CHAPTER

Introduction 1

1.1 IS PATTERN RECOGNITION IMPORTANT?

Pattern recognition is the scientific discipline whose goal is the classification of *objects* into a number of categories or *classes*. Depending on the application, these objects can be images or signal waveforms or any type of measurements that need to be classified. We will refer to these objects using the generic term *patterns*. Pattern recognition has a long history, but before the 1960s it was mostly the output of theoretical research in the area of statistics. As with everything else, the advent of computers increased the demand for practical applications of pattern recognition, which in turn set new demands for further theoretical developments. As our society evolves from the industrial to its postindustrial phase, automation in industrial production and the need for information handling and retrieval are becoming increasingly important. This trend has pushed pattern recognition to the high edge of today's engineering applications and research. Pattern recognition is an integral part of most *machine intelligence* systems built for decision making.

Machine vision is an area in which pattern recognition is of importance. A machine vision system captures images via a camera and analyzes them to produce descriptions of what is imaged. A typical application of a machine vision system is in the manufacturing industry, either for automated visual inspection or for automation in the assembly line. For example, in inspection, manufactured objects on a moving conveyor may pass the inspection station, where the camera stands, and it has to be ascertained whether there is a defect. Thus, images have to be analyzed online, and a pattern recognition system has to classify the objects into the "defect" or "nondefect" class. After that, an action has to be taken, such as to reject the offending parts. In an assembly line, different objects must be located and "recognized," that is, classified in one of a number of classes known *a priori*. Examples are the "screwdriver class," the "German key class," and so forth in a tools' manufacturing unit. Then a robot arm can move the objects in the right place.

Character (letter or number) recognition is another important area of pattern recognition, with major implications in automation and information handling. Optical character recognition (OCR) systems are already commercially available and more or less familiar to all of us. An OCR system has a "front-end" device consisting of a *light source*, a *scan lens*, a *document transport*, and a *detector*. At the output of

the light-sensitive detector, light-intensity variation is translated into "numbers" and an image array is formed. In the sequel, a series of image processing techniques are applied leading to *line* and *character segmentation*. The pattern recognition software then takes over to recognize the characters—that is, to classify each character in the correct "letter, number, punctuation" class. Storing the recognized document has a twofold advantage over storing its scanned image. First, further electronic processing, if needed, is easy via a word processor, and second, it is much more efficient to store ASCII characters than a document image. Besides the printed character recognition systems, there is a great deal of interest invested in systems that recognize handwriting. A typical commercial application of such a system is in the machine reading of bank checks. The machine must be able to recognize the amounts in figures and digits and match them. Furthermore, it could check whether the payee corresponds to the account to be credited. Even if only half of the checks are manipulated correctly by such a machine, much labor can be saved from a tedious job. Another application is in automatic mail-sorting machines for postal code identification in post offices. Online handwriting recognition systems are another area of great commercial interest. Such systems will accompany *pen computers*, with which the entry of data will be done not via the keyboard but by writing. This complies with today's tendency to develop machines and computers with interfaces acquiring human-like skills.

Computer-aided diagnosis is another important application of pattern recognition, aiming at assisting doctors in making diagnostic decisions. The final diagnosis is, of course, made by the doctor. Computer-assisted diagnosis has been applied to and is of interest for a variety of medical data, such as X-rays, computed tomographic images, ultrasound images, electrocardiograms (ECGs), and electroencephalograms (EEGs). The need for a computer-aided diagnosis stems from the fact that medical data are often not easily interpretable, and the interpretation can depend very much on the skill of the doctor. Let us take for example *X-ray mammography* for the detection of breast cancer. Although mammography is currently the best method for detecting breast cancer, 10 to 30% of women who have the disease and undergo mammography have negative mammograms. In approximately two thirds of these cases with false results the radiologist failed to detect the cancer, which was evident retrospectively. This may be due to poor image quality, eye fatigue of the radiologist, or the subtle nature of the findings. The percentage of correct classifications improves at a second reading by another radiologist. Thus, one can aim to develop a pattern recognition system in order to assist radiologists with a "second" opinion. Increasing confidence in the diagnosis based on mammograms would, in turn, decrease the number of patients with suspected breast cancer who have to undergo surgical breast biopsy, with its associated complications.

