20 Classification

20.1 Introduction

When objects are detected with suitable operators and their shape is described (Chapter 19), image processing has reached its goal for certain classes of applications. For other applications, further tasks remain to be solved. In this introduction we explore several examples which illustrate how the image processing tasks depend on the questions we pose.

In many image processing applications, the size and shape of particles such as bubbles, aerosols, drops, pigment particles, or cell nuclei must be analyzed. In these cases, the parameters of interest are clearly defined and directly measurable from the images taken. We determine the area and shape of each particle detected with the methods discussed in Sections 19.5.1 and 19.3. Knowing these parameters allows all the questions of interest to be answered. From the data collected, we can, for example, compute histograms of the particle area (Fig. 20.1c). This example is typical for a wide class of scientific applications. Object parameters that can be evaluated directly and unambiguously from the image data help to answer the scientific questions asked.

Other applications are more complex in the sense that it is required to distinguish different classes of objects in an image. The easiest case is given by a typical industrial inspection task. Are the dimensions of a part within the given tolerance? Are any parts missing? Are any defects such as scratches visible? As the result of the analysis, the inspected part either passes the test or is assigned to a certain error class.

Assigning objects in images to certain classes is — like many other aspects of image processing and analysis — a truly interdisciplinary problem which is not specific to image analysis but a very general type of technique. In this respect, image analysis is part of a more general research area known as *pattern recognition*. A classical application of pattern recognition that everybody knows is *speech recognition*. The spoken words are contained in a 1-D acoustic signal (a time series). Here, the classification task is to recognize the phonemes, words, and sentences from the spoken language. The corresponding task in image processing is *text recognition*, the recognition of letters and words from a written text, also known as *optical character recognition* (*OCR*).

A general difficulty of classification is related to the fact that the relationship between the parameters of interest and the image data is

B. Jähne, Digital Image Processing
ISBN 3-540-24035-7

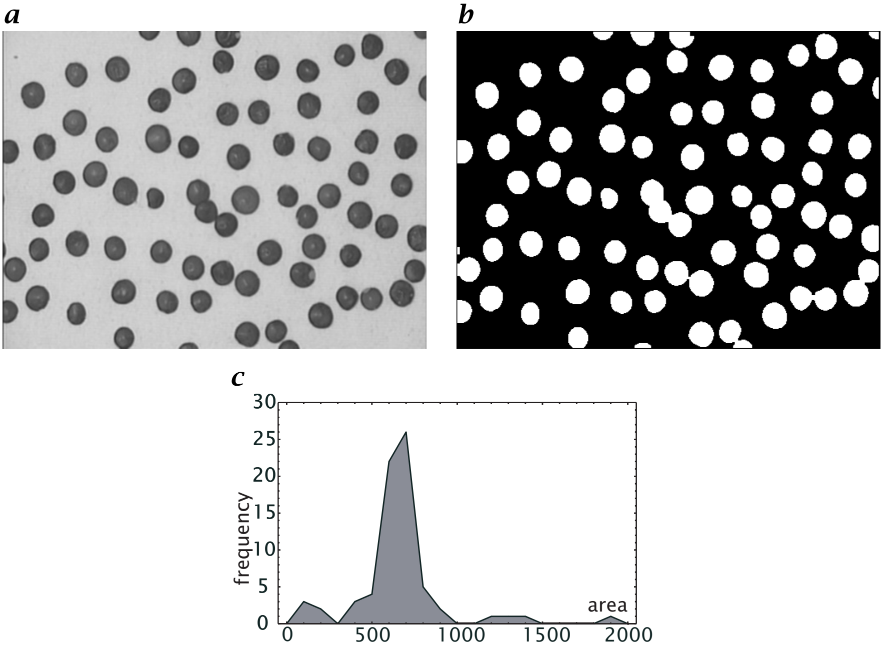

Figure 20.1: *Steps to analyze the size distribution of particles (lentils):* ***a*** *original image,* ***b*** *binary image, and* ***c*** *area distribution.*

not evident. The objects to be classified are not directly related to a certain range of values of a single feature but have to be identified by their optical signature in the image. By which features, for example, can we distinguish the lentils, peppercorns, and sunflower seeds shown in Fig. 20.2? The relation between the optical signatures and the object classes generally requires a careful investigation. We illustrate the complex relations between object features and their optical signatures with two further examples.

"*Waldsterben*" (large-scale forest damage by acid rain and other environmental pollution) is one of the many large problems environmental scientists are faced with. In remote sensing, the task is to map and classify the extent of the damage in forests from *aerial* and *satellite imagery*. In this example, the relationship between the different classes of damage and features in the images is less evident. Detailed investigations are necessary to reveal these complex relationships. Aerial images must be compared with investigations on the ground. We can expect to need more than one feature to identify certain classes of forest damage.

