# Chapter 2
# Cluster Analysis

**Abstract** This chapter outlines the major steps of cluster analysis. It starts with an informal introduction to clustering, its tools, methodology and applications. Then it proceeds with formalising the problem of data clustering. Diverse measures of object similarity, based on quantitative (like numerical measurement results) and on qualitative features (like text) as well as on their mixtures, are described. Various variants, how such similarity measures can be exploited when defining clustering cost functions are presented. Afterwards, major brands of clustering algorithms are explained, including hierarchical, partitional, relational, graph-based, density-based and kernel methods. The underlying data representations and interrelationships between various methodologies are discussed. Also possibilities of combining several algorithms for analysis of the same data (ensemble clustering) are presented. Finally, the issues related to easiness/hardness of the data clustering tasks are recalled.

Cluster analysis consists in distinguishing, in the set of analysed data, the groups, called clusters. These groups are disjoint[1] subsets of the data set, having such a property that data belonging to different clusters differ among themselves much more than the data, belonging to the same cluster. The role of cluster analysis is, therefore, to uncover a certain kind of natural structure in the data set. The means enabling performing that task is constituted usually by a certain measure of similarity or dissimilarity—the issue is discussed further on in Sect. 2.2. Cluster analysis is not only an important cognitive tool, but, as well, a method for reducing large sets of data, since it allows for the replacement of a group of data by its compact characterisation, like, e.g. the centre of gravity of the given group.

The task of cluster analysis can be perceived as a problem of grouping of objects according to their mutual similarity. Objects, which are mutually similar in a sufficiently high degree, form a homogeneous group (a cluster). It is also possible to consider the similarity of objects to certain characteristic entities (called prototypes) of the classes. In this case we deal more with the problem of classification, that is—of finding the model patterns—see [164]. Yet, if the characteristics corresponding to

[1]The requirement of disjoint subsets is used in the classical data analysis. In the general case, the groups distinguished might constitute the *coverage* of the data set. This is the case, for instance, in the fuzzy data analysis.



S. T. Wierzchoń and M. A. Kłopotek, *Modern Algorithms of Cluster Analysis*,
Studies in Big Data 34, https://doi.org/10.1007/978-3-319-69308-8_2

classes are not given a priori, they should be established. And this is exactly what cluster analysis is about.

The term *data clustering* (meaning grouping of data) appeared for the first time in 1954 in the title of a paper concerning the analysis of anthropological data [272, p. 653]. Other equivalent names, given cluster analysis, are *Q-analysis*, *typology*, *clumping*, and *taxonomy* [273], depending on the domain, in which clustering is applied. There is a number of very good books, devoted to cluster analysis. The classical ones include[2]: [19, 164, 173, 273, 284, 436, 463]. Among the more recent and specialised monographs we can mention:[9, 70, 73, 75, 140, 297, 387, 402, 453]. A reader, interested in survey works may wish to have a look at, e.g., [66, 272, 274, 515]. More advanced techniques of cluster analysis are considered, in particular, in [176].

It is worth noting that both in common language and in computer science there exists a bunch of synonyms or closely related concepts for the term *cluster analysis* which reveal various aspects of this term. One of them is the *unsupervised learning* or *learning without a teacher*. This term suggests that we are looking for a hidden structure behind the data that we want to reveal. It suggests also that it must pay off to have recovered such a structure that is there must exist a criterion saying whether or not a useful structure was discovered. Last not least the hidden structure has to be learnable from the data in the sense of learnability theory. One speaks frequently about *client segmentation*, *image segmentation* etc. Image segmentation means that we look for a particular structure behind the image data—a trace of one or more physical objects, to be separated from the background. Criteria of belonging to the same object may include local properties, like continuity of colour, shading, texture, or global ones, like alignment along a line of limited curvature etc. Client segmentation on the other hand tries to split the population into homogeneous sections concentrated around some centric concepts so that predominantly global quality criteria will be of interest. And of course one asks the representativeness of the client population via the split into categories (generalization capability), easiness of assignment of new client to an existent category (mutual prediction of attributes within a category) and about profitability of cluster assignment from the business point of view (a kind of the cluster separation criterion). The frequently used term *taxonomy formation* points at the need of not only splitting the objects into classes or categories, but also of providing with a simple, compact description. Hence it is important to note that we are not talking just about clustering or grouping of objects, but rather the term *analysis* should be stressed because various cluster analysis methods reveal in the data structures of different types and the user of such methods either should be aware of what kind of hidden information he is looking for so that he chooses appropriate methods or he should be aware of what type of output he got if he applied a bunch of various methods for exploratory purposes.

For these reasons, the authors of this book do not try to point out the best method for clustering, neither explicitly nor implicitly, but rather concentrate on pointing

[2]Many of those listed have been modified several times over and successive editions have been published.

at particular advantages of individual methods and methodologies while covering a broad spectrum of potential and actual applications of cluster analysis.

It is the researcher himself who needs to know his goals and to choose the appropriate clustering approach that fits best his purposes. In order to help to understand the importance of user's goals, let us recall the once famous "Ugly Duckling Theorem" of Watanabe[3] Assume objects in a collection are described by binary attributes taking on values "true" and "false". Assume that two objects are more similar if they share more "true"-valued attributes. Furthermore, let us include in object description not only the original attributes but also their derivatives (obtained via logical operations "and", "or", "not"). Under these circumstances any object is similar to any other so that no clustering makes sense. This fact has a number of other consequences, which are frequently ignored: dropping/selecting/weighting appropriate subset of these attributes may lead to any similarity structure we wish to have. It means also that the very same set of objects may have different meaningful clusterings depending on the purpose of clustering and that application of an irrelevant clustering algorithm may lead to completely useless cluster sets. Therefore before starting cluster analysis, we need to have clear criteria on how we evaluate the quality of a single cluster (homogeneity) and/or the quality of the interrelationships between clusters (separation), and of course, what similarity/dissimilarity of objects means in our application area.

So e.g. when applying cluster analysis for purposes of delimitation of species of plants or animals in biology, the choice of algorithms and attributes will be driven by the assumed definition of species, e.g. via the requirement of (direct or indirect) interbreeding capability. While there is a limited amount of experimental data on interbreeding, the clustering performed on other available attributes is considered correct if it matches the known interbreeding possibilities and/or restrictions. Medical classification of diseases would be validated on/driven by a compromise between common symptoms, known causes and treatment possibilities. Image segmentation will understand clusters of pixels as homogeneous if they belong to the same organ, blood vessel, building, connected area covered with same plants etc. Market segmentation will be successful if we discover groups of clients worthy/not worthy addressing in marketing campaigns. Geographical client segmentation for purposes of store chains may be evaluated on the discovery of new profitable locations of stores etc. Clustering in databases aims at efficient organization of data for query processing.

Whatever the domain-specific expectations, one may apply cluster analysis with diverse goals of later usage of the results. One may perform exploratory data analysis seeking interesting patterns just to identify hypotheses worthy further investigation. One may be interested in data reduction (compression) and structuring for efficient storage and access. Or one may look for a clustering that aligns with other groupings to investigate their pertinence or universality.

[3] Watanabe, Satosi (1969). Knowing and Guessing: A Quantitative Study of Inference and Information. New York: Wiley, pp. 376–377.

Note also that depending on the application, the criteria for homogeneity and separation may be diverse. Let us just mention a few:

- homogeneity criteria
  - small within-cluster dissimilarities,
  - clusters fitting some homogeneous probability models like the Gaussian or a uniform distribution on a convex set, or some functional, time series or spatial process models,
  - members of a cluster well represented by its centroid,
  - clusters corresponding to connected areas in data space with high density,
  - clusters fitting into convex shapes (e.g. balls)
  - clusters characterisable by a small number of variables.
  - features approximately independent within clusters;
- separation criteria
  - large between-cluster dissimilarities
  - dissimilarity matrix of the data reflected by the clustering
  - constraints on co-occurrence/not co-occurrence in the same cluster matched
  - cluster stability (e.g. under resampling, bootstrapping)
  - low number of clusters
  - clusters of roughly the same size
  - balanced distances between cluster centres along a minimum weight spanning tree
  - clusters corresponding well to an externally given partition or values of one or more variables that were not used for computing the clustering
  - clusters fitting partial labelling
  - learnability of cluster membership from data
  - for hierarchical clusters the distance between cluster members reflected by the hierarchy level when they join the same cluster

The above criteria serve only as examples of possible cluster/clustering quality and whether or not any, some or large portion of them is actually used in a concrete application case will depend on the interests of the researcher.

We would further need to specify, in what sense a given criterion should be satisfied, e.g. by just passing a threshold or by reaching a local/global optimum or deviate only by some percentage from such an optimum. If we define multiple optimality criteria, we would need to reconcile them by stating a strategy of compromise between them. These and various other aspects of clustering philosophy are discussed in-depth in the paper [250] by Hennig.

## 2.1 Formalising the Problem

It is usually assumed in cluster analysis, that a set of $m$ objects $\mathfrak{X} = \{\mathfrak{x}_1, \ldots, \mathfrak{x}_m\}$ is given.

For purposes of analysis the set of objects may be characterised by an embedded or relational representation.

In the *embedded* representation, every object is described by an $n$-dimensional vector $\mathbf{x}_i = (x_{i1} \ldots x_{in})^{\mathrm{T}}$, where $x_{ij}$ denotes the value of the $j$-th feature of the object $\mathfrak{x}_i$. The vector $\mathbf{x}_i$ is being called feature vector or image.[4]

The subject of cluster analysis is constituted, therefore, not so much by the original set of objects $\mathfrak{X}$, as by its representation, given through the matrix $X = (\mathbf{x}_1 \ldots \mathbf{x}_m)^{\mathrm{T}}$, whose $i$-th row is a vector of features, describing the $i$-th object. In view of the fact that to object $\mathfrak{x}_i$ corresponds the $i$-th row of matrix $X$, the term "object" shall be used to denote both the element $\mathfrak{x}_i \in \mathfrak{X}$ and the vector of feature values $\mathbf{x}_i$, characterising this object. In statistics, vector $\mathbf{x}_i$ is called ($n$-dimensional) observation. Even though the values of individual measurements of feature values might be expressed in different scales (nominal, ordinal or quotient), the majority of practical and theoretical results have been obtained under the assumption that the components of vectors $\mathbf{x}_i$ are real numbers. Thus, for a vast majority of cases, considered in the present book, we shall be assuming that the observations are given by the vectors $\mathbf{x}_i \in \mathbb{R}^n$.

This kind of data representation will be explored in the current chapter as well as in Chaps. 3 and 4.

