variance σ^2. For simplicity, Eq.8 is often abbreviated by writing $p(x) \approx N(\mu, \sigma^2)$, to say that 'x' is distributed normally with mean 'μ' and variance 'σ^2'. Samples from normal distributions tend to cluster about the mean, with a spread related to the standard deviation 'σ' (Figure 5.5).

There is a deep relationship between the normal distribution and 'entropy'. The entropy of a distribution is given by,

$$H(p(x)) = -\int p(x) \ln p(x) dx$$

and measured in 'nats'; If a $\log_2$ is used instead, the unit is the 'bit'. The entropy measures the fundamental uncertainty in the values of points selected randomly from a distribution. It can be shown that the normal distribution has the maximum entropy of all distributions having a given mean and variance. Moreover, as stated by the 'Central Limit Theorem', the aggregate effect of the sum of a large number of small, independent random disturbances will lead to a Gaussian distribution. Because many patterns; from fish to handwritten characters to some speech sounds, can be viewed as some ideal or prototype pattern corrupted by a large number of random processes, the Gaussian is often a good model for the actual probability distribution.

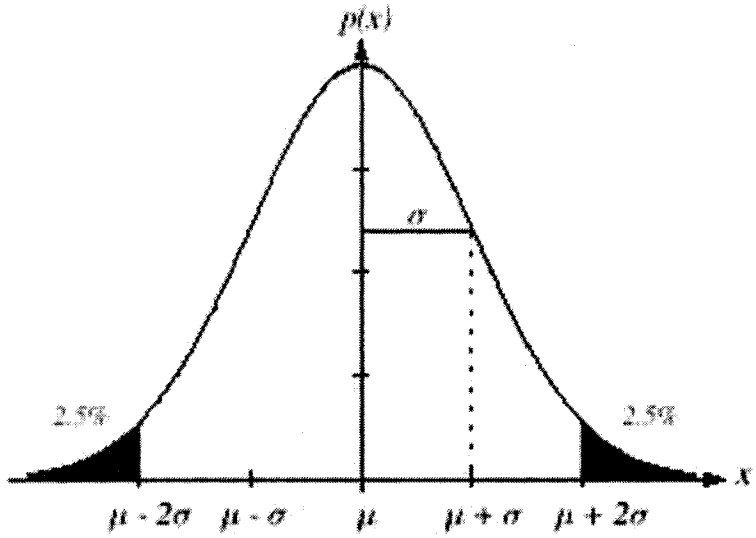


Figure 5.5. A univariate normal distribution has roughly 95% of its area in the range $|x - \mu| \leq 2\sigma$, as shown. The peak of the distribution has value $p(\mu) = 1/\sqrt{2\lambda}\,\sigma$.

5.4.2 Multivariate density

The general multivariate normal density in 'd' dimensions is written as,

$$p(x) = \frac{1}{(2\lambda)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^t \Sigma^{-1}(x-\mu)\right] \qquad (9)$$

where '**x**' is a d-component column vector, '**μ**' is the d-component mean vector, 'Σ' is the d-by-d covariance matrix, and $|\Sigma|$ and Σ^{-1} are its determinant and inverse, respectively. Also, $(\mathbf{x} - \boldsymbol{\mu})^t$ denotes the transpose of $(\mathbf{x} - \boldsymbol{\mu})$.

Formally,

$$\mu \equiv \in [x] = \int xp(x)dx$$

and,

$$\sum \equiv \in \left[(x-\mu)(x-\mu)^t\right] = \int (x-\mu)(x-\mu)^t p(x)dx$$

where the expected value of a vector or a matrix is found by taking the expected values of its components. In other words, if x_i is the i^{th} component of '**x**', μ_i the i^{th} component of '**μ**', and σ_{ij} the ij^{th} component of '$\sum$', then,

$$\mu_i = \in [x_i]$$

and,

$$\sigma_{ij} = \in [(x_i - \mu_i)(x_j - \mu_j)]$$

The covariance matrix '$\sum$' is always symmetric and positive semidefinite. Attention will be restricted to the case in which '$\sum$' is positive definite, so that the determinant of '$\sum$' is strictly positive. The diagonal elements σ_{ii} are the variances of the respective x_i (i.e., σ_i^2), and the off-diagonal elements σ_{ij} are the covariances of x_i and x_j. A positive covariance for the length and weight features of a population of fish would be expected, for instance. If x_i and x_j are statistically independent, then $\sigma_{ij} = 0$. If all the off-diagonal elements are zero, p(**x**) reduces to the product of the univariate normal densities for the components of '**x**'.

The multivariate normal density is completely specified by d + d(d+1)/2 parameters, namely the elements of the mean vector '**μ**' and the independent elements of the covariance matrix '$\sum$'. Samples drawn from a normal population tend to fall in a single

cloud or cluster (Figure 5.6); the center of the cluster is determined by the mean vector, and the shape of the cluster is determined by the covariance matrix. It follows from Eq.9 that the loci of points of constant density are hyperellipsoids for which the quadratic form $(x-\mu)^t \Sigma^{-1}(x-\mu)$ is constant. The principal axes of these hyperellipsoids are given by the eigenvectors of 'Σ' (described by Φ); the eigenvalues determine the lengths of these axes. The quantity,

$$r^2 = (x-\mu)^t \Sigma^{-1}(x-\mu)$$

is sometimes called the squared 'Mahalanobis distance' from '**x**' to '**μ**'. Thus, the contours of constant density are hyperellipsoids of constant Mahalanobis distance to '**μ**' and the volume of these hyperellipsoids measures the scatter of the samples about the mean. It can be shown that the volume of the hyperellipsoid corresponding to a Mahalanobis distance 'r' is given by,

$$V = V_d |\Sigma|^{1/2} r^d$$

where V_d is the volume of a d-dimensional unit hypersphere:

$$V_d = \begin{cases} \pi^{d/2}/(d/2) & \text{d even} \\ 2^d \pi^{(d-1)/2}\left(\frac{d-1}{2}\right)!/d! & \text{d odd} \end{cases}$$

Thus, for a given dimensionality, the scatter of the samples varies directly with $|\Sigma|^{1/2}$.

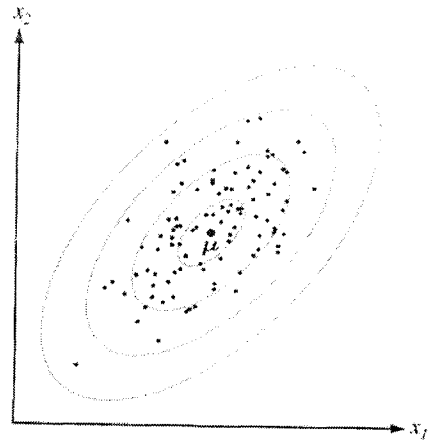


Figure 5.6. Samples drawn from a two-dimensional Gaussian lie in a cloud centered on the mean μ. The ellipses show lines of equal probability density of the Gaussian.

5.5 Discriminant functions for the normal density

The minimum-error-rate classification can be achieved by use of the discriminant functions,

$$g_i(x) = \ln p(x \mid w_i) + \ln P(w_i)$$

This expression can be readily evaluated if the densities $p(\mathbf{x} \mid w_i)$ are multivariate normal; i.e., if $p(\mathbf{x} \mid w_i) \approx N(\boldsymbol{\mu}_i, \Sigma_i)$. In this case, then from Eq.9,

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^t \Sigma_i^{-1}(x-\mu_i) - \frac{d}{2}\ln 2\lambda - \frac{1}{2}\ln|\Sigma_i| + \ln P(w_i) \qquad (10)$$

This discriminant function and resulting classification for a number of special cases are examined ahead.

5.5.1 Case 1: $\sum_i = \sigma^2 I$

The simplest case occurs when the features are statistically independent and when each feature has the same variance, σ^2. In this case the covariance matrix is diagonal, being merely σ^2 times the identity matrix **I**. Geometrically, this corresponds to the situation in which the samples fall in equal-size hyperspherical clusters, the cluster for the i^{th} class being centered about the mean vector $\boldsymbol{\mu_i}$. The computation of the determinant and the inverse of $\sum_i$ is particularly easy: $|\Sigma_i| = \sigma^{2d}$ and $\Sigma_i^{-1} = (1/\sigma^2)I$. Because both $|\sum_i|$ and the (d/2)ln2π term in Eq.10 are independent of 'i', they are unimportant additive constants that can be ignored. Thus the simple discriminant functions are obtained as,

$$g_i(x) = -\frac{\|x-\mu_i\|^2}{2\sigma^2} + \ln P(w_i) \qquad (11)$$

where || . || denotes the 'Euclidean norm', i.e.,

$$\|x-\mu_i\|^2 = (x-\mu_i)^t(x-\mu_i)$$

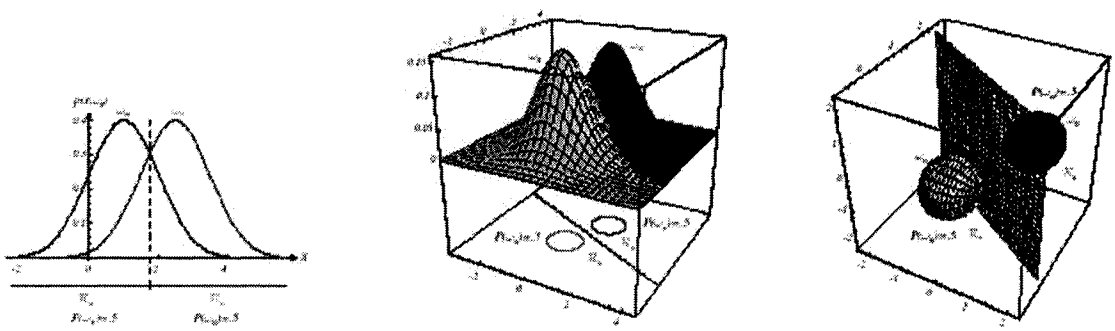


Figure 5.7. If the covariance matrices for two distributions are equal and proportional to the identity matrix, then the distributions are spherical in 'd' dimensions, and the boundary is a generalized hyperplane of d - 1 dimensions,

perpendicular to the line separating the mean. In these one-, two-, and three-dimensional examples, $p(x \mid w_i)$ and the boundaries for the case $P(w_1) = P(w_2)$ are indicated. In the three-dimensional case, the grid plane separates R_1 from R_2.

If the prior probabilities are not equal, then Eq.11 shows that the squared distance $\|\mathbf{x} - \boldsymbol{\mu}\|^2$ must be normalized by the variance σ^2 and offset by adding $\ln P(w_i)$; thus, if '**x**' is equally near two different mean vectors, the optimal decision will favor the 'a priori' more likely category.

Regardless of whether the prior probabilities are equal or not, it is not actually necessary to compute distances. Expansion of the quadratic form $(x-\mu_i)^t(x-\mu_i)$ yields,

$$g_i(x) = -\frac{1}{2\sigma^2}\left[x^t x - 2\mu_i^t x + \mu_i^t \mu_i\right] + \ln P(w_i)$$

which appears to be a quadratic function of '**x**'. However, the quadratic term $\mathbf{x}^t\mathbf{x}$ is the same for all 'i', making it an ignorable additive constant. Thus, the equivalent 'linear discriminant functions' are obtained as,

$$g_i(x) = w_i^t x + w_{i0}$$

where,

$$w_i = \frac{1}{\sigma^2}\mu_i$$

and,

$$w_{i0} = \frac{-1}{2\sigma^2}\mu_i^t \mu_i + \ln P(w_i)$$

w_{i0} is called the 'threshold' or 'bias' for the i^{th} category.

A classifier that uses linear discriminant functions is called a 'linear machine'. This kind of classifier has many interesting theoretical properties. The decision surfaces for a linear machine are pieces of hyperplanes defined by the linear equations $g_i(x) = g_j(x)$ for the two categories with the highest posterior probabilities. For this particular case, the equation can be written as,

$$w^t(x - x_0) = 0$$

where,

$$w = \mu_i - \mu_j$$

and,

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\sigma^2}{\|\mu_i - \mu_j\|^2} \ln \frac{P(w_i)}{P(w_j)} (\mu_i - \mu_j) \qquad (12)$$

These equations define a hyperplane through the point $\mathbf{x_0}$ and orthogonal to the vector '**w**'. Because $\mathbf{w} = \boldsymbol{\mu}_i - \boldsymbol{\mu}_j$, the hyperplane separating R_i and R_j is orthogonal to the line linking the means. If $P(w_i) = P(w_j)$, the second term on the right of Eq.12 vanishes, and thus the point $\mathbf{x_0}$ is halfway between the means, and the hyperplane is the perpendicular bisector of the line between the means (Figure 5.8). If $P(w_i) \neq P(w_j)$, the point $\mathbf{x_0}$ shifts away from the more likely mean. Note, however, that if the variance σ^2 is small relative to the squared distance $\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2$, then the position of the decision boundary is relatively insensitive to the exact values of the prior probabilities.

If the prior probabilities $P(w_i)$ are the same for all 'c' classes, then the $\ln P(w_i)$ term becomes another unimportant additive constant that can be ignored. When this happens, the optimum decision rule can be stated very simply: To classify a feature vector '**x**',

measure the Euclidean distance $||\mathbf{x} - \boldsymbol{\mu}_i||$ from each '**x**' to each of the 'c' mean vectors, and assign '**x**' to the category of the nearest mean. Such a classifier is called a 'minimum-distance classifier'. If each mean vector is thought of as being an ideal prototype or template for patterns in its class, then this is essentially a template-matching procedure (Figure 5.7).