There are many similar applications in medical and biological science. One of the standard questions in medicine is to distinguish between "healthy" and "diseased". Again, it is obvious that we cannot expect a

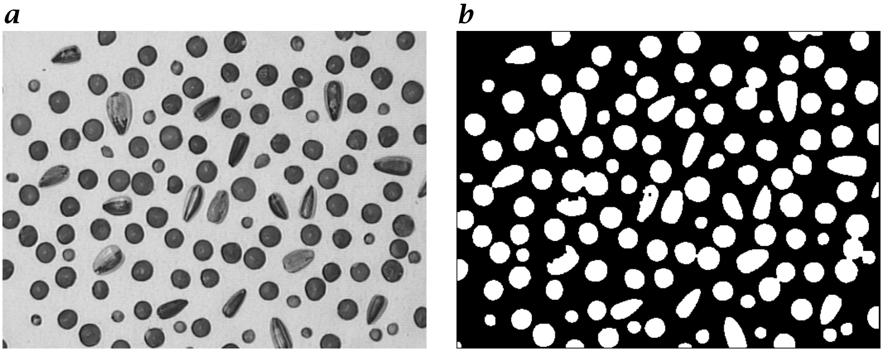

Figure 20.2: *Classification task: which of the seeds is a peppercorn, a lentil, a sunflower seed or none of the three?* ***a*** *Original image, and* ***b*** *binary image after segmentation.*

simple relationship between these two object classes and features of the observed objects in the images.

Take as another example the objects shown in Fig. 20.3. We will have no problem in recognizing that all objects but one are lamps. How could a machine vision system perform this task? What features can we extract from these images that help us recognize a lamp? While we have no problems in recognizing the lamps in Fig. 20.3, we feel quite helpless with the question as how we can solve this task using a computer. Obviously this task is complex. We recognize a lamp because we have already seen many other lamps before and somehow memorized this experience and are able to compare this stored knowledge with what we see in the image. But how is this knowledge stored and how is the comparison performed? It is obviously not just a database with geometric shapes, we also know in which context or environment lamps occur and for what they are used. Research on problems of this kind is part of a research area called *artificial intelligence*, abbreviated as *AI*.

With respect to scientific applications, another aspect of classification is of interest. As imaging techniques are among the driving forces of progress in experimental natural sciences, it often happens that unknown objects appear in images, for which no classification scheme is available so far. It is one goal of image processing to find out possible classes for these new objects. Therefore, we need classification techniques that do not require any previous knowledge.

Summing up, we conclude that classification includes two basic tasks:

1. The relation between the image features (*optical signature*) and the object classes sought must be investigated in as much detail as possible. This topic is partly comprised in the corresponding scientific

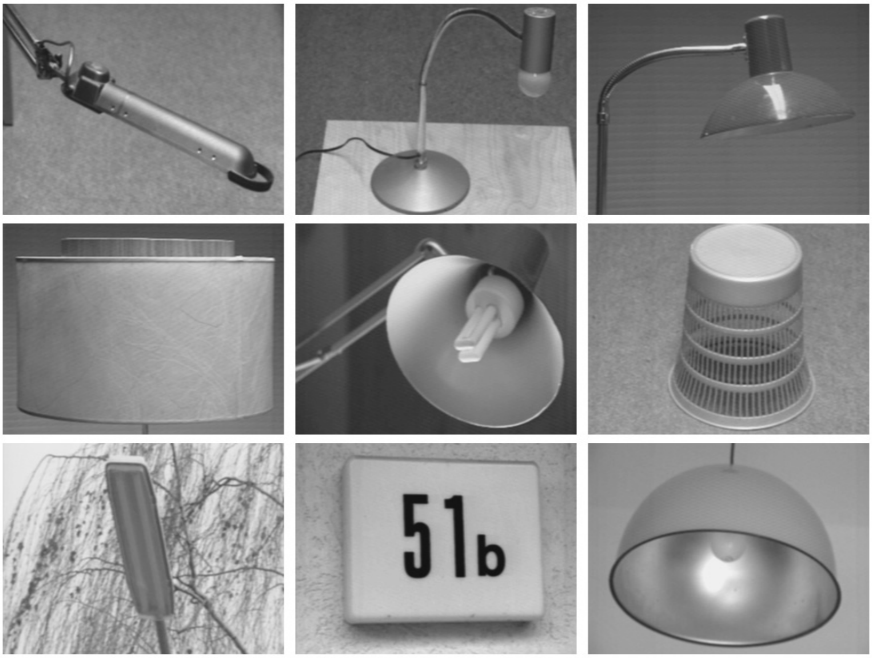

Figure 20.3: *How do we recognize that all but one of these objects are lamps?*

area and partly in image formation, i. e., optics, as discussed in Chapters 6–8.

2. From the multitude of possible image features, we must select an optimal set which allows the different object classes to be distinguished unambiguously with minimum effort and as few errors as possible by a suitable classification technique. This task, known as *classification*, is the topic of this chapter. We touch here only some basic questions such as selecting the proper type and number of features (Section 20.2) and devise some simple classification techniques (Section 20.3).