Speech recognition is another area in which a great deal of research and development effort has been invested. Speech is the most natural means by which humans communicate and exchange information. Thus, the goal of building intelligent machines that recognize *spoken information* has been a long-standing one for scientists and engineers as well as science fiction writers. Potential applications of such machines are numerous. They can be used, for example, to improve efficiency

in a manufacturing environment, to control machines in hazardous environments remotely, and to help handicapped people to control machines by talking to them. A major effort, which has already had considerable success, is to enter data into a computer via a microphone. Software, built around a pattern (spoken sounds in this case) recognition system, recognizes the spoken text and translates it into ASCII characters, which are shown on the screen and can be stored in the memory. Entering information by "talking" to a computer is twice as fast as entry by a skilled typist. Furthermore, this can enhance our ability to communicate with deaf and dumb people.

Data mining and knowledge discovery in databases is another key application area of pattern recognition. Data mining is of intense interest in a wide range of applications such as medicine and biology, market and financial analysis, business management, science exploration, image and music retrieval. Its popularity stems from the fact that in the age of information and knowledge society there is an ever increasing demand for retrieving information and turning it into knowledge. Moreover, this information exists in huge amounts of data in various forms including, text, images, audio and video, stored in different places distributed all over the world. The traditional way of searching information in databases was the description-based model where object retrieval was based on keyword description and subsequent word matching. However, this type of searching presupposes that a manual annotation of the stored information has previously been performed by a human. This is a very time-consuming job and, although feasible when the size of the stored information is limited, it is not possible when the amount of the available information becomes large. Moreover, the task of manual annotation becomes problematic when the stored information is widely distributed and shared by a heterogeneous "mixture" of sites and users. Content-based retrieval systems are becoming more and more popular where information is sought based on "similarity" between an object, which is presented into the system, and objects stored in sites all over the world. In a content-based image retrieval CBIR (system) an image is presented to an input device (e.g., scanner). The system returns "similar" images based on a measured "signature," which can encode, for example, information related to color, texture and shape. In a music content-based retrieval system, an example (i.e., an extract from a music piece), is presented to a microphone input device and the system returns "similar" music pieces. In this case, similarity is based on certain (automatically) measured cues that characterize a music piece, such as the music meter, the music tempo, and the location of certain repeated patterns.

Mining for biomedical and DNA data analysis has enjoyed an explosive growth since the mid-1990s. All DNA sequences comprise four basic building elements; the nucleotides: adenine (A), cytosine (C), guanine (G) and thymine (T). Like the letters in our alphabets and the seven notes in music, these four nucleotides are combined to form long sequences in a twisted ladder form. Genes consist of, usually, hundreds of nucleotides arranged in a particular order. Specific gene-sequence patterns are related to particular diseases and play an important role in medicine. To this end, pattern recognition is a key area that offers a wealth of developed tools for similarity search and comparison between DNA sequences. Such comparisons

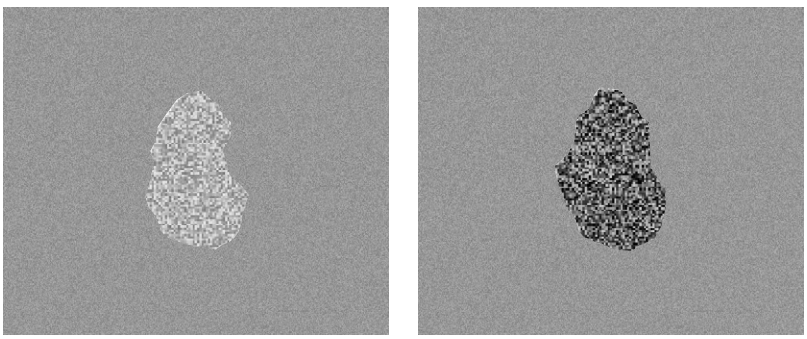
between healthy and diseased tissues are very important in medicine to identify critical differences between these two classes.