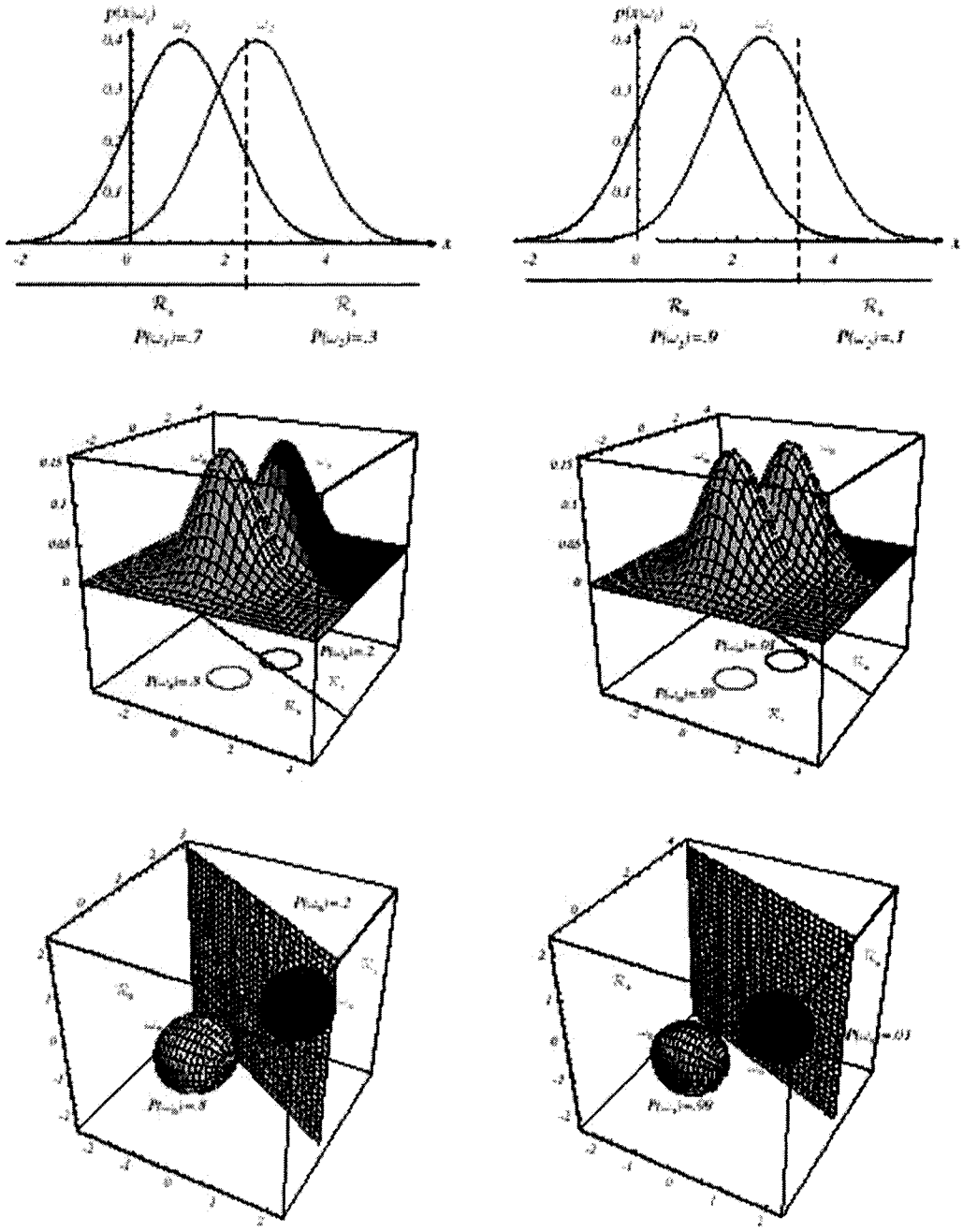


Figure 5.8. As the priors are changed, the decision boundary shifts; for sufficiently disparate priors the boundary will not lie between the means of these one-, two-, and three-dimensional spherical Gaussian distributions.

5.5.2 Case 2: $\sum_i = \sum$

Another simple case arises when the covariance matrices for all of the classes are identical but otherwise arbitrary. Geometrically, this corresponds to the situation in which the samples fall in hyperellipsoidal clusters of equal size and shape, the cluster for the i^{th} class being centered about the mean vector $\boldsymbol{\mu}_i$. Because both $|\sum_i|$ and the $(d/2)\ln 2\pi$ term in Eq.10 are independent of 'i', they can be ignored as superfluous additive constants. This simplification leads to the discriminant functions,

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^t \sum_i^{-1}(x-\mu_i) + \ln P(w_i) \qquad (13)$$

If the prior probabilities $P(w_i)$ are the same for all 'c' classes, then the $\ln P(w_i)$ term can be ignored. In this case, the optimal decision rule can once again be stated very simply: To classify a feature vector '**x**', measure the squared Mahalanobis distance from '**x**' to each of the 'c' mean vectors, and assign '**x**' to the category of the nearest mean. As before, unequal prior probabilities bias the decision in favor of the 'a priori' more likely category.

Expansion of the quadratic form $(x-\mu_i)^t \Sigma_i^{-1}(x-\mu_i)$ results in a sum involving a quadratic term $x^t \Sigma^{-1} x$ which here is independent of 'i'. After this term is dropped from Eq.13, the resulting discriminant functions are again linear:

$$g_i(x) = w_i^t x + w_{i0}$$

where,

$$w_i = \Sigma^{-1} \mu_i$$

and,

$$w_{i0} = -\frac{1}{2} \mu_i^t \Sigma^{-1} \mu_i + \ln P(w_i)$$

Because the discriminants are linear, the resulting decision boundaries are again hyperplanes (Figure 5.7). If R_i and R_j are contiguous, the boundary between them has the equation,

$$w^t(x - x_0) = 0$$

where,

$$w = \Sigma^{-1}(\mu_i - \mu_j)$$

and,

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\ln[P(w_i)/P(w_j)]}{(\mu_i - \mu_j)^t \Sigma^{-1}(\mu_i - \mu_j)}(\mu_i - \mu_j)$$

Because $w = \Sigma^{-1}(\mu_i - \mu_j)$ is generally not in the direction of $\mu_i - \mu_j$, the hyperplane separating R_i and R_j is generally not orthogonal to the line between the means. However, it does intersect that line at the point $\mathbf{x_0}$; if the prior probabilities are equal, then $\mathbf{x_0}$ is halfway between the means. If the prior probabilities are not equal, the optimal boundary

hyperplane is shifted away from the more likely mean (Figure 5.9). As before, with sufficient bias the decision plane need not lie between the two mean vectors.

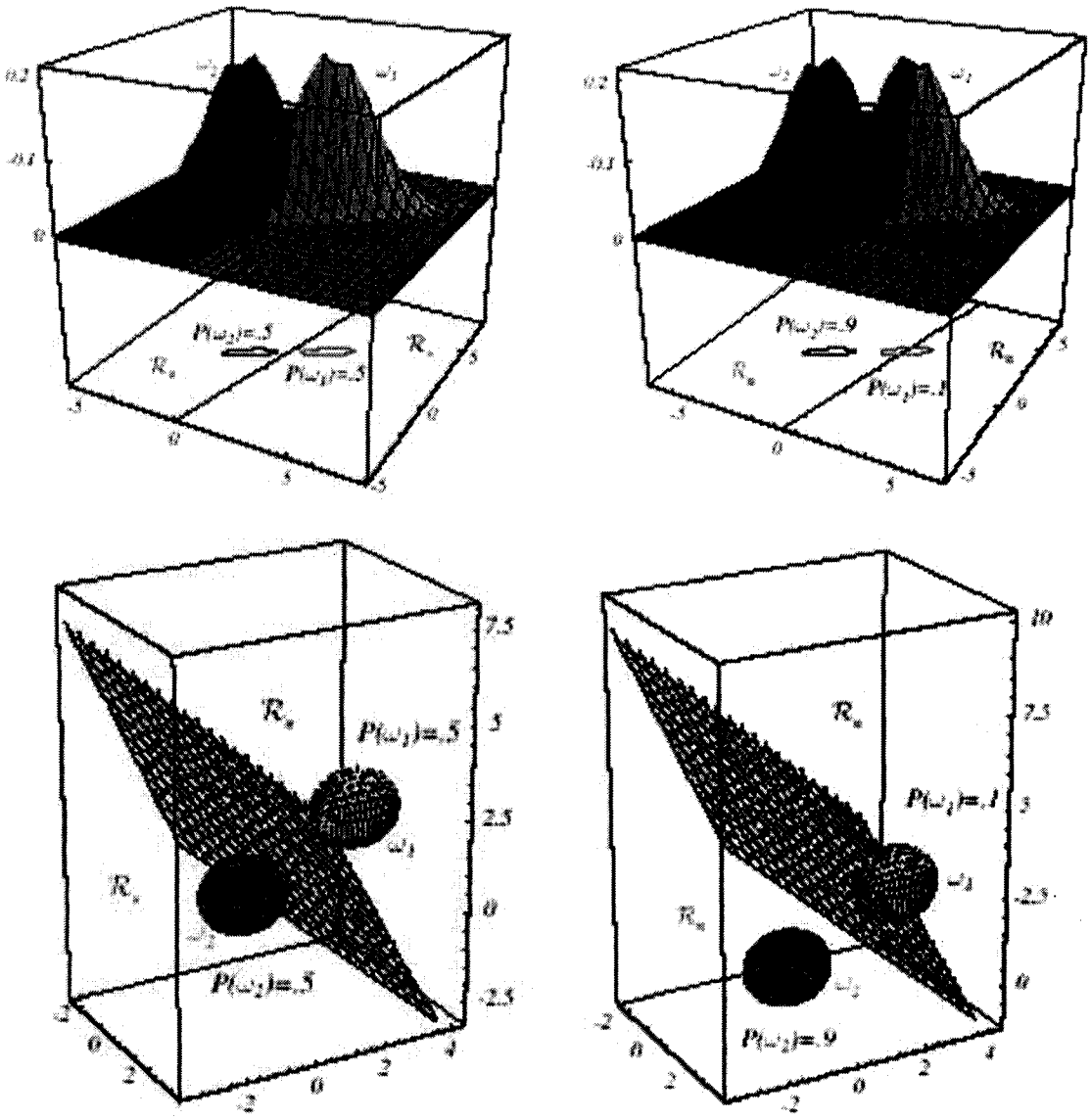


Figure 5.9. Probability densities (indicated by the surfaces in two dimensions and ellipsoidal surfaces in three dimensions) and decision regions for equal but asymmetric Gaussian distributions. The decision hyperplanes need not be perpendicular to the line connecting the means.

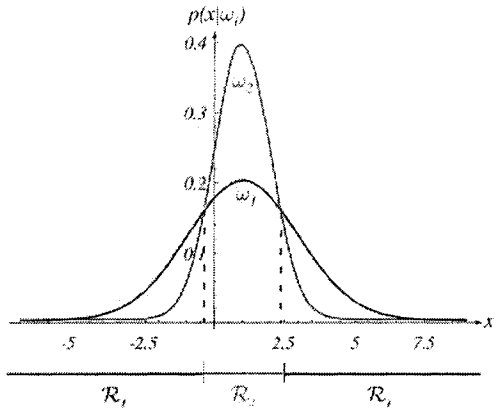


Figure 5.10. Non-simply connected decision regions can arise in one dimensions for Gaussians having unequal variance, as shown in this case with $P(w_1) = P(w_2)$.

5.5.3 Case 3: $\sum_i$ = arbitrary

In the general multivariate normal case, the covariance matrices are different for each category. The only term that can be dropped from Eq.10 is the $(d/2)\ln 2\pi$ term, and the resulting discriminant functions are inherently quadratic:

$$g_i(x) = x^t W_i x + w_i^t x + w_{i0}$$

where,

$$W_i = -\frac{1}{2}\Sigma_i^{-1}$$

$$w_i = \Sigma_i^{-1}\mu_i$$

and,

$$w_{i0} = -\frac{1}{2}\mu_i^t \Sigma_i^{-1}\mu_i - \frac{1}{2}\ln|\Sigma_i| + \ln P(w_i)$$

In the two-category case, the decision surfaces are hyperquadrics, and they can assume any of the general forms: hyperplanes, pairs of hyperplanes, hyperspheres, hyperellipsoids, hyperparaboloids, and hyperhyperboloids of various types. Even in one dimension, for arbitrary variance the decision regions need not be simply connected (Figure 5.10). The two- and three-dimensional examples in Figures 5.11 and 5.12 indicate how these different forms can arise.

The extension of these results to more than two categories is straightforward though here it needs to be clear which two of the total 'c' categories are responsible for any boundary segment. Figure 5.13 shows the decision surfaces for a four-category case made up of Gaussian distributions. Of course, if the distributions are more complicated, the decision regions can be even more complex, though the same underlying theory holds there too.