20.2 Feature Space

20.2.1 Pixel-Based Versus Object-Based Classification

Two types of classification procedures can be distinguished: *pixel-based classification* and *object-based classification*. In complex cases, a segmentation of objects is not possible using a single feature. Then it is already required to use multiple features and a classification process to decide which pixel belongs to which type of object.

A much simpler object-based classification can be used if the different objects can be well separated from the background and do not touch or overlap each other. Object-based classification should be used if at all possible, since much less data must be handled. Then all the pixel-based features discussed in Chapters 11–15, such as the mean gray value, local orientation, local wave number, and gray value variance, can be averaged over the whole area of the object and used as features describing the object's properties. In addition, we can use all the parameters describing the shape of the objects discussed in Chapter 19. Sometimes it is required to apply both classification processes: first, a pixel-based classification to separate the objects from each other and the background and, second, an object-based classification to utilize also the geometric properties of the objects for classification.

20.2.2 Cluster

A set of P features forms a P-dimensional space $\mathbb{M}$, denoted as a *feature space* or *measurement space*. Each pixel or object is represented as a *feature vector* in this space. If the features represent an object class well, all feature vectors of the objects from this class should lie close to each other in the feature space. We regard classification as a statistical process and assign a P-dimensional probability density function to each object class. In this sense, we can estimate this probability function by taking samples from a given object class, computing the feature vector, and incrementing the corresponding point in the discrete feature space. This procedure is that of a generalized P-dimensional *histogram* (Section 3.2.1). When an object class shows a narrow probability distribution in the feature space, we speak of a *cluster*. It will be possible to separate the objects into given object classes if the clusters for the different object classes are well separated from each other. With less suitable features, the clusters overlap each other or, even worse, no clusters may exist at all. In these cases, an error-free classification is not possible.

20.2.3 Feature Selection

We start with an example, the classification of the different seeds shown in Fig. 20.2 into the three classes peppercorns, lentils, and sunflower seeds. Figure 20.4a, b shows the histograms of the two features area and eccentricity (Eq. (19.6) in Section 19.3.3). While the area histogram shows two peaks, only one peak can be observed in the histogram of the eccentricity. In any case, neither of the two features alone is sufficient to distinguish the three classes peppercorns, lentils, and sunflower seeds. If we take both parameters together, we can identify at least two clusters (Fig. 20.4c). These two classes can be identified as the peppercorns and the lentils. Both seeds are almost circular and thus show a low eccen-

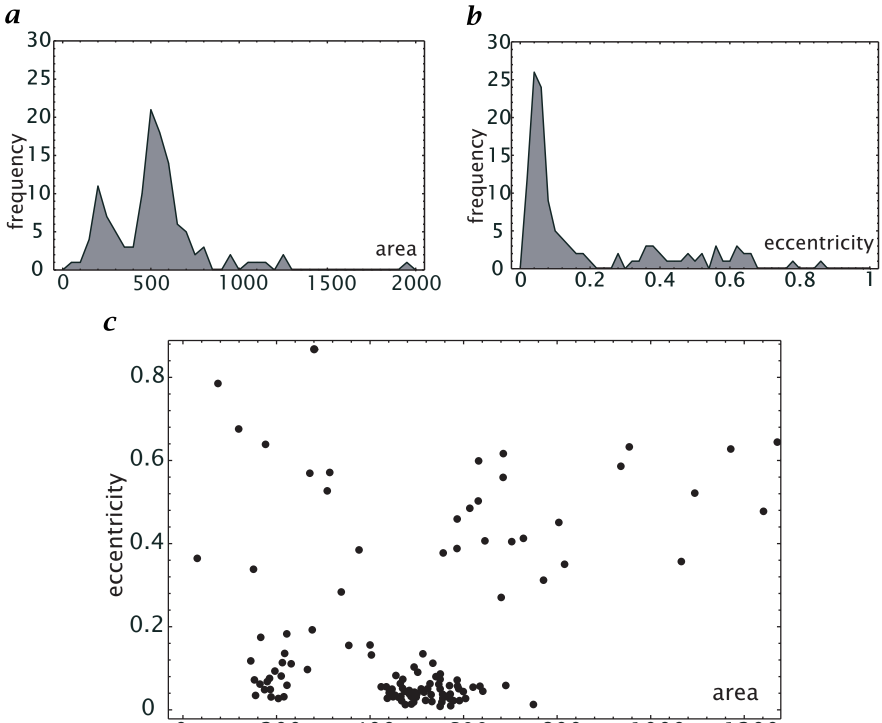

Figure 20.4: *Features for the classification of different seeds from Fig. 20.2 into peppercorns, lentils, and sunflower seeds: histogram of the features **a** area and **b** eccentricity; **c** two-dimensional feature space with both features.*

tricity between 0 and 0.2. Therefore, both classes coalesce into one peak in the eccentricity histogram (Fig. 20.4b). The sunflower seeds do not form a dense cluster since they vary significantly in shape and size. But it is obvious that they can be similar in size to the lentils. Thus it is not sufficient to use only the feature area.