The foregoing are only five examples from a much larger number of possible applications. Typically, we refer to fingerprint identification, signature authentication, text retrieval, and face and gesture recognition. The last applications have recently attracted much research interest and investment in an attempt to facilitate human–machine interaction and further enhance the role of computers in office automation, automatic personalization of environments, and so forth. Just to provoke imagination, it is worth pointing out that the MPEG-7 standard includes a provision for content-based video information retrieval from digital libraries of the type: search and find all video scenes in a digital library showing person "X" laughing. Of course, to achieve the final goals in all of these applications, pattern recognition is closely linked with other scientific disciplines, such as linguistics, computer graphics, machine vision, and database design.

Having aroused the reader's curiosity about pattern recognition, we will next sketch the basic philosophy and methodological directions in which the various pattern recognition approaches have evolved and developed.

1.2 FEATURES, FEATURE VECTORS, AND CLASSIFIERS

Let us first simulate a simplified case "mimicking" a medical image classification task. Figure 1.1 shows two images, each having a distinct region inside it. The two regions are also themselves visually different. We could say that the region of Figure 1.1a results from a benign lesion, class A, and that of Figure 1.1b from a malignant one (cancer), class B. We will further assume that these are not the only patterns (images) that are available to us, but we have access to an image database



(a) (b)

FIGURE 1.1

Examples of image regions corresponding to (a) class A and (b) class B.

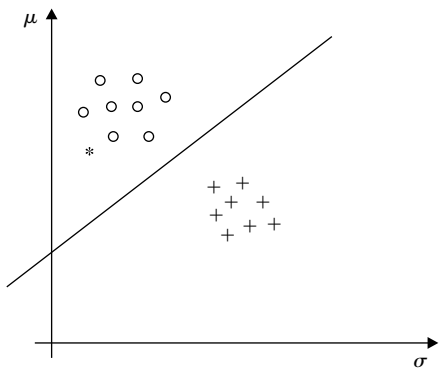


FIGURE 1.2

Plot of the mean value versus the standard deviation for a number of different images originating from class A (○) and class B (+). In this case, a straight line separates the two classes.

with a number of patterns, some of which are known to originate from class A and some from class B.

The first step is to identify the measurable quantities that make these two regions *distinct* from each other. Figure 1.2 shows a plot of the mean value of the intensity in each region of interest versus the corresponding standard deviation around this mean. Each point corresponds to a different image from the available database. It turns out that class A patterns tend to spread in a different area from class B patterns. The straight line seems to be a good candidate for separating the two classes. Let us now assume that we are given a new image with a region in it and that we do not know to which class it belongs. It is reasonable to say that we measure the mean intensity and standard deviation in the region of interest and we plot the corresponding point. This is shown by the asterisk (∗) in Figure 1.2. Then it is sensible to assume that the unknown pattern is *more likely* to belong to class A than class B.

The preceding artificial *classification* task has outlined the rationale behind a large class of pattern recognition problems. The measurements used for the classification, the mean value and the standard deviation in this case, are known as *features*. In the more general case l features x_i, $i = 1, 2, \ldots, l$, are used, and they form the *feature vector*

$$\boldsymbol{x} = [x_1, x_2, \ldots, x_l]^T$$

where T denotes transposition. Each of the feature vectors identifies *uniquely* a single pattern (object). Throughout this book features and feature vectors will be treated as *random variables* and *vectors*, respectively. This is natural, as the measurements resulting from different patterns exhibit a random variation. This is due partly to the measurement noise of the measuring devices and partly to

the distinct characteristics of each pattern. For example, in X-ray imaging large variations are expected because of the differences in physiology among individuals. This is the reason for the scattering of the points in each class shown in Figure 1.1.