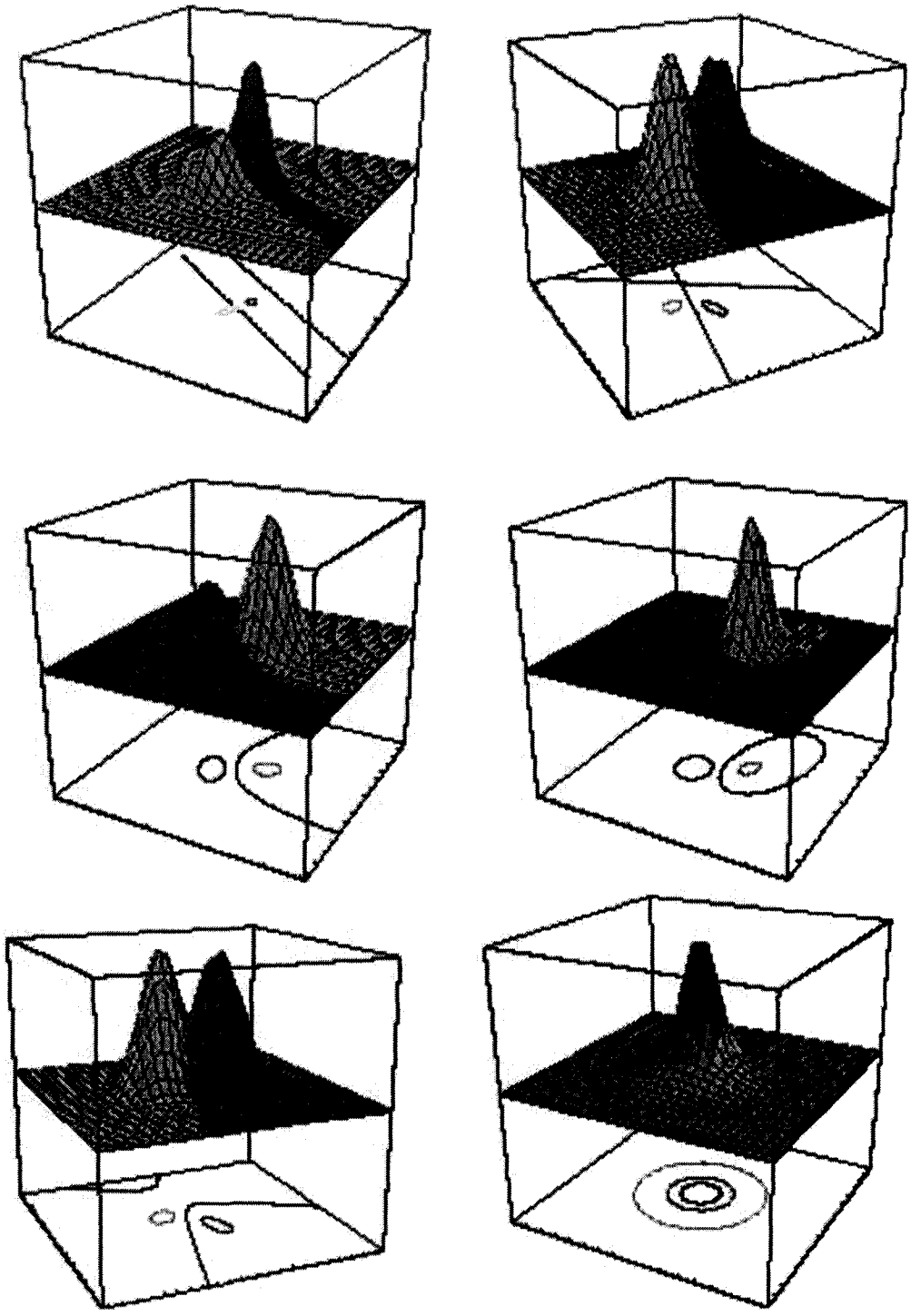


Figure 5.11. Arbitrary Gaussian distributions lead to Bayes decision boundaries that are general hyperquadrics. Conversely, given any hyperquadric, one can find

two Gaussian distributions whose Bayes decision boundary is that hyperquadric. These variances are indicated by the contours of constant probability density.

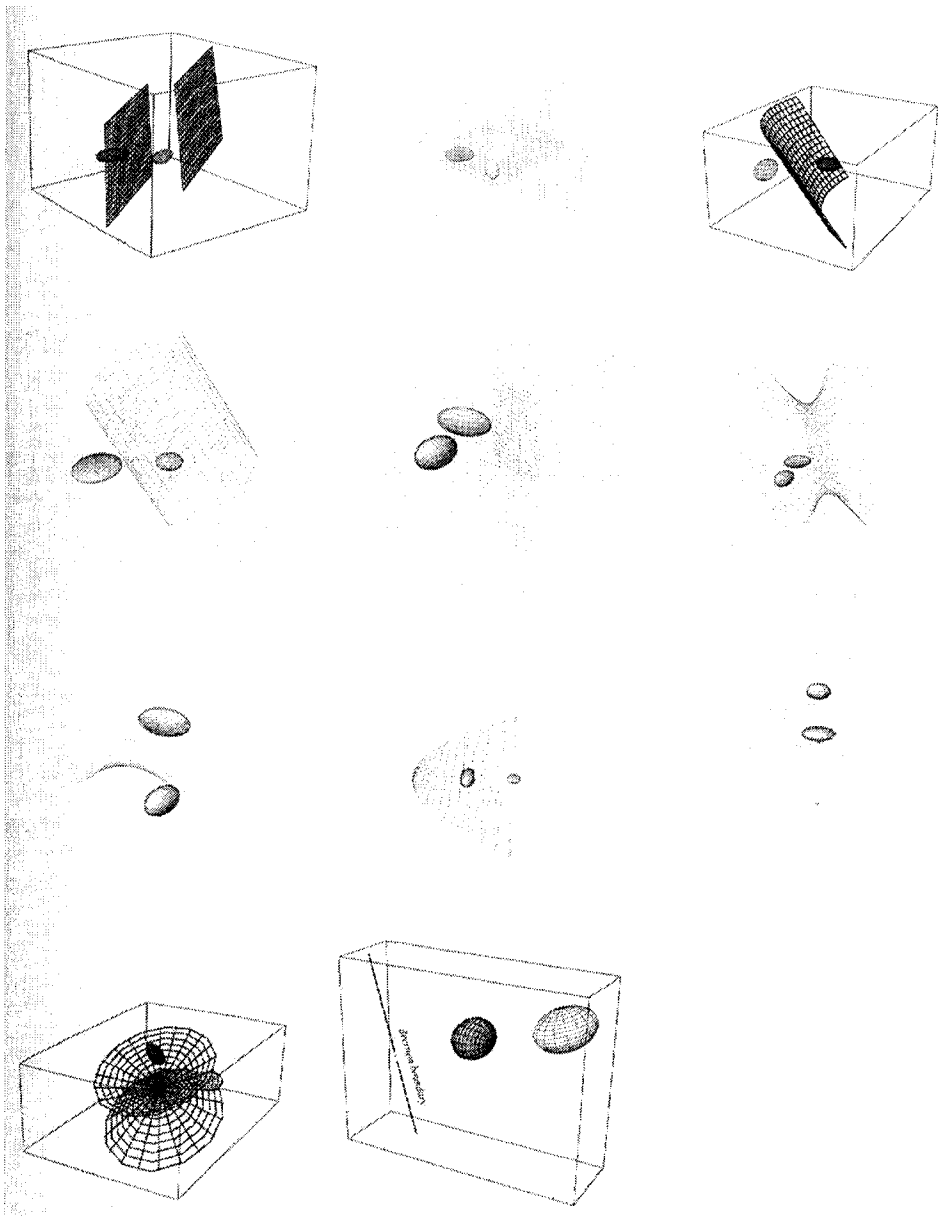


Figure 5.12. Arbitrary three-dimensional Gaussian distributions yield Bayes decision boundaries that are two-dimensional hyperquadrics. These are even degenerate cases in which the decision boundary is a line.

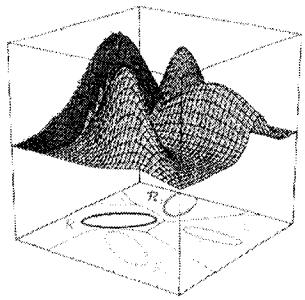


Figure 5.13. The decision regions for four normal distributions. Even with such a low number of categories, the shapes of the boundary regions can be rather complex.

5.6 Bayes decision theory - discrete features

Until now it has been assumed that the feature vector '**x**' could be any point in a d-dimensional Euclidean space, $\mathbf{R}^d$. However, in many practical applications the components of '**x**' are binary-, ternary-, or higher-integer-valued, so that '**x**' can assume only one of 'm' discrete values $\mathbf{v}_1,\ldots,\mathbf{v}_m$. In such cases, the probability density function $p(\mathbf{x} \mid w_j)$ becomes singular; integrals of the form,

$$\int p(x \mid w_j)dx$$

must then be replaced by corresponding sums, such as,

$$\sum_x P(x \mid w_j)$$

where it is understood that the summation is over all values of '**x**' in the discrete distribution. Bayes formula then involves probabilities, rather than probability densities:

$$P(w_j \mid x) = \frac{P(x \mid w_j)P(w_j)}{P(x)}$$

where,

$$P(x) = \sum_{j=1}^{c} P(x \mid w_j) P(w_j)$$

The definition of the conditional risk R(α | **x**) is unchanged, and the fundamental Bayes decision rule remains the same: To minimize the overall risk, select the action α_i for which R(α_i | **x**) is minimum, or stated formally,

$$\alpha^* = \arg\min_i R(\alpha \mid x)$$

The basic rule to minimize the error rate by maximizing the posterior probability is also unchanged as are the discriminant functions of Eqs.4-6, given the obvious replacement of densities p(.) by probabilities P(.).

5.6.1 Independent binary features

As an example of a classification involving discrete features, consider the two-category problem in which the components of the feature vector are binary-valued and conditionally independent. To be more specific, let **x** = $(x_1,\ldots,x_d)^t$, where the components x_i are either 0 or 1, with probabilities,

$$p_i = \Pr[x_i = 1 \mid w_1]$$

and,

$$q_i = \Pr[x_i = 1 \mid w_2]$$

This is a model of a classification problem in which each feature gives a yes/no answer about the pattern. If $p_i > q_i$, the ith feature is expected to give a 'yes' answer more frequently when the state of nature is w_i than when it is w_2. By assuming conditional

independence $P(\mathbf{x} \mid w_i)$ can be written as the product of the probabilities for the components of '**x**'. Given this assumption, a particularly convenient way of writing the class-conditional probabilities is as follows:

$$P(x \mid w_1) = \prod_{i=1}^{d} p_i^{x_i} (1 - p_i)^{1-x_i}$$

and,

$$P(x \mid w_2) = \prod_{i=1}^{d} q_i^{x_i} (1 - q_i)^{1-x_i}$$

Then the likelihood ratio is given by,

$$\frac{P(x \mid w_1)}{P(x \mid w_2)} = \prod_{i=1}^{d} \left(\frac{p_i}{q_i} \right)^{x_i} \left(\frac{1 - p_i}{1 - q_i} \right)^{1-x_i}$$

and consequently Eq.7 yields the discriminant function,

$$g(x) = \sum_{i=1}^{d} \left[x_i \ln \frac{p_i}{q_i} + (1 - x_i) \ln \frac{1 - p_i}{1 - q_i} \right] + \ln \frac{P(w_1)}{P(w_2)}$$

It can be noted that this discriminant function is linear in the x_i and thus,

$$g_i(x) = \sum_{i=1}^{d} w_i x_i + w_0$$

where,

$$w_i = \ln \frac{p_i (1 - q_i)}{q_i (1 - p_i)} \qquad i = 1, \ldots, d$$

and,

$$w_0 = \sum_{i=1}^{d} \ln \frac{1 - p_i}{1 - q_i} + \ln \frac{P(w_1)}{P(w_2)}$$

These results can be examined to see what insight they can give. Recall first that w_1 is decided if $g(\mathbf{x}) > 0$ and w_2 if $g(\mathbf{x}) \leq 0$. It has been observed that $g(\mathbf{x})$ is a weighted

combination of the components of '**x**'. The magnitude of the weight w_i indicates the relevance of a 'yes' answer for x_i in determining the classification. If $p_i = q_i$, x_i gives no information about the state of nature, and $w_i = 0$, just as might be expected. If $p_i > q_i$, then $1 - p_i < 1 - q_i$ and w_i is positive. Thus in this case a 'yes' answer for x_i contributes w_i votes for w_1. Furthermore, for any fixed $q_i < 1$, w_i gets larger as p_i gets larger. On the other hand, if $p_i < q_i$, w_i is negative and a 'yes' answer contributes $|w_i|$ votes for w_2.

The condition of feature independence leads to a very simple (linear) classifier; of course if the features were not independent, a more complicated classifier would be needed. To conclude, the more independent the features are, the simpler the classifier can be.

The prior probabilities $P(w_i)$ appear in the discriminant only through the threshold weight w_0. Increasing $P(w_i)$ increases w_0 and biases the decision in favor of w_1, whereas decreasing $P(w_1)$ has the opposite effect. Geometrically, the possible values for '**x**' appear as the vertices of a d-dimensional hypercube; the decision surface defined by $g(\mathbf{x}) = 0$ is a hyperplane that separates w_1 vertices from w_2 vertices.

5.7 Conclusion

The basic ideas underlying Bayes decision theory are very simple. To minimize the overall risk, the action that minimizes the conditional risk $R(\alpha \mid \mathbf{x})$ should always be chosen. In particular, to minimize the probability of error in a classification problem, the state of nature that maximizes the posterior probability $P(w_j \mid \mathbf{x})$ should always be chosen.

Bayes formula allows to calculate such probabilities from the prior probabilities $P(w_j)$ and the conditional densities $p(\mathbf{x} \mid w_j)$. If there are different penalties for misclassifying patterns from w_i as if from w_j, the posteriors must be first weighted according to such penalties before taking action.