In Figure 20.4c we can also identify several outliers. First, there are several small objects with high eccentricity. These are objects that are only partly visible at the edges of the image (Fig. 20.2). There are also five large objects where touching lentils merge into larger objects. The eccentricity of these objects is also large and it may be impossible to distinguish them from sunflower seeds using the two simple parameters area and eccentricity only.

The quality of the features is critical for a good classification. What does this mean? At first glance, we might think that as many features as possible would be the best solution. Generally, this is not the case. Figure 20.5a shows a one-dimensional feature space with three object

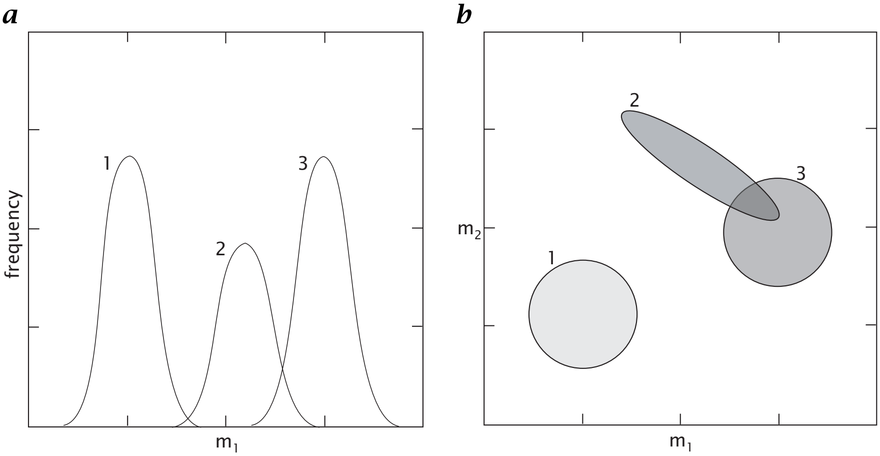

Figure 20.5: ***a*** *One-dimensional feature space with three object classes.* ***b*** *Extension of the feature space with a second feature. The gray shaded areas indicate the regions in which the probability for a certain class is larger than zero. The same object classes are shown in* ***a*** *and* ***b****.*

classes. The features of the first and second class are separated, while those of the second and third class overlap considerably. A second feature does not necessarily improve the classification, as demonstrated in Fig. 20.5b. The clusters of the second and third class are still overlaid. A closer examination of the distribution in the feature space explains why: the second feature does not tell us much new. It varies in strong correlation with the first feature. Thus, the two features are strongly correlated.

Two additional basic facts are worth mentioning. It is often overlooked how many different classes can be separated with a few parameters. Let us assume that one feature can only separate two classes. Then, ten features can separate $2^{10} = 1024$ object classes. This simple example illustrates the high separation potential of just a few parameters. The essential problem is the even distribution of the clusters in the feature space. Consequently, it is important to find the right features, i. e., to study the relationship between the features of the objects and those in the images carefully.

20.2.4 Distinction of Classes in the Feature Space

Even if we take the best features available, there may be classes that cannot be separated. In such a case, it is always worth reminding us that separating the objects in well-defined classes is only a model of reality. Often, the transition from one class to another may not be abrupt but rather gradual. For example, anomalies in a cell may be present to a vary-

1990, Oil, Island, L7R 4A6

Figure 20.6: *Illustration of the recognition of letters with very similar shape such as the large 'O' and the figure '0', or the letters 'I' and 'l' and the figure '1'.*

ing degree, there not being two distinct classes, "normal" and "pathological", but rather a continuous transition between the two. Thus, we cannot expect to find well-separated classes in the feature space in every case. We can draw two conclusions. First, it is not guaranteed that we will find well-separated classes in the feature space, even if optimal features have been selected. Second, this situation may force us to reconsider the object classification. Two object classes may either in reality be only one class or the visualization techniques to separate them may be inadequate.

In another important application, *optical character recognition*, or *OCR*, we do have distinct classes. Each character is a well-defined class. While it is easy to distinguish most letters, some, e. g., the large 'O' and the figure '0', or the letters 'I' and 'l' and the figure '1', are very similar, i. e., lie close to each other in the feature space (Fig. 20.6). Such well-defined classes that hardly differ in their features, pose serious problems for the classification task.

How can we then distinguish the large letter 'O' from the figure '0' or an 'l' from a capital 'I'? We can give two answers to this question. First, the fonts can be redesigned to make letters better distinguishable from each other. Indeed, special font sets have been designed for automated character recognition.