The straight line in Figure 1.2 is known as the *decision* line, and it constitutes the *classifier* whose role is to divide the feature space into regions that correspond to either class A or class B. If a feature vector $\boldsymbol{x}$, corresponding to an unknown pattern, falls in the class A region, it is classified as class A, otherwise as class B. This does not necessarily mean that the decision is correct. If it is not correct, a *misclassification* has occurred. In order to draw the straight line in Figure 1.2 we exploited the fact that we knew the labels (class A or B) for each point of the figure. The patterns (feature vectors) whose true class is known and which are used for the design of the classifier are known as *training patterns (training feature vectors)*.

Having outlined the definitions and the rationale, let us point out the basic questions arising in a classification task.

- How are the features generated? In the preceding example, we used the mean and the standard deviation, because we knew how the images had been generated. In practice, this is far from obvious. It is problem dependent, and it concerns the *feature generation stage* of the design of a classification system that performs a given pattern recognition task.
- What is the best number l of features to use? This is also a very important task and it concerns the *feature selection stage* of the classification system. In practice, a larger than necessary number of feature candidates is generated, and then the "best" of them is adopted.
- Having adopted the appropriate, for the specific task, features, how does one design the classifier? In the preceding example the straight line was drawn empirically, just to please the eye. In practice, this cannot be the case, and the line should be drawn optimally, with respect to an *optimality criterion*. Furthermore, problems for which a linear classifier (straight line or hyperplane in the l-dimensional space) can result in acceptable performance are not the rule. In general, the surfaces dividing the space in the various class regions are nonlinear. What type of nonlinearity must one adopt, and what type of optimizing criterion must be used in order to locate a surface in the right place in the l-dimensional *feature space*? These questions concern the *classifier design stage*.
- Finally, once the classifier has been designed, how can one assess the performance of the designed classifier? That is, what is the *classification error rate*? This is the task of the *system evaluation stage*.

Figure 1.3 shows the various stages followed for the design of a classification system. As is apparent from the feedback arrows, these stages are not independent. On the contrary, they are interrelated and, depending on the results, one may go back

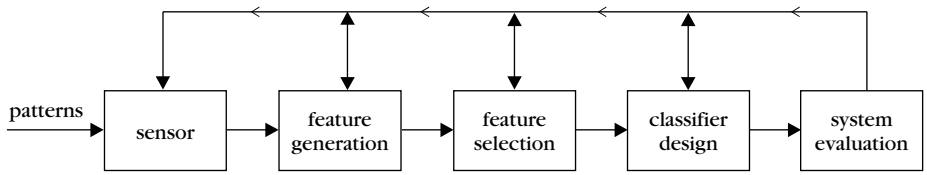


FIGURE 1.3
The basic stages involved in the design of a classification system.

to redesign earlier stages in order to improve the overall performance. Furthermore, there are some methods that combine stages, for example, the feature selection and the classifier design stage, in a common optimization task.

Although the reader has already been exposed to a number of basic problems at the heart of the design of a classification system, there are still a few things to be said.

1.3 SUPERVISED, UNSUPERVISED, AND SEMI-SUPERVISED LEARNING

In the example of Figure 1.1, we assumed that a set of training data were available, and the classifier was designed by exploiting this *a priori* known information. This is known as *supervised pattern recognition* or in the more general context of machine learning as *supervised learning*. However, this is not always the case, and there is another type of pattern recognition tasks for which training data, of known class labels, are not available. In this type of problem, we are given a set of feature vectors $\boldsymbol{x}$ and the goal is to unravel the underlying *similarities* and *cluster* (group) "similar" vectors together. This is known as *unsupervised pattern recognition* or *unsupervised learning* or *clustering*. Such tasks arise in many applications in social sciences and engineering, such as remote sensing, image segmentation, and image and speech coding. Let us pick two such problems.