If the underlying distributions are multivariate Gaussian, the decision boundaries will be hyperquadrics, whose form and position depends upon the prior probabilities, means and covariances of the distributions in question. For many pattern classification applications, the chief problem in applying these results is that the conditional densities $p(\mathbf{x} \mid w_j)$ are not known. In some cases the form these densities assume may be known, but characterizing parameter values may not be known. The classic case occurs when the densities are known to be, or can assumed to be, multivariate normal, but the values of the mean vectors and the covariance matrices are not known. More commonly, even less is known about the conditional densities, and procedures that are less sensitive to specific assumptions about the densities must be used.

CHAPTER 6

MODEL APPROACH

6.1 Classifier System

The following steps were taken in order to build the classifier system:

1. Repositioning of image by centering it within a 16 x 16 pixel array
2. Extraction of eight 'Central Moment' features and 'Covariance' of image (see Section 5.4.2)
3. Development of Six Classifiers
    a. Minimum Distance Moment Classifier (Bayes Moment Classifier with Identity Covariance Matrices) (see Section 5.5.1)
    b. Bayes Moment Classifier with Identical Covariances (see Section 5.5.2)
    c. Bayes Moment Classifier with Individual Class Covariances (see Section 5.5.3)
    d. Bayes Moment Classifier with Individual Class Covariances (Using first four moments only) (see Section 5.5.3)
    e. Minimum Distance Classifier in Binary Pixel Space (see Section 5.5.1)
    f. Bayes Classifier in Binary Pixel Space (see Section 5.6.1)

4. Training of the six classifiers on 'A' set of 100 samples belonging to six databases namely, Lowercase characters, Uppercase characters, Digits, Punctuation, Typefaces and Image Qualities.
5. Testing of the six classifiers on sets 'B', 'C' and 'D' of 100 samples each belonging to six databases namely, Lowercase characters, Uppercase characters, Digits, Punctuation, Typefaces and Image Qualities.

6.2 Performance Evaluation

The following is a summary of the results derived from training the six classifiers on set 'A' and testing on sets 'B', 'C' and 'D', of the six databases namely, Lowercase characters, Uppercase characters, Digits, Punctuation, Typefaces and Image Qualities.

Lowercase						
			Test Set			
Classifier	A	B	C	D	AvgError	%AvgError
1	712	721	694	712	709.75	70.98
2	711	732	696	711	712.50	71.25
3	377	396	377	377	381.75	38.18
4	507	534	527	507	518.75	51.88
5	35	35	37	35	35.50	3.55
6	38	37	32	38	36.25	3.63

Table 6.1. Lowercase Character Classes.

Uppercase						
			Test Set			
Classifier	A	B	C	D	AvgError	%AvgError
1	708	707	731	693	709.75	70.98
2	707	711	734	700	713.00	71.30
3	358	371	391	345	366.25	36.63
4	471	448	454	436	452.25	45.23
5	37	29	42	31	34.75	3.48
6	40	32	42	39	38.25	3.83

Table 6.2. Uppercase Character Classes.

Digit						
			Test Set			
Classifier	A	B	C	D	AvgError	%AvgError
1	772	763	775	737	761.75	76.18
2	771	761	776	741	762.25	76.23
3	517	398	497	451	465.75	46.58
4	589	494	557	558	549.50	54.95
5	20	20	15	19	18.50	1.85
6	22	25	25	24	24.00	2.40

Table 6.3. Digit Classes.

Punctuation						
			Test Set			
Classifier	A	B	C	D	AvgError	%AvgError
1	696	674	720	679	692.25	69.23
2	694	674	714	661	685.75	68.58
3	519	503	547	418	496.75	49.68
4	561	513	626	457	539.25	53.93
5	149	150	155	168	155.50	15.55
6	156	164	168	170	164.50	16.45

Table 6.4. Punctuation Classes.

Typeface						
			Test Set			
Classifier	A	B	C	D	AvgError	%AvgError
1	630	713	671	717	682.75	68.28
2	645	717	681	722	691.25	69.13
3	352	447	373	384	389.00	38.90
4	448	553	479	503	495.75	49.58
5	100	130	103	107	110.00	11.00
6	91	109	88	94	95.50	9.55

Table 6.5. Typeface Classes.

Image Quality						
			Test Set			
Classifier	A	B	C	D	AvgError	%AvgError
1	692	667	687	696	685.50	68.55
2	700	675	691	712	694.50	69.45
3	407	406	375	419	401.75	40.18
4	511	462	462	450	471.25	47.13
5	118	125	101	92	109.00	10.90
6	91	79	71	76	79.25	7.93

Table 6.6. Image Quality Classes.

Summary							
			Database				
Classifier	Lower Case	Upper Case	Digit	Punct uation	Type face	Image Quality	%Avg Error
1	70.98	70.98	76.18	69.23	68.28	68.55	70.70
2	71.25	71.30	76.23	68.58	69.13	69.45	70.99
3	38.18	36.63	46.58	49.68	38.90	40.18	41.69
4	51.88	45.23	54.95	53.93	49.58	47.13	50.45
5	3.55	3.48	1.85	15.55	11.00	10.90	7.72
6	3.63	3.83	2.40	16.45	9.55	7.93	7.30

Table 6.7. Summary of performance evaluation of the classifier system.

6.3 Conclusion

Classifiers 1, 2, 3 and 4 dealt with the 'Central Moment' features (8 features) of an image while Classifiers 5 and 6 dealt with the 'Pixel Frequency' and 'Posterior Probability' features (256 features) respectively of an image.

Classifiers 1 and 2 failed on all six databases since the decision boundaries formed were linear and hence the error rate got too high. Classifiers 3 and 4 performed comparatively well since the decision boundaries formed were quadratic and hence the error rate stayed stable. Classifier 4 failed on 'Lowercase characters', 'Digits' and 'Punctuation', and in all six databases performed lower than Classifier 3, since only four moments (reduction in features) were used as compared to eight in Classifier 3. Classifiers 5 and 6 performed the best among all classifiers regardless of the database they were tested on, since they used 256 features (increment in features), although the decision boundaries formed were linear.

REFERENCES

[AH92] J. Andre and R. D. Hersch. Teaching digital typography. *Electronic Publishing: Origination, Dissemination and Design*, 5(2):79-90, 6 1992.

[AIDG95] H. I. Avi-Itzhak, T. A. Diep, and H. Garland. High accuracy optical character recognition using neural networks with centroid dithering. *TPAMI*, 17(2):218-224, 32 1995.

[AK93] O. E. Agazzi and S-S Kuo. Hidden markov model based optical character recognition in the presence of deterministic transformations. *Pattern Recognition*, 26(12): 1813-1826, 1996.

[And93] J. Andre. Font Metrics. In R. D. Hersch, editor, *Visual and Technical Aspect of Type*, pages 64-77. Cambridge University Press, 1993.

[Ani92] Julien Ch. Anigbogu. *Reconnaissance de Textes Imprimes Multifontes a laide de Modeles Stochastiques et Metriques*. PhD thesis, Universite de Nancy I, 1992.

[Azo95] Antoine Azokly. *Une approche uniforme de la reconnaissance des structures physiques de documents composites fondee sur l'analyse des espaces*. PhD thesis, University of Fribourg, 1995.

[Bai96] H. S. Baird. Applications of multi-dimensional search to structural feature identification. In *Proceedings of the Workshop on Syntactical and Structural Pattern Recognition*, pages 1-13, Sitges, 10 1986.

[Bai93] H. S. Baird. Recognition technology frontiers. *Pattern Recognition Letters*, 14(14):327-334, 4 1993.

[Bau91] B. Bauermeister. *Manual of Comparative Typography: The PANOSE System.* VNR Company, New York, 1991.

[BB92] A. Belaid and Y. Belaid. *Reconnaissance de formes, Methodes et applications.* InterEdition, 1992.

[BBI95] F. Bapst, R. Brugger, and R. Ingold. Towards an interactive document structure recognition system. Technical report, IIUF, University of Fribourg, Switzerland, 3 1995.

[BBZI96] F. Bapst, R. Brugger, A. Zramdini, and R. Ingold. L'integration de donneees dans un syst'eme de reconnaissance de documents assistee. In *submitted to CNED'96*, Nante, France, 1996.

[BCG+93] J. L. Blue, G. T. Candela, P. J. Grother, R. Chelappa, and C. L. Wilson. Evaluation of pattern classifiers for fingerprint and OCR applications. Technical report, National Institute of Standards and Technology, Gaithersburg, MD 20899, 1993.

[BF91] H. S. Baird and R. Fossey. A 100-font classifier. In *Proceedings of the First International Conference on Document Analysis and Recognition*, pages 332-340, Saint-Malo, France, 9 1991.

[BHN92] T. Bayer, J. Hull, and G. Nagy. Character Recognition: SSPR'90 working group report. In H. S. Baird, H. Bunke, and K. Yamamoto, editors, *Structured Document Image Analysis*, page 567. Springer Verlag, 1992.

[BK88] G. Baptista and K. M. Kulkarni. A high accuracy algorithm for recognition of handwritten numerals. *PR*, 21(4):287-291, 1988.

[BN94] H. S. Baird and G. Nagy. A self-correcting 100-font classifier. In *SPIE-The international Society for Optical Engineering, Document Recognition*, pages 106-115, San Jose, California, 2 1994.

[CCP91] P. Campigli, V. Cappellini, and M. T. Pareschi. A reading system for printed documents. In *Proceedings of the First International Conference on Document Analysis and Recognition*, pages 585-593, Saint-Malo, France, September 1991.

[Che93] Y. Chenevoy. *Reconnaissance Structurelle de Documents Imprimes: Etudes et Realisations*. PhD thesis, CRIN-University of Nancy, France, 1993.

[CI94] J. L. Cochard and R. Ingold. Le project escroc: Environment de saisie et de correction de la reconnaissance optique de caracteres. *Actes de CNED'94: Traitement de l'Ecriture et des Documents*, pages 257-264, 1994.

[Col91] D. Collier. *Collier's Rules for Desktop design and Typography*. Addison-Wesley Publishing Company, 1991.

[Com90] Apple Computer. *TrueType Spec - The TrueType Font Format Specification*. Apple Computer, 1990.

[CP93] F. Cuneo and F. D. Pellet. Reconnaissance optique de caracteres. *MacInfo*, 10 1993.

[Cun89] Y. Le Cun. Backpropagation applied handwritten zip code recognition. *Neural Computation I*, pages 541-551, 1989.

[CW90] R. G. Casey and K. Y. Wong. Image analysis applications. In *Chapter 1*, pages 1-36. Marcel Dekker Inc., 1990.

[Dam64] F. J. Damerau. A technique for computer detection and correction of spelling errors. *Communication of the ACM*, 7(3): 171-176, 1964.

[Das88] B. V. Dasarathy, editor. *Nearest Neighbor (NN) Norms: NN Pattern-Classification Techniques*. IEEE Computer Society Press, 1988.

[Den90] A. Dengel. A step towards understanding paper documents. Technical report, The German Search Centre in Artificial Intelligence, 1990.

[DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1973.

[DHS2001] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2001.

[Doe93] David S. Doermann. *Document Image Understanding: Integrating Recovery and Interpretation*. PhD thesis, University of Maryland, 1993.

[DR85] J. Dreyfus and F. Richaudeau. *La chose imprimee: histoire, technique eshethique et realisation de l'imprimerie*. Editions Retz, 1985.

[DSV91] A. D'Acierno, C. De Stefano, and M. Vento. A structural character recognition method using neural networks. In *Proceedings of the First International Conference on Document Analysis and Recognition*, pages 803-811, Saint-Malo, France, September 1991.

[Egl94] H. Eglowstein. Due recognition for OCR. *Byte*, pages 145-148, 10 1994.

[Fur89] R. Furuta. Concepts and models for structured documents. In J. Andre, R. Furuta, and V. Quint, editors, *Structured Documents*, pages 7-38. Cambridge University Press, 1989.

[Gos91] B. Gosselin. Boolean neural network applied to the recognition of typographic characters. In *Proceedings of the First International Conference on Document Analysis and Recognition*, pages 426-434, Saint-Malo, France, September 1991.

[Her93a] R. D. Hersch. Font rasterization: the state of the art. In R. D. Hersch, editor, *Visual and Technical Aspects of Type*, pages 78-109. Cambridge University Press, 1993.

[Her93b] Roger Hersch, editor. *Visual and Technical Aspects of Type*. Cambridge University Press, 1993.

[HHS91] T. K. Ho, J. J. Hull, and S. N. Srihari. Word recognition with multi-level contextual knowledge. In *Proceedings of the First International Conference on Document Analysis and Recognition*, pages 905-915, Saint-Malo, France, September 1991.

[Hu94] Tao Hu. *New Methods for Robust and Efficient Recognition of the Logical Structures in Documents*. PhD thesis, University of Fribourg, 1994.

[Ing89] Rolf Ingold. *Une Nouvelle Approche de la Lecture Optique Integrant la Reconnaissance des Structures de Documents*. PhD thesis, Ecole Polytechnique Federale de Lausanne, Switzerland, 1989.

[Jol89] V. Joloboff. Document representation: Concepts and standards. In J. Andre, R. Furuta, and V. Quint, editors, *Structured Documents*, pages 75-105. Cambridge University Press, 1989.

[JSB91] M. A. Jones, G. A. Story, and B. W. Ballard. Integrating multiple knowledge sources in a Bayesian OCR post-processor. In *Proceedings of the First International Conference on Document Analysis and Recognition*, pages 925-933, Saint-Malo, France, September 1991.