Second, additional information can be brought into the classification process. This requires, however, that the classification does not stop at the level of individual letters; it must be advanced to the word level. Then, it is easy to establish rules for better recognition. One simple rule which helps to distinguish the letter 'O' from the figure '0' is that letters and figures are not mixed in a word. As a counterexample to this rule, take British or Canadian zip codes which contain a blend of letters and figures. Anybody who is not trained to read this unusual mix has serious problems in reading and memorizing them. As another example, the capital 'I' can be distinguished from the lowercase 'l' by the rule that capital letters occur only as the first letter in a word or in an all-capital-letter word.

We close this section with the comment that asking whether a classification is at all possible for a given problem either by its nature or by the type of possible features is at least as important, if not more so, than the proper selection of a classification method.

20.2.5 Principal Axes Transform

The discussion in the previous section suggested that we must choose the object features very carefully. Each feature should bring in new information which is orthogonal to what we already know about the object classes; i. e., object classes with a similar distribution in one feature should differ in another feature. In other words, the features should be uncorrelated. The correlation of features can be studied with the statistical methods discussed in Section 3.3, provided that the distribution of the features for the different classes is known (supervised classification).

The important quantity is the *cross-covariance* of two features m_p and m_q from the P-dimensional feature vector for *one* object class, which is defined as

$$\sigma_{pq} = \overline{\left(m_p - \overline{m_p}\right)\left(m_q - \overline{m_q}\right)}. \tag{20.1}$$

If the cross-covariance σ_{pq} is zero, the features are said to be uncorrelated or orthogonal. The term

$$\sigma_{pp} = \overline{\left(m_p - \overline{m_p}\right)^2} \tag{20.2}$$

is a measure for the variance of the feature. A good feature for a certain object class should show a small variance indicating a narrow extension of the cluster in the corresponding direction of the feature space. With P features, we can form a symmetric matrix with the coefficients σ_{pq}, the *covariance matrix*

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \ldots & \sigma_{1,P} \\ \sigma_{12} & \sigma_{22} & \ldots & \sigma_{2,P} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1,P} & \sigma_{2,P} & \ldots & \sigma_{P,P} \end{bmatrix}. \tag{20.3}$$

The diagonal elements of the covariance matrix contain the variances of the P features, while the off-diagonal elements constitute the cross-covariances. Like every symmetric matrix, the covariance matrix can be diagonalized (Sections 3.3.2 and 13.3). This procedure is called the *principal-axes transform*. The covariance matrix in the principal-axes coordinate system reads

$$\boldsymbol{\Sigma}' = \begin{bmatrix} \sigma'_{11} & 0 & \cdots & 0 \\ 0 & \sigma'_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma'_{pp} \end{bmatrix}. \tag{20.4}$$

The diagonalization shows that we can find a new coordinate system in which all features are uncorrelated. Those new features are linear

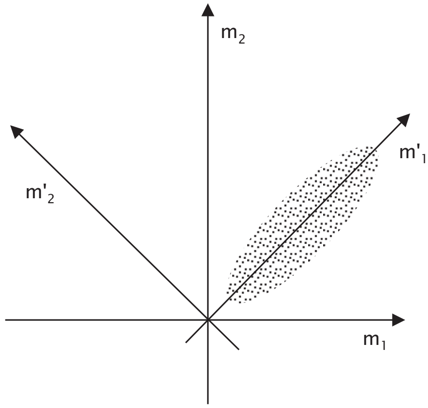

Figure 20.7: *Illustration of correlated features and the principal-axes transform.*

combinations of the old features and are the eigenvectors of the covariance matrix. The corresponding eigenvalues are the variances of the transformed features. The best features show the lowest variance; features with large variances are not of much help since they are spread out in the feature space and, thus, do not contribute much to separating different object classes. Thus, they can be omitted without making the classification significantly worse.

A trivial but illustrative example is the case when two features are nearly identical, as illustrated in Fig. 20.7. In this example, the features m_1 and m_2 for an object class are almost identical, since all points in the feature space are close to the main diagonal and both features show a large variance. In the principal-axes coordinate system $m_2' = m_1 - m_2$ is a good feature, as it shows a narrow distribution, while m_1' is as useless as m_1 and m_2 alone. Thus we can reduce the feature space from two dimensions to one without any disadvantages.

In this way, we can use the principal-axes transform to reduce the dimension of the feature space and find a smaller set of features which does nearly as good a job. This requires an analysis of the covariance matrix for all object classes. Only those features can be omitted where the analysis for *all* classes gives the same results. To avoid misunderstandings, the principal-axes transform cannot improve the separation quality. If a set of features cannot separate two classes, the same feature set transformed to the principal-axes coordinate system will not do so either. Given a set of features, we can only find an optimal subset and, thus, reduce the computational costs of classification.