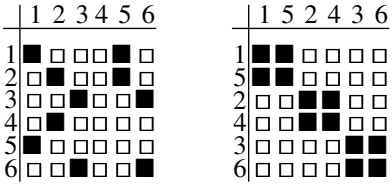
It is alternatively assumed (in the *relational* representation) that information on the data set is provided in the form of the matrix $S$ of dimension $m \times m$, the elements of this matrix $s_{ij}$ represent similarities (or dissimilarities) for the pairs of objects $\mathfrak{x}_i$, $\mathfrak{x}_j$. When making use of such a representation, we can give up the requirement of having the features, describing the objects, measured on the quantitative scales. The pioneer of such a perspective was Polish anthropologist, ethnographer, demographer and statistician—Jan Czekanowski.[5] The method, developed by Czekanowski, and presented in the first handbook of modern methods of data analysis and interpretation of its results [129], consists in replacing numbers in the matrix $S$ by the appropriately selected graphical symbols. In this manner, an unordered diagram (called Czekanowski's diagram) arises, which, after an adequate reordering of rows and columns of the matrix, makes apparent the existence of groups of objects mutually similar. Extensive description of his idea provide Graham and Hell in [213]. To illustrate this approach consider a small matrix

[4] The latter term is particularly justified, when we treat the measurements as mappings $\mathbf{f}: \mathfrak{X} \to \mathbb{R}^n$ of the set of objects into a certain set of values. Then, $\mathbf{x}_i = \mathbf{f}(\mathfrak{x}_i)$, and in the mathematical nomenclature $\mathbf{x}_i$ is the image of the object $\mathfrak{x}_i$.

[5] See J. Gajek. Jan Czekanowski. Sylwetka uczonego. *Nauka Polska*, 6(2), 1958, 118–127.

$$\nabla = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1.0 & 0.2 & 0.1 & 0.3 & 0.8 & 0.4 \\ 0.2 & 1.0 & 0.1 & 0.9 & 0.3 & 0.3 \\ 0.1 & 0.1 & 1.0 & 0.2 & 0.2 & 0.7 \\ 0.3 & 0.9 & 0.2 & 1.0 & 0.4 & 0.1 \\ 0.8 & 0.3 & 0.2 & 0.4 & 1.0 & 0.2 \\ 0.4 & 0.3 & 0.7 & 0.1 & 0.2 & 1.0 \end{bmatrix} \end{array}$$

and let us denote by □ the degrees smaller than 0.5 and by ■ the degrees greater than 0.5. Then the above matrix can be represented graphically as in left part of Eq. (2.1). After reordering its rows and columns we obtain the matrix depicted in right part of the Eq. (2.1) revelling three clusters: {1, 5}, {2, 4} and {3}.




(2.1)

Although the method was developed originally more than 100 years ago, it is still in use, mainly in archaeology,[6] in economic sciences,[7] and even in musicology. Using the method of Czekanowski, the mathematicians from Wrocław: K. Florek, J. Łukaszewicz, J. Perkal, H. Steinhaus and S. Zubrzycki, elaborated the so-called "Wrocław taxonomy", which they presented in 1957 in 17th issue of the journal *Przegląd Antropologiczny* (see also [515]). In further parts of the book we present other methods of analysing the matrix of similarities. More advanced considerations of application of the similarity matrix in cluster analysis have been presented in the references [44, 45]. This type of representation is addressed primarily in Chaps. 5 and 6.

The role of the classical cluster analysis is to split the set of objects (observations) into $k < m$ groups $\mathcal{C} = \{C_1, \ldots, C_k\}$, where each $i$-th group $C_i$ is called cluster. Such a division fulfils three natural requirements:

(i) Each cluster ought to contain at least one object, $C_j \neq \emptyset$, $j = 1, \ldots, k$.
(ii) Each object ought to belong to a certain cluster, $\bigcup_{j=1}^{k} C_j = \mathfrak{X}$.
(iii) Each object ought to belong to exactly one cluster, $C_{j_1} \cap C_{j_2} = \emptyset$, $j_1 \neq j_2$.

[6] See, e.g., A. Soltysiak, and P. Jaskulski. Czekanowski's Diagram: A method of multidimensional clustering. In: J.A. Barceló, I. Briz and A. Vila (eds.) *New Techniques for Old Times. CAA98. Computer Applications and Quantitative Methods in Archaeology*. Proc. of the 26th Conf., Barcelona, March 1998 (BAR International Series 757). Archaeopress, Oxford 1999, pp. 175–184.

[7] See, e.g., A. Wójcik. Zastosowanie diagramu Czekanowskiego do badania podobieństwa krajów Unii Europejskiej pod względem pozyskiwania energii ze źródeł odnawialnych. *Zarządzanie i Finanse* (*J. of Management and Finance*), 11(4/4), 353–365, 2013, http://zif.wzr.pl/pim/2013_4_4_25.pdf.

In particular, when $k = m$, each cluster contains exactly one element from the set $\mathfrak{X}$. This partition is trivial, and so we shall be considering the cases, in which $k$ is much smaller than $m$.

An exemplary illustration of the problem that we deal with in cluster analysis, is provided in the Fig. 2.1. The "clouds" of objects, situated in the lower left and upper right corners constitute distinctly separated clusters. The remaining objects form three, two, or one cluster, depending on how we define the notion of similarity between the objects.

Cluster analysis is also referred to as unsupervised learning. We lack here, namely, the information on the membership of the objects in classes, and it is not known, how many classes there should really be. Even though the mechanical application of the algorithms, which are presented in the further parts of the book allows for the division of any arbitrary set into a given number of classes, the partition thus obtained may not have any sense. Assume that $\mathfrak{X}$ is a set of points selected conform to the uniform distribution from the set $[0, 1] \times [0, 1]$—see Fig. 2.2a. Mechanical application of an algorithm of grouping, with the predefined parameter $k = 3$ leads to the result, shown in Fig. 2.2b. It is obvious that although the partition obtained fulfils the conditions set before, it has no sense.

In informal terms, the presence of a structure in a data set is manifested through the existence of separate areas, and hence of clusters, enjoying such a property that any two objects, belonging to the common cluster $C_i$ are more mutually similar than any two objects, picked from two different clusters, i.e.

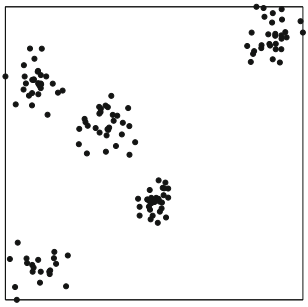


**Fig. 2.1** The task of cluster analysis: to break down the set of observations into $k$ disjoint subsets, composed of similar elements

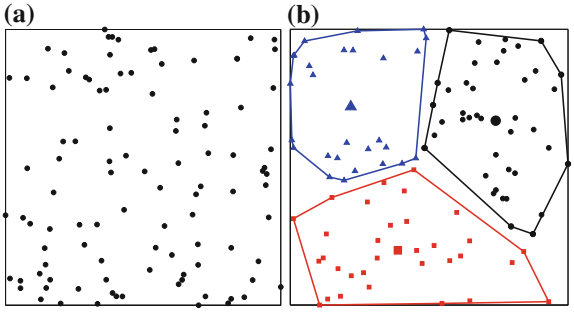



**Fig. 2.2** Grouping of the set of randomly generated objects. **a** 100 points randomly selected from the set $[0, 1] \times [0, 1]$. **b** Division of the objects into three groups using the $k$-means algorithm. Bigger marks indicate the geometrical centers of groups

$$s(\mathfrak{x}', \mathfrak{x}'') > s(\mathfrak{y}', \mathfrak{y}'')$$

if only $\mathfrak{x}', \mathfrak{x}'' \in C_i$, $\mathfrak{y}' \in C_{j_1}$, $\mathfrak{y}'' \in C_{j_2}$ and $j_1 \neq j_2$. Symbol $s$ denotes here a certain measure of similarity, that is, a mapping $s\colon \mathfrak{X} \times \mathfrak{X} \to \mathbb{R}$. In many situations it is more convenient to make use of the notion of dissimilarity (like, e.g., distance) and require the objects, belonging to different clusters, to be more distant than the objects, belonging to the same cluster. Various definitions of the measures of similarity or dissimilarity are considered in the subsequent Sect. 2.2. The choice of the appropriate measure constitutes an additional factor of complexity of the data analysis task. Yet another such factor is the choice of an adequate criterion for determining the partition of the set $\mathfrak{X}$, that is stating whether or not a given partition would be satisfactory or not. For this purpose qualitative criteria listed on p. 12 need to be chosen and formalised for the purposes of a particular clustering task.

The majority of the methods of grouping consists in an "intelligent" extraction of information from the matrix $S$, with elements representing similarity or dissimilarity of the pairs of objects. An excellent example of this kind of procedure is provided by the hierarchical methods, shortly presented in Sect. 2.3 or by the spectral methods, being the primary subject of Chap. 5.

The most popular methods of grouping include the hierarchical methods, the combinatorial methods (referred also to as relocation-based or partitional), the density-based methods, the grid methods (being a brand of density based ones) and the methods based on models. The descriptions of these methods and comments on them can be found in numerous survey studies, like, e.g. [272, 274, 515]. In the further part of the chapter we shall present their short characteristics. The rapidly developing spectral methods will be presented in Chap. 5.

## 2.2 Measures of Similarity/Dissimilarity

In order to be able to quantify associations between pairs of objects, a measure of similarity $s\colon \mathfrak{X} \times \mathfrak{X} \to \mathbb{R}$ or of dissimilarity is introduced. The two measures are, in principle, dual, that is—the lower the value of dissimilarity, the more similar the two compared objects. A particular example of dissimilarity is distance (metric), that is, a function $d\colon \mathfrak{X} \times \mathfrak{X} \to \mathbb{R}_+ \cup \{0\}$ fulfilling three conditions:

(a) $d(\mathfrak{x}, \mathfrak{y}) = 0$ if and only if $\mathfrak{x} \equiv \mathfrak{y}$,
(b) $d(\mathfrak{x}, \mathfrak{y}) = d(\mathfrak{y}, \mathfrak{x})$ (symmetry),
(c) $d(\mathfrak{x}, \mathfrak{y}) \leq d(\mathfrak{x}, \mathfrak{z}) + d(\mathfrak{z}, \mathfrak{y})$ (triangle inequality),

for arbitrary $\mathfrak{x}, \mathfrak{y}, \mathfrak{z} \in \mathfrak{X}$. When only conditions (b) and (c) are satisfied, then $d$ is called pseudo-distance.[8]

[8]Note that, as Gower et al. [209] states, see their Theorem 1, any non-metric dissimilarity measure $d(\mathfrak{z}, \mathfrak{y})$ for $\mathfrak{z}, \mathfrak{y} \in \mathfrak{X}$ where $\mathfrak{X}$ is finite, can be turned into a (metric) distance function $d'(\mathfrak{z}, \mathfrak{y}) = d(\mathfrak{z}, \mathfrak{y}) + c$ where $c$ is a constant where $c \geq \max_{\mathfrak{x},\mathfrak{y},\mathfrak{z}\in\mathfrak{X}} \|d(\mathfrak{x}, \mathfrak{y}) + d(\mathfrak{y}, \mathfrak{z}) - d(\mathfrak{z}, \mathfrak{x})\|$.