[Kar93] Peter Karow. *Typeface Statistics*. URW Verlag, Hambourg, 1993.

[Kar94a] Peter Karow. *Digital Typefaces*. URW Verlag, Hambourg, 1994.

[Kar94b] Peter Karow. *Font Technology*. URW Verlag, Hambourg, 1994.

[KNRN93] J. Kanai, T. A. Nartker, S. V. Rice, and G. Nagy. Performance metrics for document understanding systems. In *ICDAR'93: Second International Conference on*

Document Analysis and Recognition, pages 424-427, Tsukuba Science City, Japan, 10 1993.

[Knu93] D. E. Knuth. *Computer Modern Typefaces*. Addison Wesley Publishing, 1993.

[Kop93] G. E. Kopec. Least-square font metric estimation from images. *IEEE Transactions on Image Processing*, 2(4):510-519, 10 1993.

[KPB87] S. Kahan, T. Pavlidis, and H. S. Baird. On the recognition of printed characters of any font and size. *TPAMI*, 9(2):274-288, 3 1987.

[Leb91] F. Lebourgeois. *Approche mixte pour la reconnaissance des documents imprimes*. PhD thesis, Institut National des Sciences Appliquees de Lyon, France, 1991.

[LEB94] R. W. De Lange, H. L. Esterhuizen, and D. Beatty. Performance differences between Times and Helvetica in a reading task. In RIDT'94: *Third International Conference on Raster Imaging and Digital Typography*, pages 232-240, Darmstadt, Germany, 4 1994.

[LSA89] S. Liang, M. Shridhar, and M. Ahmadi. Segmentation of touching characters in printed document recognition. *PR*, 22(4):347-350, 1989.

[Lu93] Y. Lu. On the segmentation of touching characters. In *ICDAR'93: Second International Conference on Document Analysis and Recognition*, pages 440-447, Tsukuba Science City, Japan, 10 1993.

[Lun92] P. Luna. *Understanding type for desktop publishing*. BluePrint, Chapman & Hall, 1992.

[LZ94] D. Lopersi and J. Zhou. Using consensus sequence voting to correct OCR errors. In *Proceedings of the IAPR Workshop on Document Analysis Systems*, pages 191-202, Kaiserslautern, Germany, October 1994.

[MBH93] R. A. Morris, K. Berry, and K. Hargreaves. Towards quantification of the effects of typographic variation on readability. Technical report, Department of Mathematics and Computer Science, University of Massachusetts at Boston, 1993.

[MBHL91] R. A. Morris, K. Berry, K. Hargreaves, and D. Liarakopis. How typeface variation and typographic scaling affect readability at small sizes. In *IS&T's Seventh International Congress on Advances in Non-Impact Printing Technologiestrieval*, pages 6-11, Portland, Oregon, 10 1991.

[Mor92] R. A. Morris. Classification of digital typefaces using spectral signatures. *Pattern Recognition*, 25(8): 869-876, 1992.

[Nag82] G. Nagy. Optical Character Recognition-theory and practice. In P. R. Krishnaiah and L. N. Kanal, editors, *Handbook of Statistics*, volume 2, pages 621-649. North Holland Publishing Company, 1982.

[Par88] Roger C. Parker. *Type Best Remembered.* URW Verlag, Chapell Hill NC, 1988.

[Pav86] T. Pavlidis. A vectorization and feature extractor for document recognition. *Computer Vision, Graphics, and Image Processing*, 35:111-127, 1986.

[Pav93] T. Pavlidis. Recognition of printed text under realistic conditions. *Pattern Recognition Letters*, 14(14):317-326, 4 1993.

[Qui89] V. Quint. Systems for the manipulation of structured documents. In J. Andre, R. Furuta, and V. Quint, editors. *Structured Documents*, pages 39-74. Cambridge University Press, 1989.

[Ram89] S. R. Ramesh. A generalized character recognition algorithm: A graphical approach. *PR*, 22(4):347-350, 1989.

[RKN93] S. V. Rice, J. Kanai, and T. A. Nartker. An evaluation of OCR accuracy. Technical report, ISRI, University of Nevada Las Vegas, 1993.

[RKN94] S. V. Rice, J. Kanai, and T. A. Nartker. The third annual test of OCR accuracy. Technical report, ISRI, University of Nevada Las Vegas, 1994.

[Rub88] R. Rubinstein. *Digital Typography: An Introduction to type and composition for computer system design*. Addison-Wesley, 1988.

[Sen94] R. Sennhauser. Improving the recognition accuracy of text recognition systems using typographical constraints. In *RIDT'94: Third International Conference on Raster Imaging and Digital Typography*, pages 273-282, Darmstadt, Germany, 4 1994.

[SLG+92] S. Srihari, S. Lam, V. Govindaraju, R. Srihari, and J. Hull. Document understanding: Research directions. Technical report, CEDAR, State University of New York at Buffalo, 1992.

[TA90] Tsujimoto and Asada. Understanding multi-articled documents. In *Proceedings of the International Conference on Pattern Recognition*, pages 551-556, 1990.

[TCB94] K. Taghva, A. Condit, and J. Borsack. Autotag: A tool for creating structure document collections from printed materials. Technical report, ISRI, University of Nevada Las Vegas, 12 1994.

[TCB95] K. Taghva, A. Condit, and J. Borsack. Autotag: The MANICURE document processing system. Technical report, ISRI, University of Nevada Las Vegas, 3 1995.

[TIAY90] H. Takahashi, N. Itoh, T. Amano, and A. Yamashita. A spelling correction method and its application to an OCR system. PR, 23(3/4):363-377, 1990.

[TSYC91] Y. Y. Tang, C. Y. Suen, C. D. Yan, and M. Cheriet. Document analysis and understanding: A brief survey. In *Proceedings of the International Conference on Document Analysis and Recognition,* pages 17-31, 1991.

[Way93] P. Wayner. Optimal character recognition. *Byte*, pages 203-210, 12 1993.

[WCW82] K. Y. Wong, R. G. Casey, and F. M. Wahl. Document analysis system. *IBM Journal Res. Develop.*, 26(6):647-656, 1982.

[WS88] L. Wilcox and A. Spitz. Automatic recognition and representation of documents. In J. C. van Vliet, editor, *Document Manipulation and Typography*, pages 47-57. Cambridge University Press, 1988.

[ZAI92] A. Zramdini, A. Azokly, and R. Ingold. Importance de l'identification de la fonte dans la reconnaissance structurelle de documents. *BIGRE 80: Actes de CNED'92: Traitement de l'Ecriture et des Documents*, pages 233-241, 1992.

[ZI92] A. Zramdini and R. Ingold. Girafocr ou la reconnaissance de characteres dans oscar-ii. Technical report, IIUF, University of Fribourg, Switzerland, 1993.

[ZI94] A. Zramdini and R. Ingold. Optical font recognition from projection profiles. In *RIDT'94: Third International Conference on Raster Imaging and Digital Typography*, pages 249-260, Darmstadt, Germany, 4 1994.

[ZI95] A. Zramdini and R. Ingold. ApOFIS: an A priori Optical Font Identification System. In *ICIAP'95: 8th International Conference on Image Analysis and Processing*, San Remo, Italy, 9 1995.

[Zra95] A. Zramdini. *Study of Optical Font Recognition Based on Global Typographical Features*. PhD thesis, University of Fribourg, Switzerland, 1995.

APPENDIX A

TYPEFACE PRODUCTION AND STATISTICS

In this appendix, font formats and standards are summarized and some statistics, founded by P. Karow for some measurements computed from a typeface base, are discussed. These measurements allow the differentiation between typefaces from the typeface production point of view.

A.1 Digital typeface production

The present section addresses the issue of typeface production and rendering and discusses a few typeface formats which are commonly used in the industry.

A.1.1 Digital type rendering

Electronic publishing employs three kinds of actors: text editors and formatters, output devices (printer, screen, etc.) and font sets. It also requires three types of people: the author, the reader and the character designer [And93]. Output devices utilize the type designer's drawings to render the text. Through the past century, rendering procedures have evolved from the steel movable character in 1435 with Gutenberg to the digital one nowadays.

A.1.1.1 A brief history

The production process of conventional steel characters requires many artists and artisans: the type designer draws characters on paper. The punch cutter or typeface

implementer cuts steel punches in a shape articulated by the type designer. These punches are utilized to impress a negative image of the character typically in brass matrices. The founder pours molten metal into these matrices to comprise type. At last, the movable type is assembled into lines, inked and then pressed with paper.

The practice of keyboards by the 'Lynotype' and 'Monotype' completely changed the typographer's working mode. These machines, controlled by a keyboard, generated line-blocks, where characters of each line were molded together.

The first fast phototypesetters emerged in the 1950's. They carried characters on negative masters through which photographic paper was rendered one character at a time. The photographic paper was then reproduced using a photosensitive plate.

The practice of computers has made a revolution in the typesetting world, where computers are exploited to edit the text and to pass it on to the phototypesetters or printers. With the first generation, texts were loaded on the phototypesetters using punched cards. Nowadays, computers and phototypesetters communicate through networks.

A comprehensive presentation of printing history is provided in [And93], beginning with the Gutenberg invention of the mobile steer character and ending with the different aspects of digital type production.

A.1.1.2 Printing model

Roughly all printing engines utilize the following font processing algorithm [And93]:

- Character selection: each character description is chosen by its name and code;
- Rendering: an outline and filling algorithm is called that decides which pixels of a bitmap have to be blackened;
- Caching: the final character bitmap is saved into cache memory before being copied upon request into the page image;
- Spacing: the starting position of the next character is updated using character metrics such as width, kerning and side bearings. Nowadays, with raster printers and phototypesetters, these values are sparsed into the glyph definitions. They need to be made explicit in metric files because DeskTop Publishing programs require exact knowledge of metrics in order to carry out justification and hyphenation tasks. For e.g., each PostScript font is supplied with an AFM (Adobe Font Metrics) file, which consists of a set of information related either to an entire font, or to each character individually. A detailed discussion on AFM files organization is given in [And93].

A.1.1.3 Printing techniques

Plates are made from films by raising, lowering or chemically treating the print area to distinguish it from the non-printing area. There are four major printing techniques [Col91]:

1. In 'letterpress' technique, the raised surface of the area to be printed is inked, and the image is shifted from the printing plate onto the paper by pressure. The key drawback of this method (oldest technique) is the length of time it takes to craft the photoengraved plates.
2. The 'gravure' technique also employs photoengraved plates, but the image is etched into the plate rather than raised out of it. The plate is inked and wiped clean so that ink stays behind only in the etched area.
3. In 'offset lithography', metal plates are prepared photographically by a simpler process than the previous two methods. It is currently the most frequently used process being comparatively cheaper.
4. With 'silkscreen' techniques, printing plates are less costly than litho, but each print is more expensive, hence this process is most appropriate for small runs. Silk-screening offers very even solids, making it fit for jobs such as posters and printing onto fabrics like t-shirts.

A.1.2 Digital type production

Constructing a database of typefaces, independent of machine formats, engages the creation of typeface data for a common database first, which can be utilized later for conversion into machine specific formats.

A.1.2.1 Production path

Figure A.1 illustrates the course of a typeface design: (1) the designer draft, (2) outline construction using computers, (3) specific machine format conversion, (4) rendering. The reader captures the font characters via display or print. Typefaces can be fed into computers in a variety of ways [Kar94b, AH92, And93]:

- Hand digitizing: font designers frequently do their drawings by hand and then digitize the outlines manually point by point by means of a suitable software such as 'Ikarus';
- Scanning: the font drawings are scanned to get a bitmap, which can then be processed by automatic outlining software such as 'Linus' or 'Typo';
- Interactive design: the designer uses an interactive outline editor (e.g. Fontstudio, Type) to draw characters;
- Programming: character drawings are directly programmed by means of specific languages, such as PostScript or METAFONT, defined by D. Knuth [Knu93].

The digitized characters formed by designers must be shaped in order to guarantee optimal performance on a range of display and printing devices. The shaping process may imply to ensure that all stems have the same width and that serifs do not differ from one character to another. In order to make sure that optimal character rendition occurs at both low and medium resolution, hints must be added to the character outline descriptions [AH92].

A.1.2.2 Font representation

Digital characters of a typeface may be stored, symbolized and reproduced inside computers by numerous methods. These methods vary in their economy, efficiency and typographical utility:

1. 'Bitmap' representation, where each character is depicted by a grid with on and off squares (bits), as illustrated in Figure A.2(a). This form of depiction has the benefit that it is instantaneously practical for producing an image of bits used by the printer or screen. However, it needs a substantial amount of computer memory since predefined samples of the typeface characters in each size and style have to be stored.