20.2.6 Supervised and Unsupervised Classification

We can regard the classification problem as an analysis of the structure of the feature space. One object is thought of as a *pattern* in the feature space. Generally, we can distinguish between *supervised classification* and *unsupervised classification* procedures. Supervision of a classification procedure means determining the clusters in the feature space with known objects beforehand using teaching areas for identifying the clusters. Then, we know the number of classes and their location and extension in the feature space.

With unsupervised classification, no knowledge is presumed about the objects to be classified. We compute the patterns in the feature space from the objects we want to classify and then perform an analysis of the clusters in the feature space. In this case, we do not even know the number of classes beforehand. They result from the number of well-separated clusters in the feature space. Obviously, this method is more objective, but it may result in a less favorable separation.

Finally, we speak of *learning* methods if the feature space is updated by each new object that is classified. Learning methods can compensate any temporal trends in the object features. Such trends may be due to simple reasons such as changes in the illumination, which could easily occur in an industrial environment because of changes in daylight, aging, or dirtying of the illumination system.

20.3 Simple Classification Techniques

In this section, we will discuss different classification techniques. They can be used for both unsupervised and supervised classification. The techniques differ only by the method used to associate classes with clusters in the feature space (Section 20.2.6).

Once the clusters are identified by either method, the further classification process is identical for both of them. A new object delivers a feature vector that is associated with one of the classes or rejected as an unknown class. The different classification techniques differ only by the manner in which the clusters are modeled in the feature space.

Common to all *classifiers* is a many to one mapping from the feature space $\mathbb{M}$ to the *decision space* $\mathbb{D}$. The decision space contains Q elements, each corresponding to a class including a possible rejection class for unidentifiable objects. In the case of a deterministic decision, the elements in the decision space are binary numbers. Then only one of the elemets can be one; all others must be zero. If the classifiers generates a probabilistic decision, the elements in the decision space are real numbers. Then the sum of all elements in the decision space must be one.

20.3.1 Look-up Classification

This is the simplest classification technique but in some cases also the best, since it does not perform any modeling of the clusters for the different object classes, which can never be perfect. The basic approach of look-up classification is very simple. Take the feature space as it is and mark in every cell to which class it belongs. Normally, a significant amount of cells do not belong to any class and thus are marked with 0.

In case the clusters from two classes overlap, we have two choices. First, we can take that class which shows the higher probability at this cell. Second, we could argue that an error-free classification is not possible with this feature vector and mark the cell with zero. After this initialization of the feature space, the classification reduces to a simple look-up operation (Section 10.2.2). A feature vector m is taken and is looked up in the multidimensional look-up table to see which class, if any, it belongs to.

Without doubt, this is a fast classification technique which requires a minimum number of computations. The downside of the method — as with many other fast techniques — is that it requires huge amounts of memory for the look-up tables. An example: a three-dimensional feature space with only 64 bins per feature requires $64 \times 64 \times 64 = 1/4$ MB of memory — if no more than 255 classes are required so that one byte is sufficient to hold all class indices. We can conclude that the look-up table technique is only feasible for low-dimensional feature spaces. This suggests that it is worthwhile to reduce the number of features. Alternatively, features with a narrow distribution of feature values for all classes are useful, since then a rather small range of values and, thus, a small number of bins per feature sufficiently reduce the memory requirements.

20.3.2 Box Classification

The box classifier provides a simple modeling of the clusters in the feature space. A cluster of one class is modeled by a bounding box tightly surrounding the area covered by the cluster (Fig. 20.8). It is obvious that the box method is a rather crude modeling. If we assume that the clusters are multidimensional normal distributions, then the clusters have an elliptic shape. These ellipses fit rather well into the boxes when the axes of the ellipse are parallel to the axes of the feature space. In a two-dimensional feature space, for example, an ellipse with half-axes a and b has an area of πab, the surrounding box an area of $4ab$. This is not too bad.

When the features are correlated with each other the clusters become long and narrow objects along diagonals in the feature space. Then the boxes contain a lot of void space and they tend much more easily to overlap, making classification impossible in the overlapping areas. How-

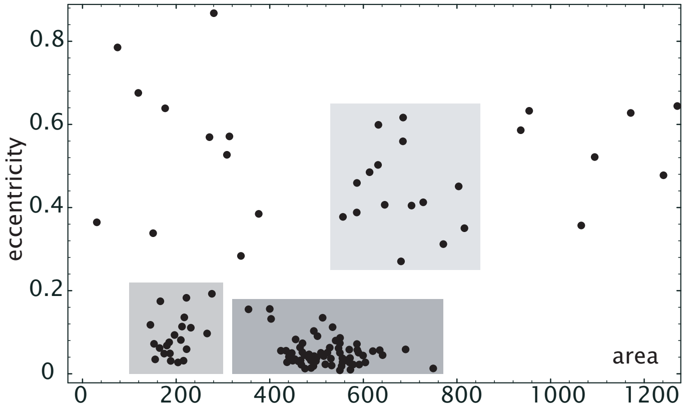

Figure 20.8: *Illustration of the box classifier for the classification of different seeds from Fig. 20.2 into peppercorns, lentils, and sunflower seeds using the two features area and eccentricity.*

Table 20.1: *Parameters and results of the simple box classification for the seeds shown in Fig. 20.2. The corresponding feature space is shown in Fig. 20.8.*

	Area	Eccentricity	Number
total	—	—	122
peppercorns	100–300	0.0–0.22	21
lentils	320–770	0.0–0.18	67
sunflower seeds	530–850	0.25–0.65	15
rejected			19

ever, correlated features can be avoided by applying of the principal-axes transform (Section 20.2.5).