For instance, if $d_{max}$ denotes the maximum value of distance between the pairs of objects from the set $\mathfrak{X}$, then distance can be transformed into a measure of similarity (proximity) $s(\mathfrak{x}_i, \mathfrak{x}_j) = d_{max} - d(\mathfrak{x}_i, \mathfrak{x}_j)$. The thus obtained measure of proximity attains the maximum values, when $i = j$ (object $\mathfrak{x}_i$ is identical with itself), and the lower the value of this measure, the less mutually similar (more dissimilar) the objects compared are.

In the above example, the maximum value of similarity is the number $d_{max} = s(\mathfrak{x}_i, \mathfrak{x}_i)$. It is more convenient to operate with the normalised similarity $s(\mathfrak{x}_i, \mathfrak{x}_j) = 1 - d(\mathfrak{x}_i, \mathfrak{x}_j)/d_{max}$.

Another, frequently applied example of the measure of similarity is provided by the transformation $s(\mathfrak{x}_i, \mathfrak{x}_j) = \exp[-d^2(\mathfrak{x}_i, \mathfrak{x}_j)/\sigma^2]$, where $\sigma > 0$ is a parameter, or, yet, $s(\mathfrak{x}_i, \mathfrak{x}_j) = 1/[d(\mathfrak{x}_i, \mathfrak{x}_j) + \epsilon]$, where $\epsilon > 0$ is a small number.[9]

Generally, if $f : \mathbb{R} \to \mathbb{R}$ is a monotonously decreasing function, such that $f(0) > 0$ and $\lim_{\xi \to \infty} f(\xi) = a \geq 0$, then

$$s(\mathfrak{x}_i, \mathfrak{x}_j) = f(d(\mathfrak{x}_i, \mathfrak{x}_j)) \tag{2.2}$$

is a measure of similarity, induced by the distance $d$.

The condition $a \geq 0$, given above, is, in principle, not necessary; it is introduced in order to preserve the symmetry with the non-negative values of the distance measure $d$. Note also that if $d$ is a distance, then the measure of similarity, defined through the transformation $f$, referred to above, fulfils the triangle condition in the form—see [110]:

$$s(\mathfrak{x}, \mathfrak{y}) + s(\mathfrak{y}, \mathfrak{z}) \leq s(\mathfrak{x}, \mathfrak{z}) + s(\mathfrak{y}, \mathfrak{y})$$

if function $f$ is *additionally* convex.[10]

The result can be derived as follows: Either (1) $d(\mathfrak{x}, \mathfrak{z}) \leq d(\mathfrak{x}, \mathfrak{y})$ or (2) this is not true but $d(\mathfrak{x}, \mathfrak{z}) \leq d(\mathfrak{y}, \mathfrak{z})$ or (3) $d(\mathfrak{x}, \mathfrak{z}) > d(\mathfrak{x}, \mathfrak{y})$ & $d(\mathfrak{x}, \mathfrak{z}) > d(\mathfrak{y}, \mathfrak{z})$ holds. In the first case $f(d(\mathfrak{x}, \mathfrak{z})) \geq f(d(\mathfrak{x}, \mathfrak{y}))$. As $0 \leq d(\mathfrak{y}, \mathfrak{z})$, we have $f(0) \geq f(d(\mathfrak{y}, \mathfrak{z}))$. But summing up both we get $f(d(\mathfrak{y}, \mathfrak{y})) + f(d(\mathfrak{x}, \mathfrak{z})) \geq f(d(\mathfrak{x}, \mathfrak{y})) + f(d(\mathfrak{y}, \mathfrak{z}))$. So the claim is proven. In the second case the reasoning is analogous (just flip $\mathfrak{x}$ and $\mathfrak{z}$).

So let us consider the third case when $d(\mathfrak{x}, \mathfrak{z}) > d(\mathfrak{x}, \mathfrak{y})\&d(\mathfrak{x}, \mathfrak{z}) > d(\mathfrak{y}, \mathfrak{z})$. As metric is assumed, $d(\mathfrak{x}, \mathfrak{z}) \leq d(\mathfrak{x}, \mathfrak{y}) + d(\mathfrak{y}, \mathfrak{z})$. Hence $1 \leq \frac{d(\mathfrak{y},\mathfrak{z})}{d(\mathfrak{x},\mathfrak{z})} + \frac{d(\mathfrak{x},\mathfrak{y})}{d(\mathfrak{x},\mathfrak{z})}$. Therefore $0 \leq 1 - \frac{d(\mathfrak{y},\mathfrak{z})}{d(\mathfrak{x},\mathfrak{z})} \leq \frac{d(\mathfrak{x},\mathfrak{y})}{d(\mathfrak{x},\mathfrak{z})} \leq 1$. Let us pick any $\lambda \in \left[0, \frac{d(\mathfrak{x},\mathfrak{y})}{d(\mathfrak{x},\mathfrak{z})}\right]$ such that $1 - \frac{d(\mathfrak{y},\mathfrak{z})}{d(\mathfrak{x},\mathfrak{z})} \leq \lambda$. Apparently $0 \leq \lambda \leq 1$. We see immediately that $\lambda d(\mathfrak{x}, \mathfrak{z}) \leq d(\mathfrak{x}, \mathfrak{y})$ and $(1 - \lambda)d(\mathfrak{x}, \mathfrak{z}) \leq d(\mathfrak{y}, \mathfrak{z})$ From the convexity definition we have that $(1 - \lambda)f(0) + \lambda f(d(\mathfrak{x}, \mathfrak{z})) \geq f((1 - \lambda) \cdot 0 + \lambda d(\mathfrak{x}, \mathfrak{z})) = f(\lambda d(\mathfrak{x}, \mathfrak{z})) \geq f(d(\mathfrak{x}, \mathfrak{y}))$, with the last inequality being due to the decreasing monotonicity of $f$.

---

[9] In practice, a small number means one that is small compared to distances but still numerically significant under the available machine precision.

[10] A real-valued function $f(x)$ is said to be convex over the interval $[a, b] \subset \mathbb{R}$ if for any $x_1, x_2 \in [a, b]$ and any $\lambda \in [0, 1]$ we have $\lambda f(x_1) + (1 - \lambda) f(x_2) \geq f(\lambda x_1 + (1 - \lambda)x_2)$. All three mentioned examples of transformation of a distance into similarity measure are in fact convex functions.

Similarly $\lambda \cdot f(0) + (1-\lambda) f(d(\mathfrak{x}, \mathfrak{z})) \geq f(\lambda \cdot 0 + (1-\lambda) d(\mathfrak{x}, \mathfrak{z})) = f((1-\lambda) d(\mathfrak{x}, \mathfrak{z})) \geq f(d(\mathfrak{y}, \mathfrak{z}))$

By summing these two inequalities we get $f(0) + f(d(\mathfrak{x}, \mathfrak{z})) \geq f(d(\mathfrak{x}, \mathfrak{y}) + f(d(\mathfrak{y}, \mathfrak{z})$ so obviously the triangle inequality holds here too.

This derivation by the way differs a bit from [110] but follows the general idea contained therein.

Since the measures of similarity and dissimilarity are dual notions, see, e.g., [110], we shall be dealing in further course primarily with various measures of dissimilarity, and in particular—with distances.[11] Making use of distances requires having the possibility of assigning to each object $\mathfrak{x}_i$ its representation $\mathbf{x}_i$. An interesting and exhaustive survey of various measures of similarity/dissimilarity can be found, for instance, in [100].[12]

### 2.2.1 Comparing the Objects Having Quantitative Features

When all the features, which are used to describe objects from the set $\mathfrak{X}$ are quantitative, then every object $\mathfrak{x}_i \in \mathfrak{X}$ is identified with an $n$-dimensional vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})^{\mathrm{T}}$. The most popular measure of dissimilarity is the Euclidean distance

$$d(\mathfrak{x}_i, \mathfrak{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{l=1}^{n} (x_{il} - x_{jl})^2}$$

or, more generally, the norm defined by the square form

$$d_W(\mathfrak{x}_i, \mathfrak{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_W = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}} W (\mathbf{x}_i - \mathbf{x}_j)} \tag{2.3}$$

where $W$ is a positive definite matrix of the dimensions $n \times n$.

If $W$ is a unit matrix, then Eq. (2.3) defines the Euclidean distance. If, on the other hand, $W$ is a diagonal matrix having the elements

$$w_{ij} = \begin{cases} \omega_i & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

then (2.3) defines the weighted Euclidean distance, i.e.

$$d_W(\mathfrak{x}_i, \mathfrak{x}_j) = \sqrt{\sum_{l=1}^{n} \omega_l (x_{il} - x_{jl})^2} = \sqrt{\sum_{l=1}^{n} (y_{il} - y_{jl})^2}$$

[11] In Chaps. 5 and 6 we will predominantly concentrate on similarity measure based clustering methods.

[12] In Sect. 6.2 we discuss some similarity measures defined for data in form of graphs/networks.

where $y_{il} = \sqrt{\omega_l} x_{il}$ is the weighted value of the feature $l$, measured for the $i$-th object.

The Euclidean distance is being generalized in various manners. These most commonly used are commented upon below.[13]

#### 2.2.1.1 Minkowski Distance

Minkowski distance (norm) is defined as follows

$$d_p(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_p = \Big[\sum_{l=1}^{n} |x_{il} - x_{jl}|^p\Big]^{1/p}, \quad p \geq 1, p \in \mathbb{R} \tag{2.4}$$

When we take $p = 1$, we obtain the city block distance (called also taxicab or Manhattan distance)

$$d_1(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_1 = \sum_{l=1}^{n} |x_{il} - x_{jl}| \tag{2.5}$$

For $p = 2$, Eq. (2.4) defines the Euclidean distance. In view of the popularity of this distance definition, we shall be writing $\|\mathbf{x}_i - \mathbf{x}_j\|$ instead of $\|\mathbf{x}_i - \mathbf{x}_j\|_2$.

Finally, when $p = \infty$, we get Chebyshev distance

$$d_\infty(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_\infty = \max_{l=1,\ldots,n} |x_{il} - x_{jl}| \tag{2.6}$$

Minkowski distance is used not only in exact sciences, but also in psychology,[14] industrial design and generally in designing. Unit circles are described in Minkowski metric by the equation

$$|x|^p + |y|^p = 1$$

which is also called the curve (or oval) of Lamé. The respective shapes for three values of the parameter $p$ are presented in Fig. 2.3a. Danish mathematician Piet Hein[15] concluded that the case of $p = 2.5$ leads to the shape, featuring *high aesthetic qualities*, see Fig. 2.3b, this fact having been made use of in designing Sergels roundabout in Stockholm.

---

[13] Note that the Euclidean distance has been investigated itself to a great depth, see e.g. [209]. See Sect. B.5 for a discussion of criteria of a dissimilarity matrix being Euclidean distance matrix. Gower et al. [209] points out that any dissimilarity matrix $D$ may be turned to an Euclidean distance matrix, see their Theorem 7, by adding an appropriate constant, e.g. $d'(\mathfrak{z}, \mathfrak{y}) = \sqrt{d(\mathfrak{z}, \mathfrak{y})^2 + h}$ where $h$ is a constant such that $h \geq -\lambda_m$, $\lambda_m$ being the smallest eigenvalue of $(\mathbf{I} - \mathbf{1}\mathbf{1}^T/m)(-1/2D_{sq})(\mathbf{I} - \mathbf{1}\mathbf{1}^T/m)$, $D_{sq}$ is the matrix of squared values of elements of $D$, $m$ is the number of rows/columns in $D$.