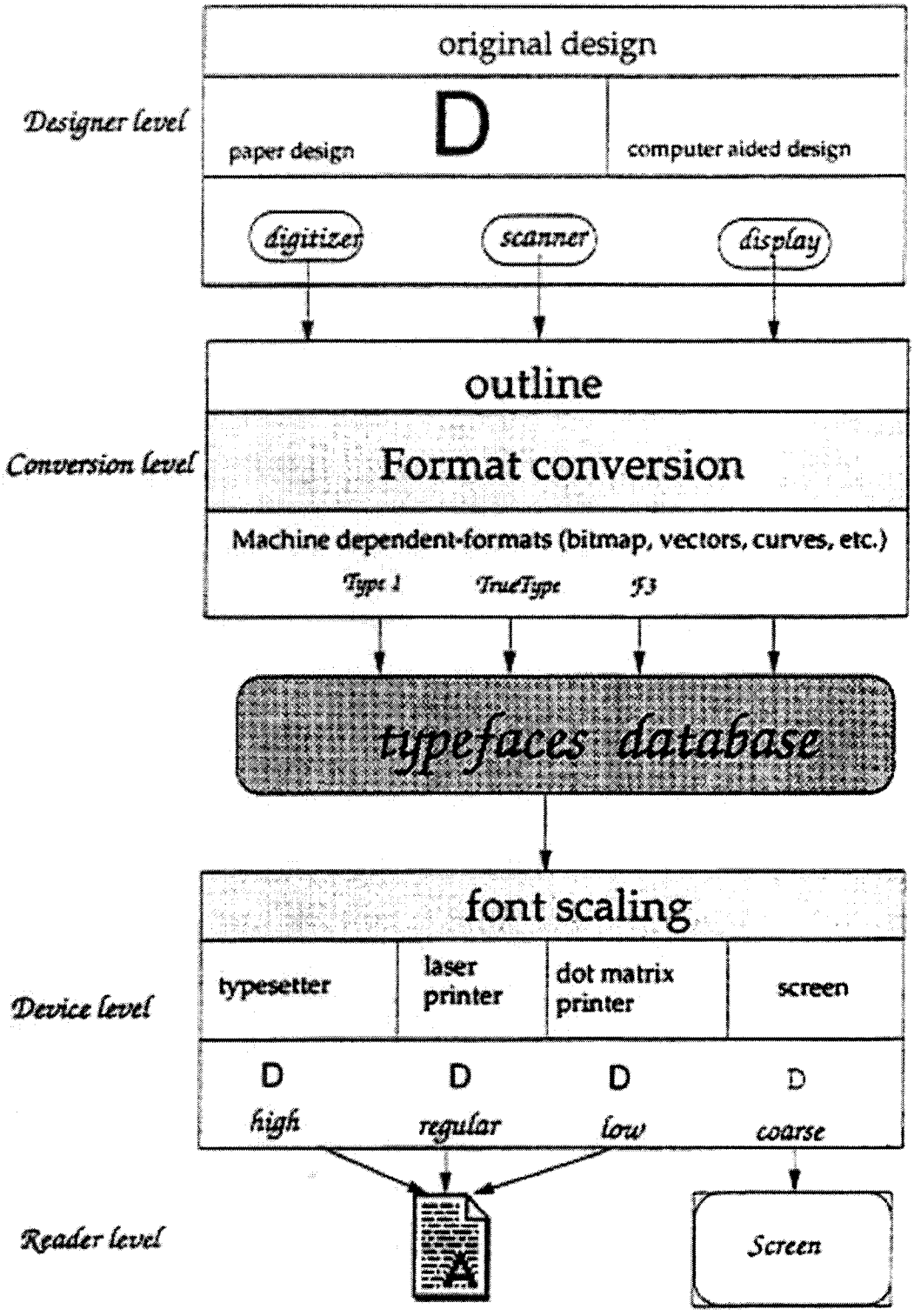


Figure A.1. The path of a typeface from designer to reader (from [Kar94b]).

The adjustment to a non-available size is done by an extrapolation of existing sizes, which leads usually to a stylistic deformation of the characters. In order to save memory, other encodings are also available such as 'run lengths'.

This representation remains especially for display on screens. Additionally, a number of printing systems still employ bitmap formats such as 'HP Laser Jet' printers [Col91];

2. 'Contour' representation, where each character is depicted by a point series representing its contours (see Figure A.2(b)). Since only points on the character boundaries are stored, fewer data is needed to encode the glyphs precisely. The outline curves may be characterized in several ways such as 'vectors', 'Bezier functions' or 'quadratic splines'.

   In order to print characters in this representation, an adaptation into a discrete grid format is needed. The alteration of the contour to the grid is still not a solved problem [Her93a, And93]. The outline description is usually accompanied by a set of 'hints', allowing software:

   - To solve the issue of the contour alteration to the grid;
   - To produce, without any deformation, characters of different sizes and styles.

3. 'Algorithmic' representation, which may be parametric, allows a variety of designs to be produced merely by parameter changing. METAFONT is a renowned language permitting the description of characters as programs [Knu93].

Figure A.2. Characters described (a) by their bitmaps and (b) by their contours.

The description and utilization of these techniques followed a technological evolution [And93]:

- Characters are depicted and used directly as bitmaps;
- Characters are expressed by their contours;
- A better alteration of contours to the grid, respecting typographical properties of individual characters, is established;
- A reflection on typographical properties between characters (optical adjustment, kerning, etc.) is done;
- Gray level characters are devised and utilized to render texts [Her93b].

A.1.3 Standard formats

Nowadays, computers store fonts as outlines and furnish them with hints for sharp rendering. This method utilizes the same format to produce text on display screens, laser printers or typesetting machines. 'Hints' play a vital role in the quality of the text rendered on low resolution devices, especially for small size texts.

Many formats describing characters by their contours are accessible in today's market:

1. 'TrueType' format, supported by 'Apple' for their 'Macintosh systems' [Com90], is presented as a programming language based on stack manipulations. It does not hold rules for the alteration to the grid, but possess some tools allowing to define rules and to apply them on the contour. In this type of format, the character contour is deformed in order to be tailored to the discrete points grid.

   The curves of an outline are represented by a series of parabola sections, defined by on-curve and off-curve control points. Besides character data, the 'TrueType' format consists of keyboard layout information, spacing, naming, font statistics and other typographical data.

2. 'Adobe Type1' format implemented by the 'PostScript' language, pursues the approach contrary to the TrueType's one. It deforms the discrete points grid to adjust it to the character contour. The grid lines and columns are spread vertically and horizontally in an optimal way to achieve an exact conversion of characters.

   In this format, character outlines are expressed by straights, Bezier curves and other special operators.

3. 'F3 Language', from SUN Microsystems, is a common programming language employed especially to describe geometric forms (characters) and their conversion to ensure optimum bitmap-format output at different resolutions.

Contours of F3 type characters are expressed as closed curves using straights, Bezier curves or conics.

4. 'Intellifont System' from Agfa Compugraphics is a native format of the 'Hewlet Packard Laserjet III' series. It is a rasterizer software producing character bitmaps or outlines in different resolutions and sizes using fonts in the FAIS format (Font Access and Interchange Standard).

In reality, each system comes with its particular font format, which leads to several, slightly dissimilar, versions of the original typeface design.

A.2 Typeface statistics

Peter Karow [Kar93] did a statistical study of about 1795 out of 3000 hand-digitized typefaces stored in the 'Ikarus' format [Kar94b]. The statistics were computed from measurements corresponding to some font metrics. From this set, 1049 fonts were seriffed and 485 sans-serif. The key measurements were:

1. Guidelines: four guidelines were described as illustrated in Figure A.3(a): X-height (Xh), x-height (xh), ascender (ah) and descender (dh), computed from special characters (I, h, p, e, c, o).

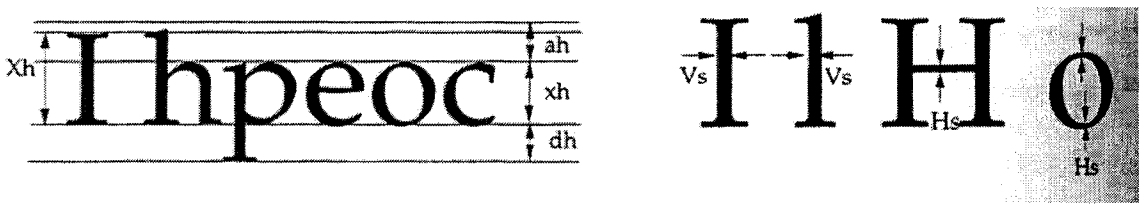


Figure A.3. (a) Guidelines definition. (b) Vertical and horizontal stroke definition.

The measurements proved that all fonts have roughly the same 'Xh', but different 'xh', 'ah' and 'dh' proportions. Table A.1 demonstrates the average proportion of the 'xh', 'ah' and 'dh' measurements in the 'Xh' values. It can be noticed that:

- Sans-serif fonts have bigger 'xh' than seriffed ones but smaller 'ah' and 'dh';
- Italic characters have slightly bigger 'ah' and 'dh' than upright ones.

	x-height / Xh		ascender / Xh		descender / Xh		contrast	
	Upright	Italic	Upright	Italic	Upright	Italic	Upright	Italic
seriffed	69.0%	69.0%	35.7%	36.0%	31.4%	32.3%	0.50%	0.47%
sans-serif	72.0%	71.5%	29.6%	30.9%	27.2%	28.1%	0.82%	0.82%

Table A.1. Proportion of the 'xh', 'ah' and 'dh' in the 'Xh' value, and contrast values.

2. Stroke widths: characters are largely composed of vertical and horizontal strokes (Vs and Hs) (see Figure A.3(b)). The statistics measured the width (thickness) of these strokes and computed them from special characters, i.e. I, l and H. The 'contrast' measures the ratio of horizontal to vertical stroke widths.

Table A.1 shows an obvious distinction between seriffed and sans-serif fonts. Sans-serif typefaces have a very low contrast (82%), while seriffed ones have high contrast with a ratio value of 50%. This measurement authenticates the mono-line feature of sans-serif fonts versus the stressed feature of seriffed fonts.

3. Character width and side bearings: these were measured on characters H, O, V as well as on n, o, and v. As shown in Table A.2, the left and right side bearings amount to 10% of the total width for sans-serif typefaces against 5% for the seriffed ones.

   In reality, seriffed glyphs are relatively wider than sans-serif ones: 75% vs. 63% of the body size. This is largely due to serifs which take up an important place in the character width.

	character width		left side bearing		right side bearing	
	Upright	Italic	Upright	Italic	Upright	Italic
seriffed	75.0%	75.0%	3.0%	0.0%	3.0%	-5.0%
sans-serif	63.0%	63.0%	6.0%	3.0%	6.0%	-2.0%

Table A.2. Proportion of the total width, left and right side bearings in the body size (character H).

4. Complexity: this represents the average number of contour elements of a character set. Corner points, straight lines, inflection points and curves, extracted from the character outline, are assumed as contour elements.

   The measurements separated sans-serif typefaces (simple typefaces), with an average of 20 contour elements, from the seriffed ones (more complex) with 30

elements. Still, a sans-serif 'm' is more complex than a seriffed 'l'. Therefore, this measurement is applicable only if computed from a large number of characters.

5. Serifs: measurements were taken for the serif length, foot and leg heights, as illustrated in Figure A.4. They were logically computed from seriffed typefaces and mostly extracted from characters I, i and 1. Table A.3 traces the intervals representing the measurement variations.

Measurements, especially serifs length, have revealed a wide range of values: from 1% to 24% of the typeface body, but they have also proved that serif lengths of main typefaces are distributed around the mean value (11%).

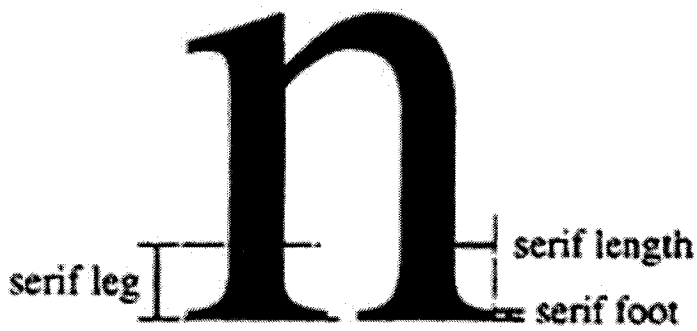


Figure A.4. Serif components.

	capitals (I)			lowercase (i)			numerals (1)		
	min	max	means	min	max	means	min	max	means
length	1%	24%	11%	0.5%	10%	6%	0.5%	20%	9%
foot height	0.1%	14%	4%	0.1%	14%	4%	0.1%	14%	4%
leg height	0.5%	25%	10%	0.2%	20%	8%	0.2%	30%	10%

Table A.3. Serif measurements values expressed as percentage of the body size.

In conclusion, guidelines, contrast, character width and complexity measurements have principally differentiated seriffed from sans-serif fonts, while serif measurements have revealed their wide shapes and hence do not allow distinguishing seriffed typefaces.

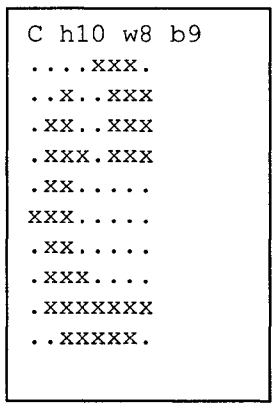
APPENDIX B

FONT SAMPLES

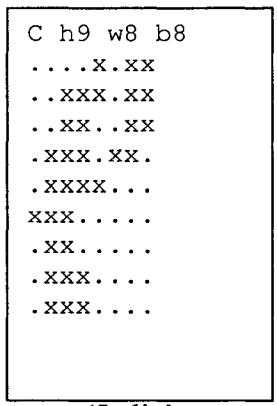
The classifier system represents 10 fonts corresponding to 4 typefaces with 3 font per typeface except the Avante-Garde, which only has a single font.

- 1 seriffed Postscript typeface: Times;
- 2 sans-serif Postscript typefaces: Helvetica and Avant-Garde;
- 1 typewriter Postscript typeface: Courier

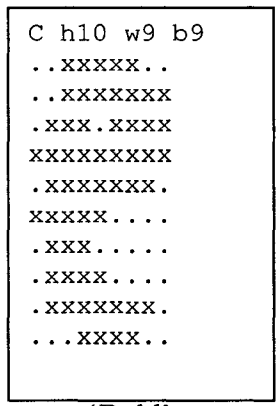
For each typeface 2 weights (regular, bold), 1 slope (italic) and 1 size (12pt) were considered. Following are samples of the considered typefaces of the character 'e'.