The computations required for the box classifier are still modest. For each class and for each dimension of the feature space, two comparison operations must be performed to decide whether a feature vector belongs to a class or not. Thus, the maximum number of comparison operations for Q classes and a P-dimensional feature space is $2PQ$. In contrast, the look-up classifier required only P address calculations; the number of operations did not depend on the number of classes.

To conclude this section we discuss a realistic classification problem. Figure 20.2 showed an image with three different seeds, namely sunflower seeds, lentils, and peppercorns. This simple example shows many properties which are typical for a classification problem. Although the three classes are well defined, a careful consideration of the features to be used for classification is necessary since it is not immediately evi-

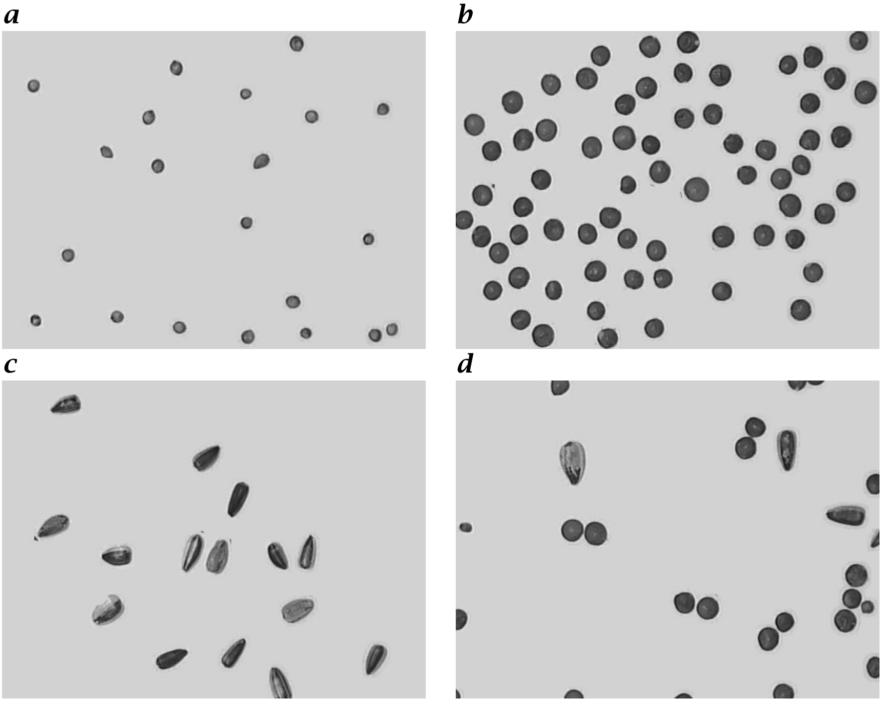

Figure 20.9: *Masked classified objects from image Fig. 20.2 showing the classified **a** peppercorns, **b** lentils, **c** sunflower seeds, and **d** rejected objects.*

dent which parameters can be successfully used to distinguish between the three classes. Furthermore, the shape of the seeds, especially the sunflower seeds, shows considerable fluctuations. The feature selection for this example was already discussed in Section 20.2.3.

Figure 20.8 illustrates the box classification using the two features area and eccentricity. The shaded rectangles mark the boxes used for the different classes. The conditions for the three boxes are summarized in Table 20.1. As the final result of the classification, Fig. 20.9 shows four images. In each of the images, only objects belonging to one of the subtotals from Table 20.1 are masked out. From a total of 122 objects, 103 objects were recognized. Thus 19 objects were rejected. They could not be assigned to any of the three classes for one of the following reasons:

- Two or more objects were so close to each other that they merged into one object. Then the values of the area and/or the eccentricity are too high.