[14] See Chap. 3 in: C.H. Coombs, R.M. Dawes, A. Tversky. *Mathematical Psychology: An Elementary Introduction*. Prentice Hall, Englewood Cliffs, NJ 1970.

[15] His profile can be found on the website http://www.piethein.com/.

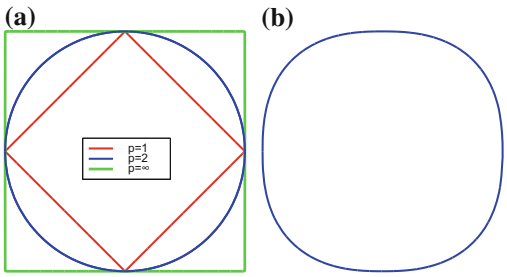



**Fig. 2.3** Unit circles for the selected values of the parameter $p$: (a) $p = 1, 2, \infty$, (b) $p = 2.5$ (Hein's super-ellipse)

Distances, deriving from the Minkowski metric, have two important shortcomings. First, as the dimensionality of the problem increases, the difference between the close and the far points in the space $\mathbb{R}^n$ disappears, this being the effect of summing of the differences in the locations of objects in the particular dimensions.[16]

More precisely, the situation is as follows. Denote by $d_{p,n}^{min}$, $d_{p,n}^{max}$, respectively, the minimum and the maximum values of distance (measured with distance $d_p$) between two arbitrary points, selected from the set of $m$ randomly generated points in $n$-dimensional space. Then, see [8]

$$C_p \leq \lim_{n\to\infty} \mathbb{E}\left[\frac{d_{p,n}^{max} - d_{p,n}^{min}}{n^{1/p-1/2}}\right] \leq (m-1)C_p \tag{2.7}$$

where $C_p$ is a constant, depending upon the value of $p$, and $\mathbb{E}$ denotes the expected value. This inequality implies that in a high-dimensional space the difference $d_{p,n}^{max} - d_{p,n}^{min}$ increases proportionally to $n^{1/p-1/2}$ irrespective of the distribution of data [8]. This property plays a dominating role, when $n \geq 15$. In particular (see Fig. 2.4a)

$$d_{p,n}^{max} - d_{p,n}^{min} \to \begin{cases} C_1\sqrt{n} & \text{if } p = 1 \\ C_2 & \text{if } p = 2 \\ 0 & \text{if } p \geq 3 \end{cases}$$

In order to prevent this phenomenon, Aggarwal, Hinnenburg and Keim proposed in [8] application of the fractional Minkowski distances with the parameter $p \in (0, 1]$—see Fig. 2.4b. Yet, in this case (2.4) is no longer a distance, since the triangle condition is not satisfied. If, for instance, $\mathbf{x} = (0, 0)$, $\mathbf{y} = (1, 1)$ and $\mathbf{z} = (1, 0)$, then

$$d(\mathbf{x}, \mathbf{y}) = 2^{1/p} > d(\mathbf{x}, \mathbf{z}) + d(\mathbf{y}, \mathbf{z}) = 1 + 1$$

A subsequent, but not so critical issue is constituted by the fact that the values of the Minkowski metric are dominated by these features, whose values are measured on the scales with the biggest ranges. This issue can be relatively easily resolved by introducing weighted distance, that is—by replacing each component of the

[16] Equivalently, one can say that two arbitrary vectors in $\mathbb{R}^n$ are orthogonal [402, p. 7.1.3].

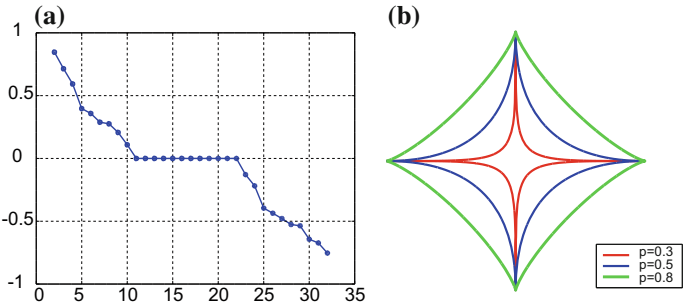



**Fig. 2.4** The influence of the parameter $p$ on the properties of Minkowski distance. **a** Average values of the difference between the most distant points from a 100-point set in dependence upon the number of dimensions, $n$, and the value of the exponent, $p$. **b** Unit circles for $p < 1$

Eq. (2.4) by the expression $\omega_l(x_{il} - x_{jl})^p$, where $w_l$ is the weight equal, e.g., the inverse of the standard deviation of the $l$-th feature, or the inverse of the range of variability of the $l$-th feature. The counterpart to the second variant is constituted by the initial normalization of data, ensuring that $x_{il} \in [0, 1]$ for each of the features $l = 1, \ldots, n$. This is, usually, a routine procedure, preceding the proper data analysis. In some cases, instead of a simple normalization, standardization is applied, that is—the original value $x_{il}$ is replaced by the quotient $(x_{il} - \mu_l)/\sigma_l$, where $\mu_l, \sigma_l$ are the average value and the standard deviation of the $l$-th feature.

#### 2.2.1.2 Mahalanobis Distance

When defining distance (2.4) it is by default assumed that features are not mutually correlated. When this assumption is not satisfied, Mahalanobis distance is usually applied,

$$d_\Sigma(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}} \Sigma^{-1} (\mathbf{x}_i - \mathbf{x}_j)} \tag{2.8}$$

which is a variant of the distance (2.3), where $W$ is equal the inverse of the covariance matrix. Covariance matrix is calculated in the following manner

$$\mathbf{\Sigma} = \frac{1}{m} \sum_{i=1}^{m} (\mathbf{x}_i - \overline{\boldsymbol{\mu}})(\mathbf{x}_i - \overline{\boldsymbol{\mu}})^{\mathrm{T}} \tag{2.9}$$

with

$$\overline{\boldsymbol{\mu}} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i \tag{2.10}$$

being the vector of average values.

Note that:

(a) By applying the transformation $\mathbf{y}_i = \Sigma^{-1/2}\mathbf{x}_i$ we reduce Mahalanobis distance $d_\Sigma(\mathbf{x}_i, \mathbf{x}_j)$ to Euclidean distance between the transformed vectors, that is $d_\Sigma(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{y}_i - \mathbf{y}_j\|$.
(b) When the features are independent, then the covariance matrix is a diagonal matrix: the nonzero elements are equal to the variances of the particular features. In such a case Mahalanobis distance becomes the weighted Euclidean distance of the form

$$d_S(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{l=1}^{n} \left(\frac{x_{il} - x_{jl}}{\sigma_l}\right)^2} \tag{2.11}$$

Mahalanobis distance is useful in identification of the *outliers* (atypical observations). A number of properties of this distance are provided in the tutorial [341].

#### 2.2.1.3 Bregman Divergence

The distances, that is—the measures of dissimilarity—considered up till now, have a differential character: $d(\mathbf{x}, \mathbf{y}) = \varphi(\mathbf{x} - \mathbf{y})$, where $\varphi\colon \mathbb{R}^n \to \mathbb{R}$ is an appropriately selected function. In certain situations, e.g. in problems concerning signal compression, measures are needed that would account for more complex relations between the vectors compared. An exhaustive survey thereof is given in [54]. An instance is represented in this context by the measure, introduced for the systems of speech compression by Chaffee[17]

$$d_{Ch}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^{\mathrm{T}} R(\mathbf{x})(\mathbf{x} - \mathbf{y}) \tag{2.12}$$

While in the case of Mahalanobis distance (2.8) the covariance matrix, which appears there, is established a priori, the matrix of weights, which is used in the Definition (2.12) depends upon the currently considered object $\mathbf{x}$. Another example is provided by the measure of Itakura-Saito.[18]

All these measures are generalized by the so-called Bregman divergence. It is defined as follows [50].

**Definition 2.2.1** Let $\phi\colon S \to \mathbb{R}$ be a strictly convex function, defined on a convex set $S \subset \mathbb{R}^n$. Besides, we assume that the relative interior, $rint(S)$, of the set $S$ is not

[17] D.L. Chaffee, Applications of rate distortion theory to the bandwidth compression of speech, Ph.D. dissertation, Univ. California, Los Angeles, 1975. See also R.M. Gray, et al., Distortion measures for speech processing, *IEEE Trans. on Acoustics, Speech and Signal Processing*, **28**(4), 367–376, Aug. 1980.

[18] See F. Itakura, S. Saito, Analysis synthesis telephony based upon maximum likelihood method, *Repts. of the 6th Intl. Cong. Acoust*. Tokyo, C-5-5, C-17-20, 1968.

**Table 2.1** Bregman divergences generated by various convex functions [50]

| Domain | $\phi(\mathbf{x})$ | $d_\phi(\mathbf{x}, \mathbf{y})$ | Divergence |
|---|---|---|---|
| $\mathbb{R}$ | $\mathbf{x}^2$ | $(\mathbf{x}-\mathbf{y})^2$ | Quadratic loss function |
| $\mathbb{R}_+$ | $\mathbf{x}\log\mathbf{x}$ | $\mathbf{x}\log(\frac{\mathbf{x}}{\mathbf{y}}) - (\mathbf{x}-\mathbf{y})$ | |
| [0,1] | $\mathbf{x}\log\mathbf{x} + (1-\mathbf{x})\log(1-\mathbf{x})$ | $\mathbf{x}\log(\frac{\mathbf{x}}{\mathbf{y}}) + (1-\mathbf{x})\log(\frac{1-\mathbf{x}}{1-\mathbf{y}})$ | Logistic loss function |
| $\mathbb{R}_{++}$ | $-\log\mathbf{x}$ | $\frac{\mathbf{x}}{\mathbf{y}} - \log(\frac{\mathbf{x}}{\mathbf{y}}) - 1$ | Itakura-Saito distance |
| $\mathbb{R}_+^n$ | $\sum_{j=1}^n \left(\mathbf{x}_j \log\mathbf{x}_j - \mathbf{x}_j\right)$ | $\sum_{j=1}^n \left(\mathbf{x}_j \log(\frac{\mathbf{x}_j}{\mathbf{y}_j}) - (\mathbf{x}_j - \mathbf{y}_j\right)$ | Generalized I-divergence |
| $\mathbb{R}^n$ | $\|\mathbf{x}\|^2$ | $\|\mathbf{x}-\mathbf{y}\|^2$ | Squared Euclidean distance |
| $\mathbb{R}^n$ | $\mathbf{x}^{\mathrm{T}} W x$ | $(\mathbf{x}-\mathbf{y})^{\mathrm{T}} W(\mathbf{x}-\mathbf{y})$ | Mahalanobis distance |
| $n$-simplex | $\sum_{j=1}^n \mathbf{x}_j \log_2 \mathbf{x}_j$ | $\sum_{j=1}^n \mathbf{x}_j \log_2(\frac{x_j}{y_j})$ | KL-divergence |

empty, and $\phi$ is a function differentiable on $rint(S)$. Bregman divergence is then such a function $d_\phi \colon S \times rint(S) \to [0, \infty)$ that for the arbitrary $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$

$$d_\phi(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) - \phi(\mathbf{y}) - (\mathbf{x}-\mathbf{y})^{\mathrm{T}} \nabla\phi(\mathbf{y}) \tag{2.13}$$

Symbol $\nabla\phi(\mathbf{y})$ denotes the gradient of the function $\phi(\mathbf{y})$. □

Examples of Bregman divergence are shown in Table 2.1. Special attention ought to be paid to the last three examples. If we take for $\phi(\mathbf{x})$ the squared length of vector $\mathbf{x}$, then $d_\phi(\mathbf{x}, \mathbf{y})$ is equal squared Euclidean distance between the points, represented by the vectors $\mathbf{x}$ and $\mathbf{y}$. If the mapping $\phi$ is defined as $\mathbf{x}^{\mathrm{T}} W \mathbf{x}$, where $W$ is a positive definite matrix, then we obtain the squared distance (2.3), in particular—defined in Sect. 2.2.1.2 Mahalanobis distance. Finally, if $\mathbf{x}$ is a stochastic vector,[19] while $\phi$ is a negative value of entropy, $\phi(\mathbf{x}) = \sum_{j=1}^n x_j \log_2 x_j$, then we obtain Kullback-Leibler divergence, referred to frequently as KL-divergence. This notion plays an important role in information theory, machine learning, and in information retrieval.