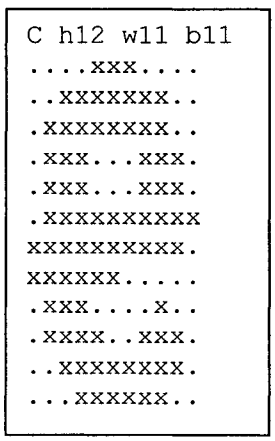
'Regular'
(Times New Roman)



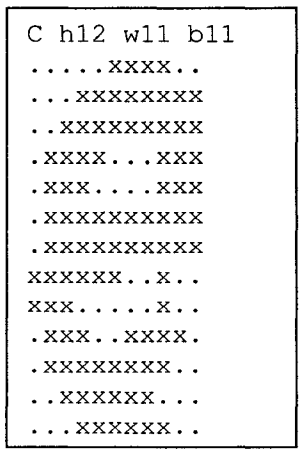
'Italic'
(Times New Roman)



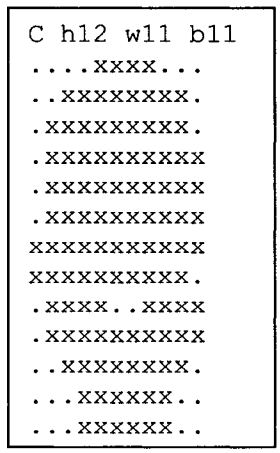
'Bold'
(Times New Roman)



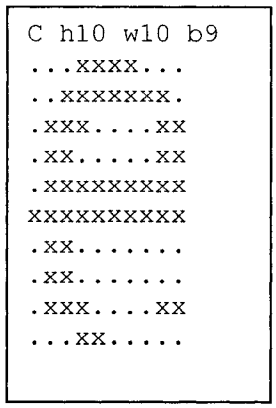
'Regular'
(Helvetica)



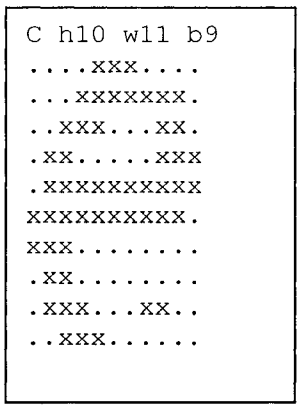
'Italic'
(Helvetica)



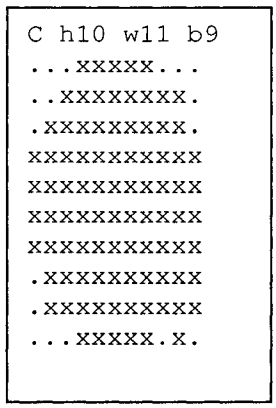
'Bold'
(Helvetica)



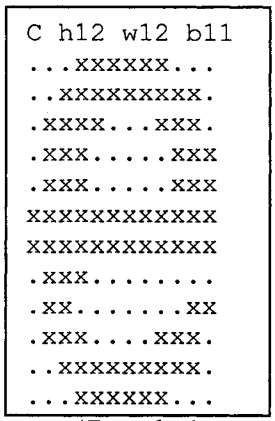
'Regular'
(Courier)



'Italic'
(Courier)



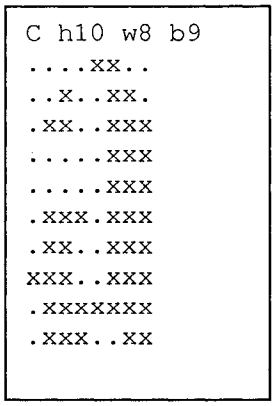
'Bold'
(Courier)



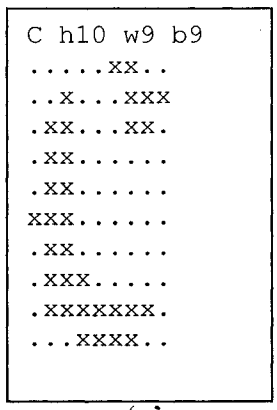
'Regular'
(Avante-Garde)

The classifier system was also tested on five other databases, namely Lowercase, Uppercase, Digit, Punctuation and Image Quality.

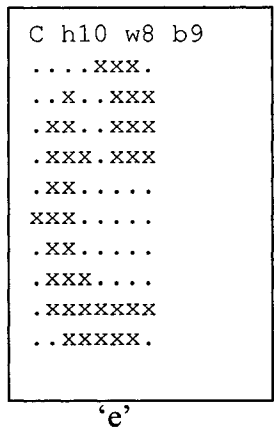
Following are samples of the considered lowercase characters.



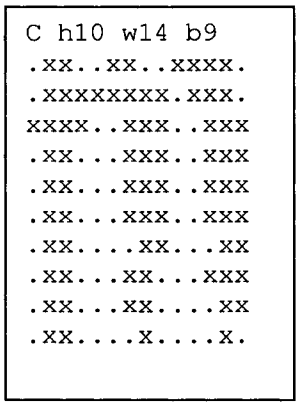
'a'
(Times New Roman)



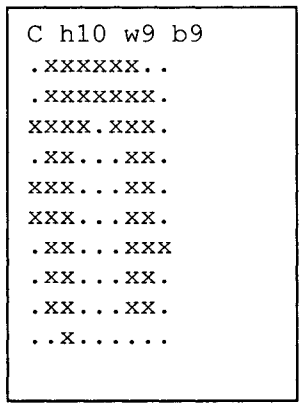
'c'
(Times New Roman)



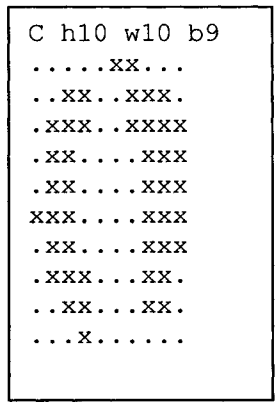
'e'
(Times New Roman)



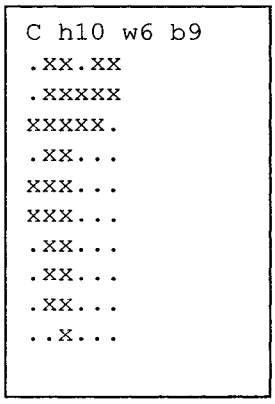
'm'
(Times New Roman)



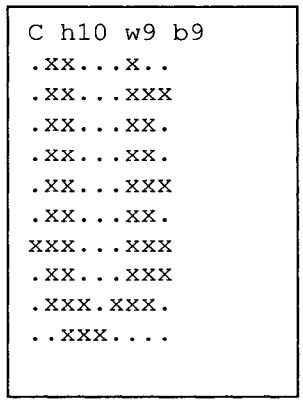
'n'
(Times New Roman)



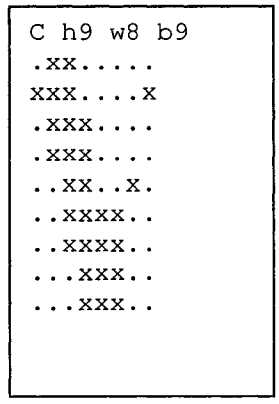
'o'
(Times New Roman)



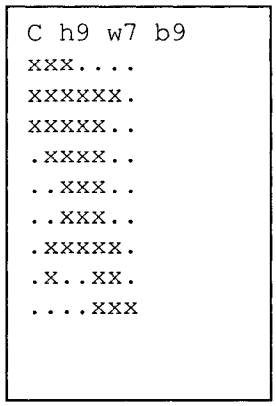
'r'
(Times New Roman)



'u'
(Times New Roman)

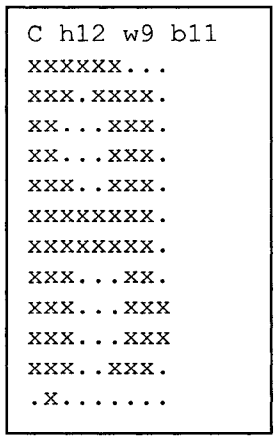


'v'
(Times New Roman)

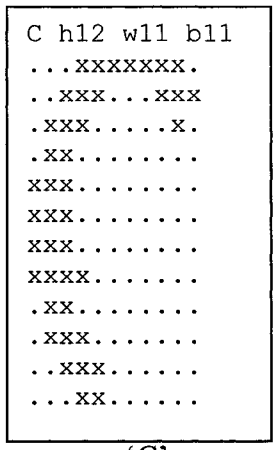


'x'
(Times New Roman)

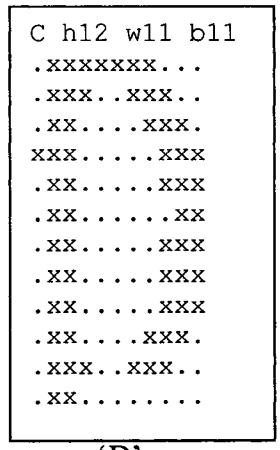
Following are samples of the considered uppercase characters.



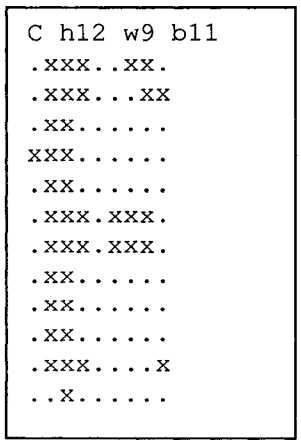
'B'
(Times New Roman)



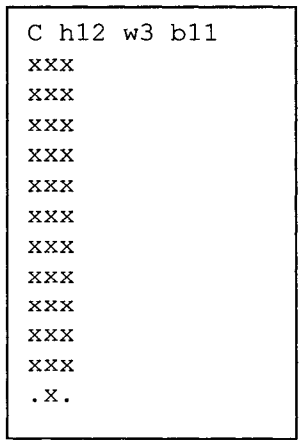
'C'
(Times New Roman)



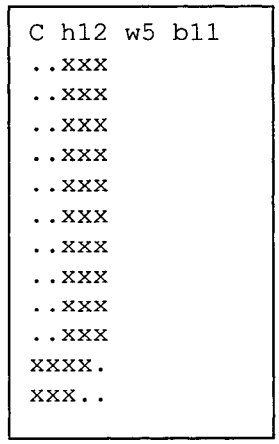
'D'
(Times New Roman)



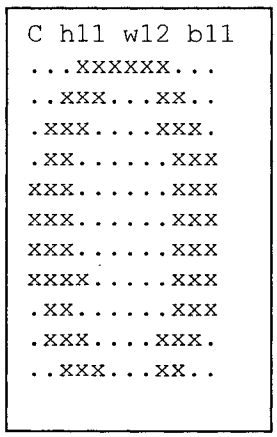
'E'
(Times New Roman)



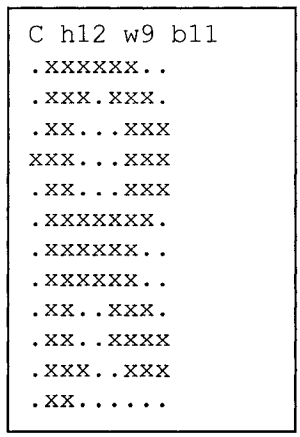
'I'
(Times New Roman)



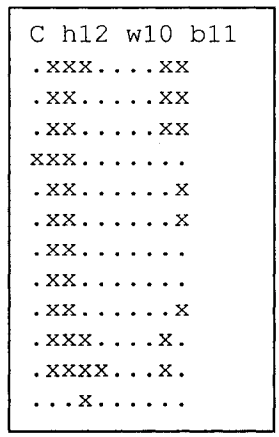
'J'
(Times New Roman)



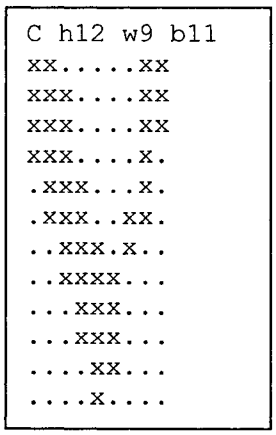
'O'
(Times New Roman)



'R'
(Times New Roman)

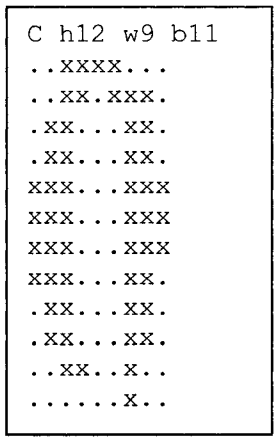


'U'
(Times New Roman)

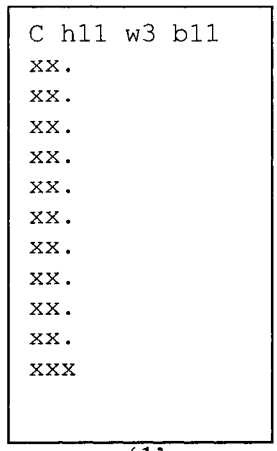


'V'
(Times New Roman)

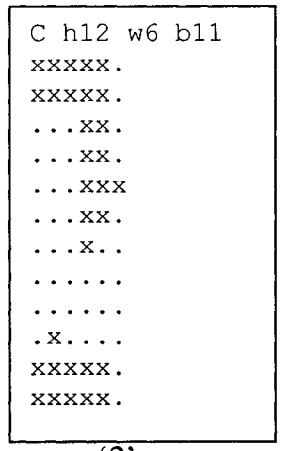
Following are samples of the considered digits.