In *multispectral remote sensing*, the electromagnetic energy emanating from the earth's surface is measured by sensitive scanners located aboard a satellite, an aircraft, or a space station. This energy may be reflected solar energy (passive) or the reflected part of the energy transmitted from the vehicle (active) in order to "interrogate" the earth's surface. The scanners are sensitive to a number of wavelength bands of the electromagnetic radiation. Different properties of the earth's surface contribute to the reflection of the energy in the different bands. For example, in the visible–infrared range properties such as the mineral and moisture contents of soils, the sedimentation of water, and the moisture content of vegetation are the main contributors to the reflected energy. In contrast, at the thermal end of the infrared, it is the thermal capacity and thermal properties of the surface and near subsurface that contribute to the reflection. Thus, each band measures different properties

of the same patch of the earth's surface. In this way, images of the earth's surface corresponding to the spatial distribution of the reflected energy in each band can be created. The task now is to exploit this information in order to identify the various ground cover types, that is, built-up land, agricultural land, forest, fire burn, water, and diseased crop. To this end, one feature vector $\boldsymbol{x}$ for each cell from the "sensed" earth's surface is formed. The elements $x_i, i = 1, 2, \ldots, l$, of the vector are the corresponding image pixel intensities in the various spectral bands. In practice, the number of spectral bands varies.

A *clustering* algorithm can be employed to reveal the groups in which feature vectors are clustered in the l-dimensional feature space. Points that correspond to the same ground cover type, such as water, are expected to cluster together and form groups. Once this is done, the analyst can identify the type of each cluster by associating a sample of points in each group with available reference ground data, that is, maps or visits. Figure 1.4 demonstrates the procedure.

Clustering is also widely used in the social sciences in order to study and correlate survey and statistical data and draw useful conclusions, *which will then lead to the right actions*. Let us again resort to a simplified example and assume that we are interested in studying whether there is any relation between a country's gross national product (GNP) and the level of people's illiteracy, on the one hand, and children's mortality rate on the other. In this case, each country is represented by a three-dimensional feature vector whose coordinates are indices measuring the quantities of interest. A clustering algorithm will then reveal a rather compact cluster corresponding to countries that exhibit low GNPs, high illiteracy levels, and high children's mortality expressed as a population percentage.

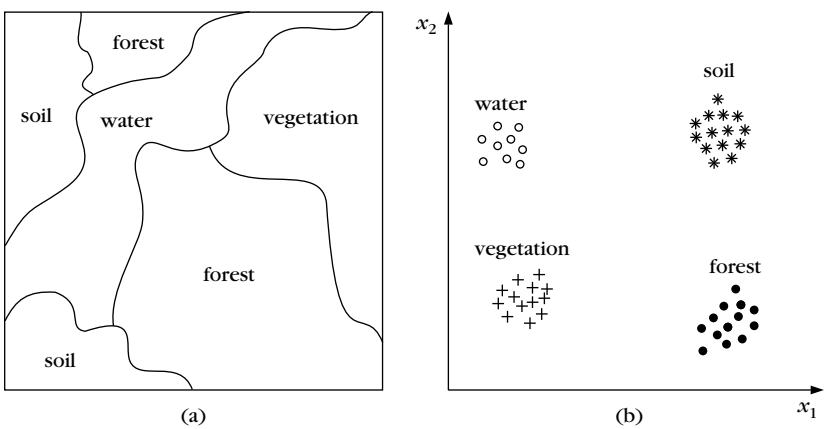


FIGURE 1.4

(a) An illustration of various types of ground cover and (b) clustering of the respective features for multispectral imaging using two bands.

A major issue in unsupervised pattern recognition is that of defining the "similarity" between two feature vectors and choosing an appropriate measure for it. Another issue of importance is choosing an algorithmic scheme that will cluster (group) the vectors on the basis of the adopted similarity measure. In general, different algorithmic schemes may lead to different results, which the expert has to interpret.

Semi-supervised learning/pattern recognition for designing a classification system shares the same goals as the supervised case, however now, the designer has at his or her disposal a set of patterns of unknown class origin, in addition to the training patterns, whose true class is known. We usually refer to the former ones as *unlabeled* and the latter as *labeled* data. Semi-supervised pattern recognition can be of importance when the system designer has access to a rather limited number of labeled data. In such cases, recovering additional information from the unlabeled samples, related to the general structure of the data at hand, can be useful in improving the system design. Semi-supervised learning finds its way also to clustering tasks. In this case, labeled data are used as constraints in the form of *must-links* and *cannot-links*. In other words, the clustering task is constrained to assign certain points in the same cluster or to exclude certain points of being assigned in the same cluster. From this perspective, semi-supervised learning provides an *a priori* knowledge that the clustering algorithm has to respect.