A closely related concept is the *Bregman information* of a set $X$ of data points $\phi(\mathbf{x}_j)$, drawn from a probability distribution $\pi$.

$$I_\phi(X) = \min_{\mathbf{y} \in \mathbb{R}^n} \sum_{j=1}^m \pi_j d_\phi(\mathbf{x}_j, \mathbf{y})$$

The most interesting property of Bregman information is that the $\mathbf{y}$ minimising $I_\phi(X)$ does not depend on the particular Bregman divergence $\phi$ and is always equal

[19] That is—all of its components are non-negative and $\sum_{j=1}^n x_j = 1$.

to $\boldsymbol{\mu} = \sum_{j=1}^{m} \pi_j \mathbf{x}_j$ that is the mean value vector of the data set. As we will see, this can be used in construction of clustering algorithms.[20]

Define $J_\phi(\mathbf{y}) = \sum_{j=1}^{m} \pi_j d_\phi(\mathbf{x}_j, \mathbf{y})$. Then $I_\phi = \min_{\mathbf{y}} J_\phi(\mathbf{y})$. Let us now compute

$$
\begin{aligned}
J_\phi(\mathbf{y}) - J_\phi(\boldsymbol{\mu}) &= \sum_{j=1}^{m} \pi_j d_\phi(\mathbf{x}_j, \mathbf{y}) - \sum_{j=1}^{m} \pi_j d_\phi(\mathbf{x}_j, \boldsymbol{\mu}) \\
&= \sum_{j=1}^{m} \left(\pi_j \left(\phi(\mathbf{x}_j) - \phi(\mathbf{y}) - (\mathbf{x}_j - \mathbf{y})^{\mathrm{T}} \nabla\phi(\mathbf{y})\right)\right) \\
&- \sum_{j=1}^{m} \left(\pi_j \left(\phi(\mathbf{x}_j) - \phi(\boldsymbol{\mu}) - (\mathbf{x}_j - \boldsymbol{\mu})^{\mathrm{T}} \nabla\phi(\boldsymbol{\mu})\right)\right) \\
&= \sum_{j=1}^{m} \pi_j \phi(\mathbf{x}_j) - \sum_{j=1}^{m} \pi_j \phi(\mathbf{y}) - \sum_{j=1}^{m} \pi_j \mathbf{x}_j^{\mathrm{T}} \nabla\phi(\mathbf{y}) + \sum_{j=1}^{m} \pi_j \mathbf{y}^{\mathrm{T}} \nabla\phi(\mathbf{y}) \\
&- \sum_{j=1}^{m} \pi_j \phi(\mathbf{x}_j) + \sum_{j=1}^{m} \pi_j \phi(\boldsymbol{\mu}) + \sum_{j=1}^{m} \pi_j \mathbf{x}_j^{\mathrm{T}} \nabla\phi(\boldsymbol{\mu}) - \sum_{j=1}^{m} \pi_j \boldsymbol{\mu}^{\mathrm{T}} \nabla\phi(\boldsymbol{\mu}) \\
&= \sum_{j=1}^{m} \pi_j \phi(\mathbf{x}_j) - \phi(\mathbf{y}) - \sum_{j=1}^{m} \pi_j \mathbf{x}_j^{\mathrm{T}} \nabla\phi(\mathbf{y}) + \mathbf{y}^{\mathrm{T}} \nabla\phi(\mathbf{y}) - \sum_{j=1}^{m} \pi_j \phi(\mathbf{x}_j) + \phi(\boldsymbol{\mu}) \\
&+ \sum_{j=1}^{m} \pi_j \mathbf{x}_j^{\mathrm{T}} \nabla\phi(\boldsymbol{\mu}) - \boldsymbol{\mu}^{\mathrm{T}} \nabla\phi(\boldsymbol{\mu}) \\
&= -\boldsymbol{\mu}^{\mathrm{T}} \nabla\phi(\boldsymbol{\mu}) + \mathbf{y}^{\mathrm{T}} \nabla\phi(\mathbf{y}) - \phi(\mathbf{y}) + \sum_{j=1}^{m} \pi_j \phi(\mathbf{x}_j) - \sum_{j=1}^{m} \pi_j \mathbf{x}_j^{\mathrm{T}} \nabla\phi(\mathbf{y}) \\
&- \sum_{j=1}^{m} \pi_j \phi(\mathbf{x}_j) + \phi(\boldsymbol{\mu}) + \sum_{j=1}^{m} \pi_j \mathbf{x}_j^{\mathrm{T}} \nabla\phi(\boldsymbol{\mu}) \\
&= +\phi(\boldsymbol{\mu}) - \phi(\mathbf{y}) + \sum_{j=1}^{m} \pi_j \mathbf{x}_j^{\mathrm{T}} \nabla\phi(\boldsymbol{\mu}) - \boldsymbol{\mu}^{\mathrm{T}} \nabla\phi(\boldsymbol{\mu}) \\
&+ \mathbf{y}^{\mathrm{T}} \nabla\phi(\mathbf{y}) - \sum_{j=1}^{m} \pi_j \mathbf{x}_j^{\mathrm{T}} \nabla\phi(\mathbf{y}) + \sum_{j=1}^{m} \pi_j \phi(\mathbf{x}_j) - \sum_{j=1}^{m} \pi_j \phi(\mathbf{x}_j) \\
&= +\phi(\boldsymbol{\mu}) - \phi(\mathbf{y}) + \left(\left(\sum_{j=1}^{m} \pi_j \mathbf{x}_j\right) - \boldsymbol{\mu}\right)^{\mathrm{T}} \nabla\phi(\boldsymbol{\mu}) \\
&+ \left(\mathbf{y} - \left(\sum_{j=1}^{m} \pi_j \mathbf{x}_j\right)\right)^{\mathrm{T}} \nabla\phi(\mathbf{y})
\end{aligned}
$$

[20] Note that Bregman divergence is in general neither symmetric nor fits the triangle condition hence it is not a metric distance. However, there exist some families of Bregman distances, like Generalized Symmetrized Bregman and Jensen-Bregman divergence, which under some conditiomns are a square of a metric distance, and may be even embedded in Euclidean space. For details see Acharyya, S., Banerjee, A., and Boley, D. (2013). Bregman Divergences and Triangle Inequality. In SDM'13 (SIAM International Conference on Data Mining), pp. 476–484.

$$
\begin{aligned}
&= +\phi(\boldsymbol{\mu}) - \phi(\mathbf{y}) + (\boldsymbol{\mu} - \boldsymbol{\mu})^{\mathrm{T}} \nabla\phi(\boldsymbol{\mu}) + (\mathbf{y} - \boldsymbol{\mu})^{\mathrm{T}} \nabla\phi(\mathbf{y}) \\
&= +\phi(\boldsymbol{\mu}) - \phi(\mathbf{y}) - (\boldsymbol{\mu} - \mathbf{y})^{\mathrm{T}} \nabla\phi(\mathbf{y}) \\
&= d_{\phi}(\boldsymbol{\mu}, \mathbf{y}) \geq 0
\end{aligned} \tag{2.14}
$$

So we see that $J_{\phi}(\mathbf{y}) \geq J_{\phi}(\boldsymbol{\mu})$ hence $\boldsymbol{\mu}$ minimises $J_{\phi}$.

#### 2.2.1.4 Cosine Distance

Another manner of coping with the "curse of dimensionality", pointed at in Sect. 2.2.1.1, is suggested by, for instance, Hamerly [229], namely by introducing distance $d_{cos}(\mathbf{x}_i, \mathbf{x}_j)$, defined as 1 minus cosine of the angle between the vectors $\mathbf{x}_i, \mathbf{x}_j$,

$$
d_{cos}(\mathbf{x}_i, \mathbf{x}_j) = 1 - \cos(\mathbf{x}_i, \mathbf{x}_j) = 1 - \frac{\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|} = 1 - \frac{\sum_{l=1}^{n} x_{il} x_{jl}}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|} \tag{2.15}
$$

The value of cosine of an angle, appearing in the above formula, constitutes an example of a measure of similarity, which is the basic measure, applied in the information retrieval systems for measuring the similarity between the documents [39]. In this context the components $\mathbf{x}_{il}$ represent the frequency of appearance of a keyword indexed $l$ in the $i$-th document. Since frequencies are non-negative, then, for two arbitrary vectors, representing documents, $0 \leq \cos(\mathbf{x}_i, \mathbf{x}_j) \leq 1$ and $0 \leq d_{cos}(\mathbf{x}_i, \mathbf{x}_j) \leq 1$. Other measures, which quantify the similarity of documents, are considered in [259].

Note that $d_{cos}(\mathbf{x}_i, \mathbf{x}_j)$ is not metric (the triangle condition does not hold). In order to obtain a metric distance measure, the $d_{arccos}(\mathbf{x}_i, \mathbf{x}_j) = \arccos(\cos(\mathbf{x}_i, \mathbf{x}_j))$ distance is introduced, which is simply the angle between the respective vectors. Other metric possibility is the sine distance, $d_{sin}(\mathbf{x}_i, \mathbf{x}_j) = \sin(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{1 - \cos^2(\mathbf{x}_i, \mathbf{x}_j)}$.

Still another cosine similarity based metric distance is $d_{sqrtcos}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{1 - \cos(\mathbf{x}_i, \mathbf{x}_j)}$.