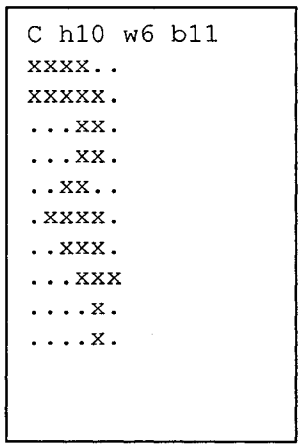
'0'
(Times New Roman)



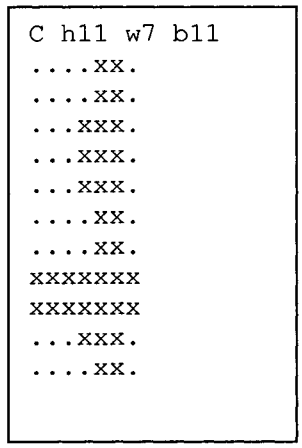
'1'
(Times New Roman)



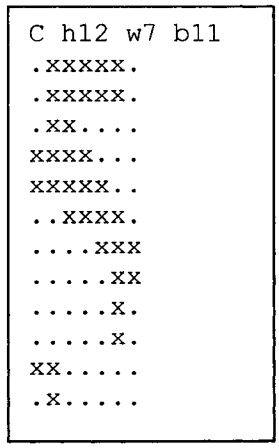
'2'
(Times New Roman)



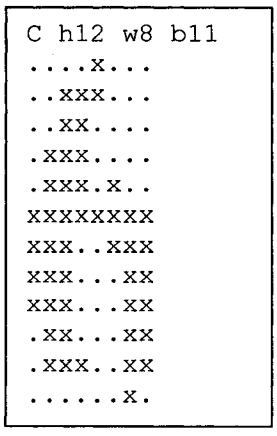
'3'
(Times New Roman)



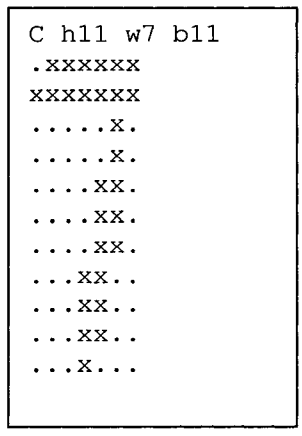
'4'
(Times New Roman)



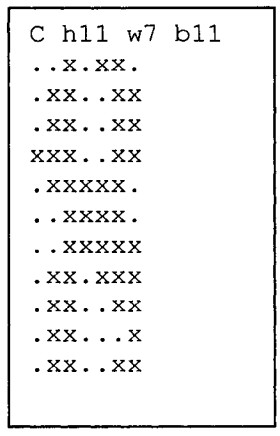
'5'
(Times New Roman)



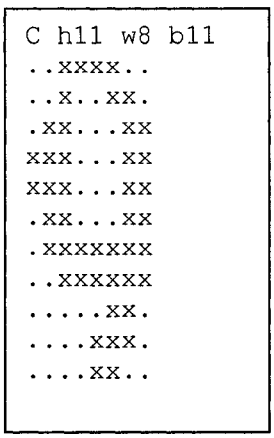
'6'
(Times New Roman)



'7'
(Times New Roman)

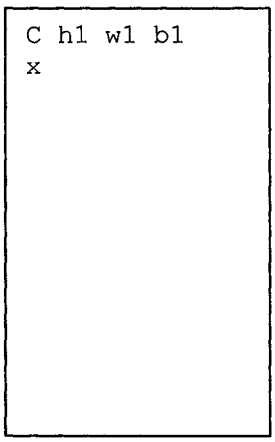


'8'
(Times New Roman)

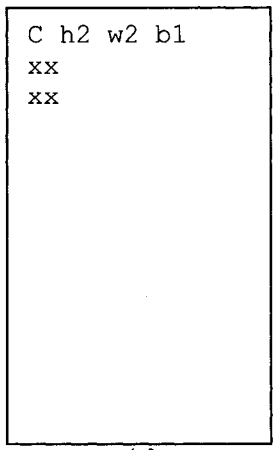


'9'
(Times New Roman)

Following are samples of the considered punctuation.

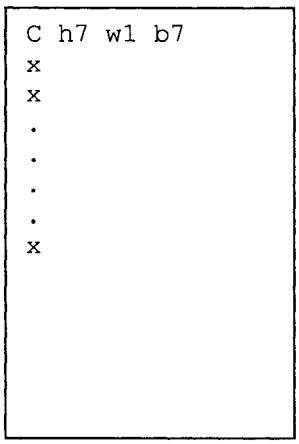


'.'
(Times New Roman)

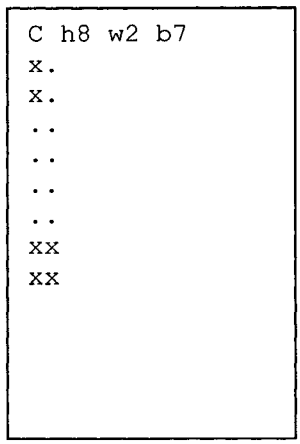


','
(Times New Roman)

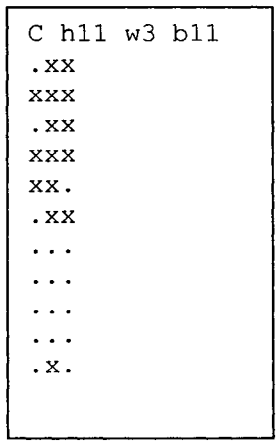
'-'
(Times New Roman)



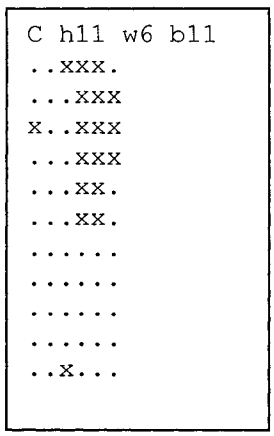
':'
(Times New Roman)



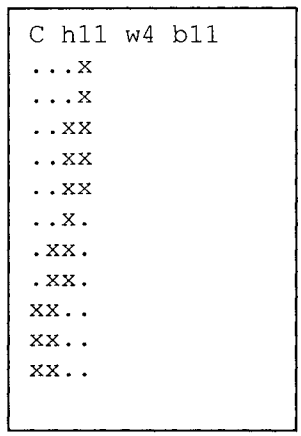
';'
(Times New Roman)



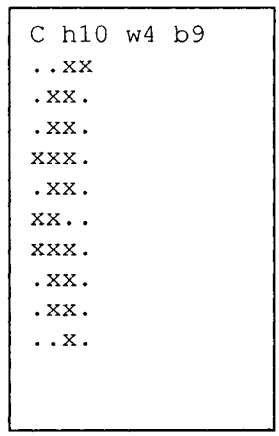
'!'
(Times New Roman)



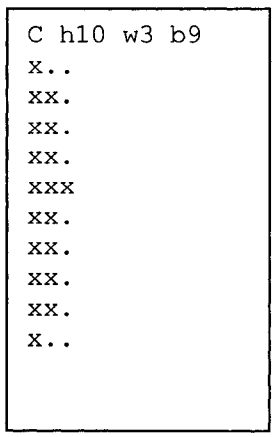
'?'
(Times New Roman)



'/'
(Times New Roman)

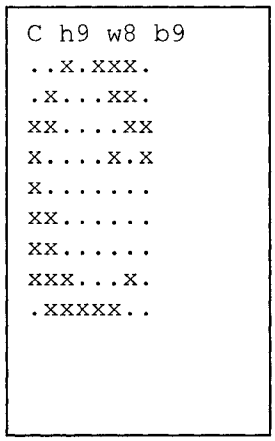


'('
(Times New Roman)

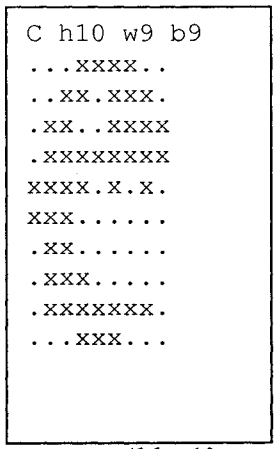


')'
(Times New Roman)

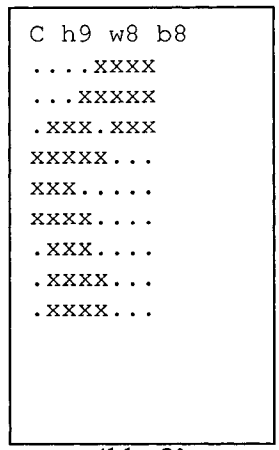
Following are samples of the considered image qualities of the character 'e'.



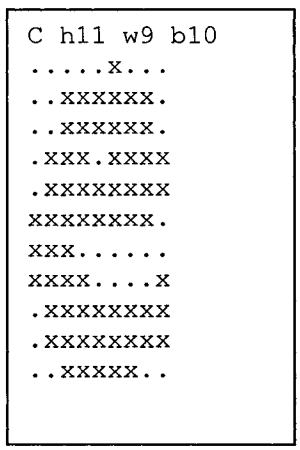
'ideal'
(Times New Roman)



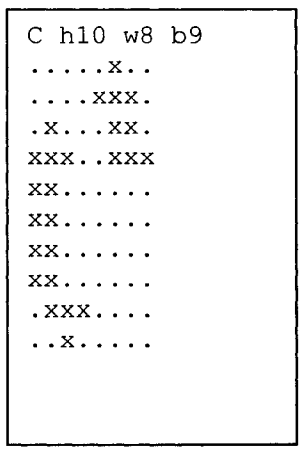
'blur1'
(Times New Roman)



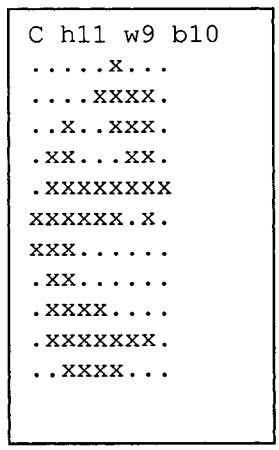
'blur2'
(Times New Roman)



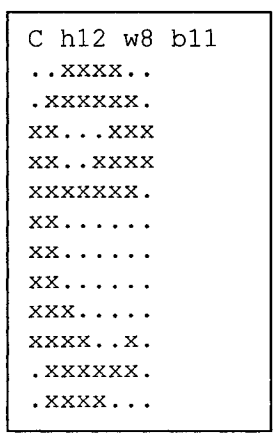
'fat'
(Times New Roman)



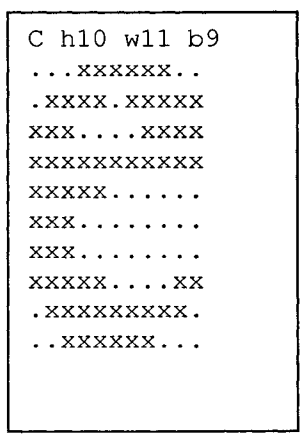
'thin'
(Times New Roman)



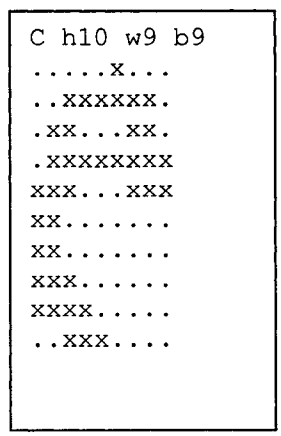
'noisy'
(Times New Roman)



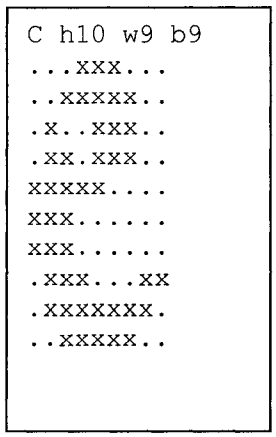
'tall'
(Times New Roman)



'wide'
(Times New Roman)



'rot1'
(Times New Roman)



'rot2'
(Times New Roman)

VITA

Osama Ahmed Khan

1609 W McIntyre St, Apt 6, Edinburg, TX, 956-380-6961, osamaahmedkhan@ieee.org

OBJECTIVE:

- Designing and implementing Bioengineering algorithms

EDUCATION:

- The University Of Texas-Pan American, Edinburg, TX *Dec 2004*

  Master of Science in Computer Science, Cumulative GPA: 3.91

  Thesis: Parametric Classification In Domains Of Characters, Numerals, Punctuation, Typefaces And Image Qualities

- Ned University Of Engg. & Tech., Karachi, Pakistan *Mar 2002*

  Bachelor of Engineering in Computer Engineering, Cumulative GPA: 3.78

INDUSTRIAL EXPERIENCES:

- The University Of Texas-Pan American, Edinburg, TX, TchAssist Aug 2003-May 2004
- Softech Microsystems, Karachi, Pakistan, Systems Engineer Mar 2002 - Nov 2002
- Mobilink, Karachi, Pakistan, Senior Year Project Jan 2001- Dec 2001
- Cressoft Inc., Karachi, Pakistan, Intern Winter, 2000
- Goldsoft, Karachi, Pakistan, Intern Summer, 2000
- EOBI, Karachi, Pakistan, Intern Summer, 2000