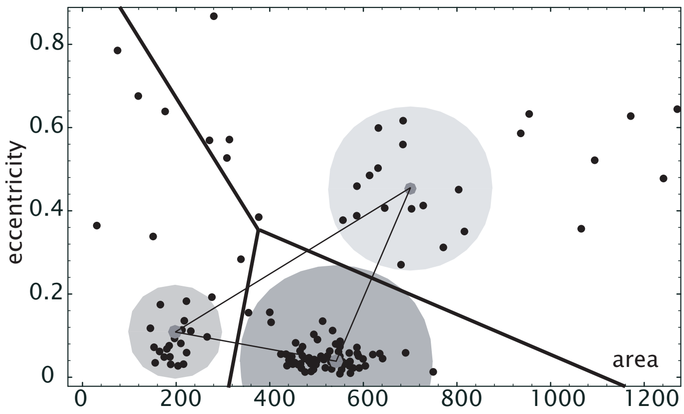

Figure 20.10: *Illustration of the minimum distance classifier with the classification of different seeds from Fig. 20.2 into peppercorns, lentils, and sunflower seeds using the two features area and eccentricity. A feature vector belongs to the cluster to which it has the minimal distance to the cluster center.*

- The object was located at the edge of the image and thus was only partly visible. This leads to objects with relatively small area but high eccentricity.
- Three large sunflower seeds were rejected because of too large an area. If we increased the area for the sunflower seed class then also merged lentils would be recognized as sunflower seeds. Thus this classification error can only be avoided if we avoid the merging of objects with a more advanced segmentation technique.

20.3.3 Minimum Distance Classification

The minimum distance classifier is another simple way to model the clusters. Each cluster is simply represented by its center of mass $\boldsymbol{m}_q$. Based on this model, a simple partition of the feature space is given by searching for the minimum distance from the feature vector to each of the classes. To perform this operation, we simply compute the distance of the feature vector $\boldsymbol{m}$ to each cluster center $\boldsymbol{m}_q$:

$$d_q^2 = |\boldsymbol{m} - \boldsymbol{m}_q|^2 = \sum_{p=1}^{P} (m_p - m_{qp})^2. \tag{20.5}$$

The feature is then assigned to the class to which it has the shortest distance.

Geometrically, this approach partitions the feature space as illustrated in Fig. 20.10. The boundaries between the clusters are hyperplanes perpendicular to the vectors connecting the cluster centers at a distance halfway between them.

The minimum distance classifier, like the box classifier, requires a number of computations that is proportional to the dimension of the feature space and the number of clusters. It is a flexible technique that can be modified in various ways.

The size of the cluster could be taken into account by introducing a scaling factor into the distance computation Eq. (20.5). In this way, a feature must be closer to a narrow cluster to be associated with it. Secondly, we can define a maximum distance for each class. If the distance of a feature is larger than the maximum distance for all clusters, the object is rejected as not belonging to any of the identified classes.

20.3.4 Maximum Likelihood Classification

The maximum likelihood classifier models the clusters as statistical probability density functions. In the simplest case, P-dimensional normal distributions are taken. Given this model, we compute for each feature vector the probability that it belongs to any of the P classes. We can then associate the feature vector with the class for which it has the maximum likelihood. The new aspect with this technique is that probabilistic decisions are possible. It is not required that we decide to put an object into a certain class. We can simply give the object probabilities for membership in the different classes.

20.4 *Exercises*

20.1: Elementary classification methods

Interactive demonstration of elementary classification methods (dip6ex20.01)

20.2: *__Classes and features__

Given below are some typical classification tasks. Compare them by answering the following questions:

1. How many classes do the classification problems have?
2. Are the different classes clearly separated from each other or is there a potential overlap?
3. Does a hierarchical class structure exist?
4. What could be potential features that are suitable to distinguish the different classes?

Here are the classification tasks:

A Images were taken from bubbles, submerged into the water by breaking waves. The goal is to measure the size distribution of the bubbles.

B The task is to distinguish tumor cells from healthy cells in microscopic cell images.

C The task is to distinguish distant point-like objects into stars, galaxies, and quasars using telescope images. The images were taken in 10 to 12 spectral channels range from the visible to the near infrared.

D Optical character recognition (OCR): an automatic imaging system should read numbers on forms automatically containing the numeric characters 0 to 9, the decimal point, and the plus and minus signs.

E The task is to generate land usage maps in order to distinguish building areas, streets, forests, fields, etc.

Problem 20.3: ***Storage needs and computational effort**

Compare the storage needs and the computational effort for the following classification tasks. Assume that you have 4 features with a resolution of 6 bits and four known classes. The classification techniques are:

1. Lookup method
2. Box method
3. Method of minimum distance
4. Method of maximum likelihood

20.5 *Further Readings*

Classification was discussed in this chapter only in an introductory way without the whole theoretical background. Interested readers, who like to deepen their knowledge in this area, are referred to some more advanced literature. From the vast amount of literature about classification, we mention only a few textbooks and monographs here. Two of the most successful textbooks are Duda et al. [40] and Webb [214]. Both textbooks emphasize statistical approaches. The book of Schürmann [184] shows in a unique way the common concepts of classification techniques based on classical statistical techniques and on neural networks. The application of *neural networks* for classification is detailed by Bishop [11]. One of the most recent advances in classification, the so-called *support vector machines*, are very readably introduced by Christianini and Shawe-Taylor [24] and Schöllkopf and Smola [182].

Part V

Reference Part