#### 2.2.1.5 Power Distance

When we wish to increase or decrease the growing weight, assigned to these dimensions, for which the objects considered differ very much, we can apply the so-called power distance[21]:

$$
d_{p,r}(\mathbf{x}_i, \mathbf{x}_j) = \left( \sum_{l=1}^{n} |x_{il} - x_{jl}|^p \right)^{1/r} \tag{2.16}
$$

[21] See e.g. *Web-based handbook of statistics. Cluster analysis: Agglomeration.* http://www.statsoft.pl/textbook/stathome.html.

where $p$ and $r$ are parameters. The parameter $p$ controls the increasing weight, which is assigned to the differences for the particular dimensions, while parameter $r$ controls the increasing weight, which is assigned to the bigger differences between objects. Of course, when $p = r$, this distance is equivalent to Minkowski distance.

In general, power distance is not metric.

### 2.2.2 Comparing the Objects Having Qualitative Features

Similarly as in the preceding point, we assume here, that for reasons related to the facility of processing, each object is represented by the vector $\mathbf{x}$, but now its components are interpreted more like labels. If, for instance, the feature of interest for us is *eye color*, then this feature may take on such values as 1–blue, 2–green, etc. It is essential that for such labels there may not exist a natural order of such label "values", proper for a given phenomenon under consideration. In such situations one can use as the measure of dissimilarity the generalized Hamming distance: $d_H(\mathbf{x}_i, \mathbf{x}_j)$, equal the number of these features, whose values for the compared objects are different.

When we deal with mixed data, that is—a part of features have a qualitative character, and a part—quantitative character, then we can apply the so-called Gower coefficient [303], which is the weighted sum of the partial coefficients of divergence $\delta(i, j, l)$, determined for each feature of the objects $\mathbf{x}_i$ i $\mathbf{x}_j$. For a nominal (qualitative) feature we take $\delta(i, j, l) = 1$, when the value of this feature in both objects is different and $\delta(i, j, l) = 0$ in the opposite case. If, on the other hand, we deal with the quantitative feature having values from the interval $[x_l^{min}, x_l^{max}]$, then we replace $\delta(i, j, l)$ by

$$\delta(i, j, l) = \frac{|x_{il} - x_{jl}|}{x_l^{max} - x_l^{min}}$$

Ultimately, we take as the distance between two objects

$$d_m(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{l=1}^{n} w(i, j, l)\delta(i, j, l)}{\sum_{l=1}^{n} w(i, j, l)} \tag{2.17}$$

where $w(i, j, l)$ is the weight equal zero when either one of the values $x_{il}$, $x_{jl}$ was not observed, or we deal with a so-called asymmetric binary feature[22] and for one of the compared objects the value of this feature is equal zero. In the remaining cases we have $w(i, j, l) = 1$.

The recommender systems [89], whose purpose is to suggest to the users the selection of broadly understood goods (books, movies, discs, information, etc.) matching in a possibly best manner the tastes and the preferences of the users, make use of

[22] E.g. in medical tests it is often assumed that lack of a given feature for a patient is denoted by symbol 0, while its presence—by the symbol 1. In such situations it is better not to account in the comparisons for the number of zeroes.

the similarity measure $s(\mathbf{x}_i, \mathbf{x}_j)$ between the preferences of the given user, $\mathbf{x}_i$, and the preferences of other users, $\mathbf{x}_j$, where $j = 1, \ldots, m$, $j \neq i$. Here, the $l$-th component of the preference vector corresponds to the evaluation of the $l$-th good. One of the most often applied measures of similarity is, in this case, the modified Pearson correlation coefficient

$$r(\mathbf{x}_i, \mathbf{x}_j) = \frac{(\mathbf{x}_i - \overline{\mathbf{x}}_i) \cdot (\mathbf{x}_j - \overline{\mathbf{x}}_j)}{\|\mathbf{x}_i - \overline{\mathbf{x}}_i\| \|\mathbf{x}_j - \overline{\mathbf{x}}_j\|} \tag{2.18}$$

where $\overline{\mathbf{x}}_i$ denotes the average of the vector $\mathbf{x}_i$. Modification concerns the numerator of the above expression: when summing up the respective products one accounts for only those components of the vectors, which represent the common evaluations of the users compared. Instead of Pearson correlation one can apply, of course, other measures of correlation, adapted to the character of the features used in describing objects. The most popular variants applied for the qualitative features are Spearman or Kendall correlations.

Just like in the case of the cosine similarity measure, also here one can introduce the correlation-based distance $d_{r1}(\mathbf{x}_i, \mathbf{x}_j) = 1 - r(\mathbf{x}_i, \mathbf{x}_j)$, taking values from the interval $[0, 2]$. Another variant was proposed in [467]: $d_{r2}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{2[1 - r(\mathbf{x}_i, \mathbf{x}_j)]}$. This distance also takes values from the interval $[0, 2]$; the positively and strongly correlated variables correspond to small distances, while negatively and strongly correlated variables correspond to large distances, the weakly correlated variables being situated midway. One should also note that when we deal with the centered variables (i.e. $\overline{\mathbf{x}}_i = \overline{\mathbf{x}}_j = 0$), then the value of the correlation coefficient is identical with the value of cosine of the angle between the two vectors. This fact was taken advantage of by Trosset [467] to formulate the angular representation of correlation, used then to construct a new algorithm of cluster analysis. Finally, in [203], side by side with $d_{r2}$, the (pseudo-)distance

$$d_{r3}(\mathbf{x}_i, \mathbf{x}_j) = \left(\frac{1 - r(\mathbf{x}_i, \mathbf{x}_j)}{1 + r(\mathbf{x}_i, \mathbf{x}_j)}\right)^{\beta}$$

where $\beta > 0$ is a parameter, was introduced. This distance was applied in the FCM algorithm (which is presented in Sect. 3.3). The coefficient $\beta$ controls, in this case, the degree of fuzziness of the resulting partition. When $r(\mathbf{x}_i, \mathbf{x}_j) = -1$, then the distance is undefined.

Since the value of $r$ represents the cosine of the angle between the (centered) vectors $\mathbf{x}_i, \mathbf{x}_j$, then

$$d_{\tan}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\frac{1 - r^2(\mathbf{x}_i, \mathbf{x}_j)}{r^2(\mathbf{x}_i, \mathbf{x}_j)}} \tag{2.19}$$

can be treated as tangent distance. The tangent distance measure, $d_{\tan}$, is based on the volume of joint information, carried by the features analysed. The parallel vectors ($r = 1$), as well as the anti-parallel ones ($r = -1$) carry the very same information,

and hence their distance is equal 0, while similarity is the highest and equal 1. On the other hand, the orthogonal vectors are infinitely distant and have zero similarity, since each of them carries entirely different information. Vectors, having correlation coefficients different from 0 and $\pm 1$ contain partly a specific information, and partly common information. The volume of the common information constitutes the measure of their similarity. The tangent measure of distance has nowadays a wide application in the analysis of similarity.[23] Another, normalized variant of the correlation-based similarity measure is

$$r_n(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}\Big(r(\mathbf{x}_i, \mathbf{x}_j) + 1\Big) \tag{2.20}$$

which guarantees that $r_n(\mathbf{x}_i, \mathbf{x}_j) \in [0, 1]$. In bioinformatics the so-called squared correlation distance is being applied $d_b(\mathbf{x}_i, \mathbf{x}_j) = 1 - r^2(\mathbf{x}_i, r_j)$ when comparing genetic profiles.[24] One can find a number of interesting comments on the properties of the correlation coefficient in [407].

Pearson correlation is an adequate yardstick when the variables compared are normally distributed. When this is not the case, other measures of similarity ought to be applied.

Let us also mention the chi-square distance, which is defined as follows

$$d_{\chi^2}(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\sum_{i=1}^{n}\frac{(x_i - y_i)^2}{x_i + y_i} \tag{2.21}$$

and which is used in comparing histograms.[25] This distance finds application in correspondence analysis, as well as in the analysis of textures of digital images. Another measure, which is used in this context, is the Bhattacharyya distance, which measures the separability of classes; this distance is defined as follows:

$$d_B(\mathbf{x}, \mathbf{y}) = \Big(1 - BC(\mathbf{x}, \mathbf{y})\Big)^{1/2} \tag{2.22}$$

where $BC(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n}\sqrt{x_i y_i}$ is the so-called Bhattacharyya coefficient. Sometimes, the following definition is used: $d_B(\mathbf{x}, \mathbf{y}) = -\ln BC(\mathbf{x}, \mathbf{y})$.

---

[23] see, e.g., J. Mazerski. *Podstawy chemometrii*, Gdańsk 2004. Electronic edition available at http://www.pg.gda.pl/chem/Katedry/Leki_Biochemia/dydaktyka/chemometria/podstawy_chemometrii.zip.

[24] See http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Pearson_Correlation_and_Pearson_Squared_Distance_Metric.htm.

[25] See V. Asha, N.U. Bhajantri, and P. Nagabhushan: GLCM-based chi-square histogram distance for automatic detection of defects on patterned textures. *Int. J. of Computational Vision and Robotics*, **2**(4), 302–313, 2011.

A survey on the measures of similarity/dissimilarity, used in grouping of the time series is provided, for instance, in [328].[26]

## 2.3 Hierarchical Methods of Cluster Analysis

Hierarchical methods are among the traditional techniques of cluster analysis. They consist in successive aggregation or division of the observations and their subsets. Resulting from this kind of procedure there is a tree-like structure, which is referred to as dendrogram.

The agglomerative techniques start from the set of observations, each of which is treated as a separate cluster. Clusters are aggregated in accordance with the decreasing degree of similarity (or the increasing degree of dissimilarity) until one, single cluster is established. The manner of proceeding is represented by the following pseudo-code 2.1:

**Algorithm 2.1** Algorithm of agglomerative cluster analysis

**Require:** Data $X = (\mathbf{x}_1, \ldots, \mathbf{x}_m)^{\mathrm{T}}$.
**Ensure:** Dendrogram $\mathcal{C} = \{C_1, \ldots, C_{2m-1}\}$.
1: *Initialization*. Establish $m$ single-element clusters and calculate distance for each pair of such clusters. Memorize the distances calculated in the symmetric square matrix $D = [d_{ij}]$.
2: Find a pair $C_i, C_j$ of the clusters that are the closest to each other.
3: Form a new cluster $C_k = C_i \cup C_j$. In the generated dendrogram this corresponds to introducing a new node and connecting it with the nodes, corresponding to the clusters $C_i, C_j$.
4: Update the distance matrix, i.e. calculate the distance between the cluster $C_k$ and the remaining clusters, except for $C_i$ and $C_j$.
5: Remove from the matrix $D$ rows and columns, corresponding to the aggregated clusters $C_i, C_j$ and add a row and a column, for the new cluster $C_k$.
6: Repeat steps (2)–(5) until only one, single cluster is created.