1.4 MATLAB PROGRAMS

At the end of most of the chapters there is a number of MATLAB programs and computer experiments. The MATLAB codes provided are not intended to form part of a software package, but they are to serve a purely pedagogical goal. Most of these codes are given to our students who are asked to play with and discover the "secrets" associated with the corresponding methods. This is also the reason that for most of the cases the data used are simulated data around the Gaussian distribution. They have been produced carefully in order to guide the students in understanding the basic concepts. This is also the reason that the provided codes correspond to those of the techniques and algorithms that, to our opinion, comprise the backbone of each chapter and the student has to understand in a first reading. Whenever the required MATLAB code was available (at the time this book was prepared) in a MATLAB toolbox, we chose to use the associated MATLAB function and explain how to use its arguments. No doubt, each instructor has his or her own preferences, experiences, and unique way of viewing teaching. The provided routines are written in a way that can run on other data sets as well. In a separate accompanying book we provide a more complete list of MATLAB codes embedded in a user-friendly Graphical User Interface (GUI) and also involving more realistic examples using real images and audio signals.

1.5 OUTLINE OF THE BOOK

Chapters 2–10 deal with supervised pattern recognition and Chapters 11–16 deal with the unsupervised case. Semi-supervised learning is introduced in Chapter 10. The goal of each chapter is to start with the basics, definitions, and approaches, and move progressively to more advanced issues and recent techniques. To what extent the various topics covered in the book will be presented in a first course on pattern recognition depends very much on the course's focus, on the students' background, and, of course, on the lecturer. In the following outline of the chapters, we give our view and the topics that we cover in a first course on pattern recognition. No doubt, other views do exist and may be better suited to different audiences. At the end of each chapter, a number of problems and computer exercises are provided.

Chapter 2 is focused on Bayesian classification and techniques for estimating unknown probability density functions. In a first course on pattern recognition, the sections related to Bayesian inference, the maximum entropy, and the expectation maximization (EM) algorithm are omitted. Special focus is put on the Bayesian classification, the minimum distance (Euclidean and Mahalanobis), the nearest neighbor classifiers, and the naive Bayes classifier. Bayesian networks are briefly introduced.

Chapter 3 deals with the design of linear classifiers. The sections dealing with the probability estimation property of the mean square solution as well as the bias variance dilemma are only briefly mentioned in our first course. The basic philosophy underlying the support vector machines can also be explained, although a deeper treatment requires mathematical tools (summarized in Appendix C) that most of the students are not familiar with during a first course class. On the contrary, emphasis is put on the linear separability issue, the perceptron algorithm, and the mean square and least squares solutions. After all, these topics have a much broader horizon and applicability. Support vector machines are briefly introduced. The geometric interpretation offers students a better understanding of the SVM theory.

Chapter 4 deals with the design of nonlinear classifiers. The section dealing with exact classification is bypassed in a first course. The proof of the backpropagation algorithm is usually very boring for most of the students and we bypass its details. A description of its rationale is given, and the students experiment with it using MATLAB. The issues related to cost functions are bypassed. Pruning is discussed with an emphasis on generalization issues. Emphasis is also given to Cover's theorem and radial basis function (RBF) networks. The nonlinear support vector machines, decision trees, and combining classifiers are only briefly touched via a discussion on the basic philosophy behind their rationale.

Chapter 5 deals with the feature selection stage, and we have made an effort to present most of the well-known techniques. In a first course we put emphasis on the t-test. This is because hypothesis testing also has a broad horizon, and at the same time it is easy for the students to apply it in computer exercises. Then, depending on time constraints, divergence, Bhattacharyya distance, and scattered matrices are presented and commented on, although their more detailed treatment

is for a more advanced course. Emphasis is given to Fisher's linear discriminant method (LDA) for the two-class case.