In step 3 of the above algorithm we join together two closest clusters. By defining more precisely the notion, related to the new distances from the cluster thus created to the remaining ones, one obtains seven different variants of the agglomerative algorithms (abbreviations in brackets correspond to the names, introduced in [434]):

(a) Single linkage method or nearest neighbour method (*single linkage*): Distance between two clusters is equal to the distance between two closest elements belonging to different clusters. The resulting clusters form, in this case, long

[26] An extensive overview of distance measures is provided in the *Encyclopedia of Distances* by Michel Marie Deza and Elena Deza, Springer 2009, http://www.uco.es/users/ma1fegan/Comunes/asignaturas/vision/Encyclopedia-of-distances-2009.pdf.

"chains". In order to find the optimum solution to the task, involving the method specified, the algorithms are used, referring to the minimum spanning tree.[27]

(b) Complete linkage method or the farthest neighbour method (*complete linkage*): Distance between two clusters is equal to the distance between two farthest objects, belonging to different clusters. This method is most appropriate, when the real objects form well separated and compact clusters.

(c) Average linkage method (*unweighted pair-group average*, UPGA): Distance between two clusters is equal the average distance between all pairs of objects belonging to both clusters considered.

(d) Weighted pair-group average method (*weighted pair-group average*, WPGA): This method is similar to the preceding one, but calculations are carried out with the weights, equal the numbers of objects in the two clusters considered. This method is advised in cases, when we deal with clusters having distinctly different numbers of objects.

(e) Centroid method (*unweighted pair-group centroid*, UPGC): Distance between two clusters is equal to distance between their centroids (gravity centers).

(f) Method of weighted centroids or of the median (*weighted pair-group centroid*, WPGC): Distance between two clusters is calculated as in the previous method, but with introduction of weights, which are equal the numbers of objects in the clusters considered.

(g) Ward method of minimum variance. In this method the sum of squares of distances between objects and the center of the cluster, to which the objects belong, is minimized. This method, even though considered to be very effective, tends to form clusters having similar (low) cardinalities.

The opposition to the agglomerative techniques is constituted by the divisive techniques. Here, analysis starts from a single all-encompassing cluster, which is subject to successive divisions, according to the increasing degree of similarity. These techniques, even though apparently symmetric to the agglomerative ones, are used in practice much less frequently.

Since for a given set of observations one can obtain multiple different hierarchies, the question arises: "To what extent a dendrogram reflects the distances between the particular pairs from the set $X$?" One of the popular means for assessing the quality of grouping was introduced by Sokal and Rohlf,[28] namely the *cophenetic correlation coefficient*. Given a dendrogram, the matrix $D_T$ is formed, presenting the levels of aggregation, at which the pairs of objects appeared for the first time in the same cluster. Let further $E$ be a vector (variable) formed of the elements located above the main diagonal of the distance matrix $D$ and let $T$ be the vector (variable)

[27] M. Delattre and P. Hansen. "Bicriterion cluster analysis", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol-2, No. 4, pp. 277–291, 1980.

[28] R.R. Sokal, F.J. Rohlf. 1962. The comparison of dendrograms by objective methods. *Taxon*, 11(2), 1962, 33–40.

formed out of the elements situated above the main diagonal[29] of the $D_T$ matrix. The cophenetic correlation coefficient is the Pearson correlation coefficient between these two variables. The computational details are presented in the example 2.3.1. Another coefficient, which allows for assessing the degree of matching between the dendrogram and the matrix of distances (similarities) is the Goodman-Kruskal coefficient[30] (*Goodman-Kruskal gamma coefficient*, *gamma index*). It was introduced with the intention of assessing the concordance of orderings of features expressed on the ordinal scale.

The hierarchical methods have some important shortcomings. We mention below some of the most important ones:

(i) They lose their clarity with the increase of the number of analysed objects.
(ii) There is no way to shift objects from one cluster to another, even if they had been wrongly classified at the initial stages of the procedure.
(iii) The results reflect the degree, to which the data match the structure implied by the algorithm selected ("chain" or a compact "cloud").

*Example 2.3.1* Consider the data from the Fig. 2.5a. The matrix of distances between the objects is of the form

$$D = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 4.4721 & 4.2426 & 2.2361 & 2.8284 & 3.1623 \\ 2 & & 0 & 1.4142 & 3.0000 & 2.0000 & 3.1623 \\ 3 & & & 0 & 2.2361 & 1.4142 & 2.0000 \\ 4 & & & & 0 & 1.0000 & 1.0000 \\ 5 & & & & & 0 & 1.4142 \\ 6 & & & & & & 0 \end{array}$$

By applying the complete link (farthest neighbour) method, we construct the corresponding dendrogram. In the first step we aggregate the objects with numbers 4 and 6, between which distance is equal 1 unit. In the next step we add object number 5, situated at the distance $d(\{\mathfrak{x}_4, \mathfrak{x}_6\}, \mathfrak{x}_5) = \max\left(d(\mathfrak{x}_4, \mathfrak{x}_5), d(\mathfrak{x}_6, \mathfrak{x}_5)\right) = 1.4142$. In the third step we glue together objects having numbers 2 and 3, between which distance is equal 1.4142. Then, in the fourth step, we aggregate the clusters formed until now into one cluster $\{\{\{\mathfrak{x}_4, \mathfrak{x}_6\}, \mathfrak{x}_5\}, \{\mathfrak{x}_2, \mathfrak{x}_3\}\}$, the distance between these clusters being equal 3.1623. Finally, in the last step, we add the element $\mathfrak{x}_1$, which is situated at the distance of 4.4721 from the cluster already established. The dendrogram obtained therefrom is shown in Fig. 2.5b. The matrix of the dendritic distances, determined on the basis of the calculations performed, is as follows:

[29] Both distance matrices are symmetric, it suffices, therefore, to consider their elements situated above (or below) the main diagonal.

[30] L.A. Goodman, W.H. Kruskal. Measures of association for cross classifications. *J. of the American Statistical Association*, 49(268), 1954, 732–764.

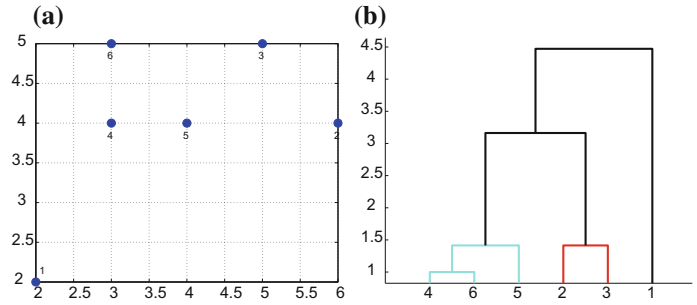



**Fig. 2.5** The exemplary data set **a** and the corresponding dendrogram, **b** obtained from the complete link method

$$D_T = \begin{bmatrix} 0 & 4.4721 & 4.4721 & 4.4721 & 4.4721 & 4.4721 \\ & 0 & 1.4142 & 3.1623 & 3.1623 & 3.1623 \\ & & 0 & 3.1623 & 3.1623 & 3.1623 \\ & & & 0 & 1.4142 & 1.0000 \\ & & & & 0 & 1.4142 \end{bmatrix}$$

Hence, vectors $E$ and $T$ are equal

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $E$ | 4.47 | 4.24 | 2.23 | 2.83 | 3.16 | 1.41 | 3.00 | 2.00 | 3.16 | 2.24 | 1.41 | 2.00 | 1.00 | 1.00 | 1.41 |
| $T$ | 4.47 | 4.47 | 4.47 | 4.47 | 4.47 | 1.41 | 3.16 | 3.16 | 3.16 | 3.16 | 3.16 | 3.16 | 1.41 | 1.00 | 1.41 |

The cophenetic correlation coefficient, expressing the degree of match between the dendrogram from Fig. 2.5b and the matrix of distances $D$ is equal the coefficient of Pearson correlation $c(E, T) = 0.7977$. □

Interestingly, there exist solid theoretical foundations for in-the-limit behaviour of hierarchical clustering. Imagine, following [105], a (continuous) subset $\mathcal{X}$ of an $n$-dimensional real-valued space $\mathbb{R}^n$ with a probabilistic density function $f : \mathcal{X} \to \mathbb{R}$. Let an Euclidean distance function be defined in this space and $B(\mathbf{x}, r)$ be a ball of radius $r$ around $\mathbf{x}$ in this space. Furthermore let $S$ be a subset of $\mathcal{X}$. We say that $\mathbf{x}$, $\mathbf{y}$ are connected in $S$ if there exists a continuous function (called "path") $P : [0, 1] \to S$ such that $P(0) = \mathbf{x}$ and $P(1) = \mathbf{y}$. If there exists a path between two points in $S$, then we say that they are connected in $S$. The relation "be connected in $S$" is an equivalence relation in $S$, hence it splits $S$ into disjoint sets of points where the points are connected within the sets but not between them.

**Definition 2.3.1** Let us define, for a real value $\lambda$, the set $S = \{\mathbf{x} \in \mathcal{X};\ f(\mathbf{x}) >\geq \lambda\}$. Then we will say that the set $\mathbb{C}_{f,\lambda}$ of disjoint subsets of $S$ is a *clustering* of $\mathcal{X}$, and each of these subsets will be called a *cluster*.

A cluster is intuitively a "high density connected subregion of $\mathcal{X}$". the smaller $\lambda$, the bigger will be the clusters.

**Definition 2.3.2** The function $\mathbb{C}_f$ assigning each real value $\lambda$ a clustering $\mathbb{C}_{f,\lambda}$ would be called a *cluster tree* of $\mathcal{X}$. $\mathbb{C}_f(\lambda) = \mathbb{C}_{f,\lambda}$

This function is called "a cluster tree" because for any two $\lambda' \leq \lambda$ it has the following properties:

- If $C \in \mathbb{C}_f(\lambda)$, then there exists such a cluster $C' \in \mathbb{C}_f(\lambda')$ that $C \subseteq C'$
- If $C \in \mathbb{C}_f(\lambda)$ and $C' \in \mathbb{C}_f(\lambda')$ then either $C \subseteq C'$ or $C \cap C' = \emptyset$.

Let us mention also their notion of $\sigma, \epsilon$-separation under the density function $f$.

**Definition 2.3.3** Two sets $A, A' \subseteq \mathcal{X}$ are said to be $\sigma, \epsilon$-*separated*, if there exists a set $S \subseteq \mathcal{X}$

- Any path in $\mathcal{X}$ from $A$ to $A'$ intersects with $S$
- $\sup_{x \in S_\sigma} f(x) < (1 - \epsilon) \inf_{x \in A_\sigma \cup A'_\sigma} f(x)$

Hereby for any set $Y$ the notation $Y_\sigma$ means all points not more distant from $Y$ than $\sigma$.

Under this definition, $A_\sigma \cup A'_\sigma$ must lie within $\mathcal{X}$ (so that the density is non-zero), but $S_\sigma$ does not need to.