Chapter 6 deals with the feature generation stage using transformations. The Karhunen–Loève transform and the singular value decomposition are first introduced as dimensionality reduction techniques. Both methods are briefly covered in the second semester. In the sequel the independent component analysis (ICA), nonnegative matrix factorization and nonlinear dimensionality reduction techniques are presented. Then the discrete Fourier transform (DFT), discrete cosine transform (DCT), discrete sine transform (DST), Hadamard, and Haar transforms are defined. The rest of the chapter focuses on the discrete time wavelet transform. The incentive is to give all the necessary information so that a newcomer in the wavelet field can grasp the basics and be able to develop software, based on filter banks, in order to generate features. All these techniques are bypassed in a first course.

Chapter 7 deals with feature generation focused on image and audio classification. The sections concerning local linear transforms, moments, parametric models, and fractals are not covered in a first course. Emphasis is placed on first- and second-order statistics features as well as the run-length method. The chain code for shape description is also taught. Computer exercises are then offered to generate these features and use them for classification for some case studies. In a one-semester course there is no time to cover more topics.

Chapter 8 deals with template matching. Dynamic programming (DP) and the Viterbi algorithm are presented and then applied to speech recognition. In a two-semester course, emphasis is given to the DP and the Viterbi algorithm. The edit distance seems to be a good case for the students to grasp the basics. Correlation matching is taught and the basic philosophy behind deformable template matching can also be presented.

Chapter 9 deals with context-dependent classification. Hidden Markov models are introduced and applied to communications and speech recognition. This chapter is bypassed in a first course.

Chapter 10 deals with system evaluation and semi-supervised learning. The various error rate estimation techniques are discussed, and a case study with real data is treated. The leave-one-out method and the resubstitution methods are emphasized in the second semester, and students practice with computer exercises. Semi-supervised learning is bypassed in a first course.

Chapter 11 deals with the basic concepts of clustering. It focuses on definitions as well as on the major stages involved in a clustering task. The various types of data encountered in clustering applications are reviewed, and the most commonly used proximity measures are provided. In a first course, only the most widely used proximity measures are covered (e.g., l_p norms, inner product, Hamming distance).

Chapter 12 deals with sequential clustering algorithms. These include some of the simplest clustering schemes, and they are well suited for a first course to introduce students to the basics of clustering and allow them to experiment with

the computer. The sections related to estimation of the number of clusters and neural network implementations are bypassed.

Chapter 13 deals with hierarchical clustering algorithms. In a first course, only the general agglomerative scheme is considered with an emphasis on single link and complete link algorithms, based on matrix theory. Agglomerative algorithms based on graph theory concepts as well as the divisive schemes are bypassed.

Chapter 14 deals with clustering algorithms based on cost function optimization, using tools from differential calculus. Hard clustering and fuzzy and possibilistic schemes are considered, based on various types of cluster representatives, including point representatives, hyperplane representatives, and shell-shaped representatives. In a first course, most of these algorithms are bypassed, and emphasis is given to the isodata algorithm.

Chapter 15 features a high degree of modularity. It deals with clustering algorithms based on different ideas, which cannot be grouped under a single philosophy. Spectral clustering, competitive learning, branch and bound, simulated annealing, and genetic algorithms are some of the schemes treated in this chapter. These are bypassed in a first course.

Chapter 16 deals with the clustering validity stage of a clustering procedure. It contains rather advanced concepts and is omitted in a first course. Emphasis is given to the definitions of internal, external, and relative criteria and the random hypotheses used in each case. Indices, adopted in the framework of external and internal criteria, are presented, and examples are provided showing the use of these indices.

Syntactic pattern recognition methods are not treated in this book. Syntactic pattern recognition methods differ in philosophy from the methods discussed in this book and, in general, are applicable to different types of problems. In syntactic pattern recognition, the structure of the patterns is of paramount importance, and pattern recognition is performed on the basis of a set of pattern *primitives*, a set of rules in the form of a *grammar*, and a recognizer called *automaton*. Thus, we were faced with a dilemma: either to increase the size of the book substantially, or to provide a short overview (which, however, exists in a number of other books), or to omit it. The last option seemed to be the most sensible choice.