From this point of view, an actual data set $\mathfrak{X}$ can be considered as a sample from the set $\mathcal{X}$ and the result of hierarchical clustering of $\mathfrak{X}$ may be deemed of as an approximation of the (intrinsic) cluster tree of $\mathcal{X}$. We may ask how well and under what circumstances this approximation is good. Hartigan [236] introduced the following notion of consistency of a clustering $\mathbb{C}_\mathfrak{X}$, where $\mathfrak{X}$ is a sample containing $m$ elements, with cluster tree of $\mathcal{X}$. For any two sets $A, A' \subset \mathcal{X}$, let $A_*, A'_*$ resp. denote the smallest cluster of $\mathbb{C}_\mathfrak{X}$ containing $A \cap \mathcal{X}$, $A' \cap \mathcal{X}$ resp. We say that $\mathbb{C}_\mathfrak{X}$ is consistent, if for a $\lambda$ and $A, A'$ being disjoint elements of $\mathbb{C}_f(\lambda)$ probability that $A_*, A'_*$ are disjoint equals 1 in the limit when the sample size $m$ grows to infinity. Hartigan [236] showed that the single link algorithm is consistent only in a one dimensional space.

Chaudhuri and Dasgupta [105] proposed therefore the "robust" single link Algorithm 2.2. This algorithm reduces to single link algorithm in case of $\alpha = 1$ and $k = 1$, or to Wishart algorithm for $\alpha = 1$ and $k$ larger. In order to broaden the class of consistent hierarchical clustering algorithms, Chaudhuri and Dasgupta suggest to use (end of Sect. 3.2.) $\alpha \in [\sqrt{2}, 2]$ and $k_m$ equal to closest bigger integer to $n(\ln m)^2$ where $n$ is the dimensionality and $m$ is the sample size.

Please note in passing that conceptually the idea of a "cluster tree" refers to some probability density distribution. Nonetheless a distance-based clustering algorithm could be analysed. It can be stated generally, that the clustering concepts based on distance, similarity graphs and density can be considered as strongly related because it is always possible for a data set, for which we know only distances, or similarities or linkage graphs, to embed it in an Euclidean space reflecting faithfully the distances/similarities so that the respective clusters can be analysed in a density-based manner.

**Algorithm 2.2** "Robust" Single Link

**Require:** Data $X = (\mathbf{x}_1, \ldots, \mathbf{x}_m)^{\mathrm{T}}$.
**Ensure:** Dendrogram $\mathcal{C} = \{C_1, \ldots, C_{2m-1}\}$.
1: *Initialization*. Natural number $k \geq 1$ and a real number $\alpha \in [1, 2]$ are user-defined parameters. Establish $m$ single-element clusters and calculate distance for each pair of such clusters. Memorize the distances calculated in the symmetric square matrix $D = [d_{ij}]$. For each data element $\mathbf{x}$ calculate $r_k(\mathbf{x})$ as the diameter of the smallest ball around $\mathbf{x}$ containing $k$ elements (including $\mathbf{x}$).
2: **for** $r$ taking on distinct values of $r_k(\mathbf{x})$ computed above, from the lowest to the largest **do**
3: Find a pair $C_i$, $C_j$ of the clusters that contain the closest elements to each other among those pairs for which for each $\mathbf{x} \in C_i \cup C_j$ $r_k(\mathbf{x}) \leq r$ and there exist $\mathbf{x} \in C_i$, $\mathbf{y} \in C_j$ such that $||\mathbf{x}, \mathbf{y}|| \leq \alpha r$.
4: Form a new cluster $C_k = C_i \cup C_j$. In the generated dendrogram this corresponds to introducing a new node and connecting it with the nodes, corresponding to the clusters $C_i$, $C_j$.
5: Update the distance matrix, i.e. calculate the distance between the cluster $C_k$ and the remaining clusters, except for $C_i$ and $C_j$.
6: Remove from the matrix $D$ rows and columns, corresponding to the aggregated clusters $C_i$, $C_j$ and add a row and a column, for the new cluster $C_k$.
7: Repeat steps (2)–(5) until no more link can be added.
8: Connected components at this point are regarded as clusters belonging to one level of the cluster tree
9: **end for**

## 2.4 Partitional Clustering

Let $U = [u_{ij}]_{m \times k}$ denote the matrix with elements indicating the fact of assignment of $i$-th object to the $j$-th class. When $u_{ij} \in \{0, 1\}$, then we speak of a "crisp" partition of the set $\mathfrak{X}$, while for $u_{ij} \in [0, 1]$—we deal with the "fuzzy" partition. The latter case is considered in Sect. 3.3. Here, we concentrate on the "crisp" partitions.

If matrix $U$ is supposed to represent the partition of the set of objects (see conditions, mentioned in Sect. 2.1), then this matrix has to satisfy the following conditions: (a) each object has to belong to exactly one cluster, that is $\sum_{j=1}^{k} u_{ij} = 1$ and (b) each cluster must contain at least one object, but cannot contain all the objects, i.e. $1 \leq \sum_{i=1}^{m} u_{ij} < m$. Denote by $\mathcal{U}_{m \times k}$ the set of all the matrices, representing the possible partitions of the set of $m$ objects into $k$ disjoint classes. It turns out, see, e.g., [19, 303], that for the given values of $m$ and $k$ the cardinality of the set $\mathcal{U}_{m \times k}$ is defined by the formula

$$\vartheta(m, k) = \frac{1}{k!} \sum_{j=1}^{k} (-1)^{k-j} \binom{k}{j} j^m \tag{2.23}$$

For instance, $\vartheta(3, 2) = 3$, but already $\vartheta(100, 5)$, is the number of the order of $10^{68}$. Generally, the number of ways, in which $m$ observations can be divided among $k$ clusters is approximately equal $k^m/k!$, meaning that it is of the order of $O(k^m)$. In

particular, when $k = 2$, then $\vartheta(m, 2) = 2^{m-1} - 1$. The problem of selection of the proper partition is, therefore, an $\mathcal{NP}$-complete task of combinatorial optimisation.

An effective navigation over the sea of the admissible partitions is secured by the criteria of grouping and the methods of optimisation, coupled with them.

### 2.4.1 Criteria of Grouping Based on Dissimilarity

The fundamental criteria, applied in the partitional clustering are homogeneity and separation. Homogeneity of a cluster means that two arbitrary objects, which belong to it, are sufficiently similar, while separation of clusters means that two arbitrary objects, belonging to different clusters are sufficiently different. In other words, the partition, induced by the clustering algorithm should contain homogeneous and well separated groups of objects.

Let $D = [d_{ij}]_{m\times m}$ denote the matrix with elements $d_{ij}$ corresponding to the dissimilarities of objects $i$ and $j$. We assume that $D$ is a symmetric and non-negative matrix, having zeroes on the diagonal. The homogeneity of a cluster $C_l$ composed of $n_l$ objects can be measured with the use of one of four indicators,[31] see, e.g., [233]:

$$\begin{aligned} h_1(C_l) &= \sum_{\mathbf{x}_i,\mathbf{x}_j\in C_l} d_{ij} \\ h_2(C_l) &= \max_{\mathbf{x}_i,\mathbf{x}_j\in C_l} d_{ij} \\ h_3(C_l) &= \min_{\mathbf{x}_i\in C_l} \max_{\mathbf{x}_j\in C_l} d_{ij} \\ h_4(C_l) &= \min_{\mathbf{x}_i\in C_l} \sum_{\mathbf{x}_j\in C_l, j\neq i} d_{ij} \end{aligned} \tag{2.24}$$

The first of these indicators is the sum of dissimilarities between the pairs of objects belonging to the cluster $C_l$. If the elements of the set $C_l$ are mutually sufficiently similar, the set can be treated as a clique,[32] and so $h_1$ represents the weight of a clique. The second indicator is the maximum value of dissimilarity in the group $C_l$; it corresponds to the diameter of the set $C_l$. The third indicator is being referred to as the radius of the set $C_l$, while the fourth one is defined as a minimum sum of dissimilarities between the objects from the set $C_l$ and its representative. This latter indicator is called *star index* or medoid.

Separation is quantified with the use of the following indicators

[31] For computational reasons, instead of distance, its squared value is often used, allowing for omitting the square root operation.

[32] In graph theory, a clique is such a subgraph, in which any two vertices are connected by an edge. Admitting that we connect with an edge the vertices that are little dissimilar, we treat a clique as a set of mutually similar vertices.

$$\begin{aligned} s_1(C_l) &= \sum_{\mathbf{x}_i \in C_l} \sum_{\mathbf{x}_j \notin C_l} d_{ij} \\ s_2(C_l) &= \min_{\mathbf{x}_i \in C_l, \mathbf{x}_j \notin C_l} d_{ij} \end{aligned} \tag{2.25}$$

The first of them, $s_1(C_l)$, called the cutting cost, is the sum of distances between the objects from the set $C_l$ and the objects from outside of this set. The second one, $s_2(C_l)$ is equal the minimum dissimilarity between the elements of the set $C_l$ and the remaining elements of the set $\mathfrak{X}$.

Based on the measures as defined above, we can quantify the quality $J(m, k)$ of the partition of the set of $m$ elements into $k$ disjoint groups with the use of the indicators given below:

$$\begin{aligned} J_1(m, k) &= \frac{1}{k} \sum_{i=1}^{k} \mathfrak{w}_i \\ J_2(m, k) &= \max_{i=1,\ldots,k} \mathfrak{w}_i \\ J_3(m, k) &= \min_{i=1,\ldots,k} \mathfrak{w}_i \end{aligned} \tag{2.26}$$

Symbol $\mathfrak{w}_i$ denotes one of the previously defined indicators of homogeneity/separability, assigned to the $i$-th group. If $\mathfrak{w}$ represents homogeneity, then the indicator $J_1(n, k)$ corresponds to the average homogeneity inside groups, $J_2(n, k)$—the maximum homogeneity, and $J_3(n, k)$—the minimum homogeneity in the partition produced. A good partition ought to—in the case of homogeneity—be characterised by the possibly low values of these indicators, while in the case of separability—by their possibly high values.

### 2.4.2 The Task of Cluster Analysis in Euclidean Space

Conform to the convention adopted, each object $\mathfrak{x}_i \in \mathfrak{X}$ is described by the $n$-dimensional vector of features, so that the set $\mathfrak{X}$ is identified with the set of $m$ points in the $n$-dimensional Euclidean space. Let

$$\overline{\boldsymbol{\mu}} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i \tag{2.27}$$

be the gravity centre of the set of $m$ objects and let

$$\boldsymbol{\mu}_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} \mathbf{x}_i = \frac{1}{\sum_{i=1}^{m} u_{ij}} \sum_{i=1}^{m} u_{ij} \mathbf{x}_i = \frac{1}{n_j} \sum_{i=1}^{m} u_{ij} \mathbf{x}_i \tag{2.28}$$

denote the gravity centre of the $j$-th cluster, where $n_j = |C_j| = \sum_{i=1}^{m} u_{ij}$ is the cardinality of the $j$-th cluster.

We define two matrices (see, e.g., [173, 303]):

$$W = \sum_{i=1}^{m} \sum_{j=1}^{k} u_{ij} (\mathbf{x}_i - \boldsymbol{\mu}_j)(\mathbf{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}} \tag{2.29}$$

$$B = \sum_{j=1}^{k} \left( \sum_{i=1}^{m} u_{ij} \right) (\boldsymbol{\mu}_j - \overline{\boldsymbol{\mu}})(\boldsymbol{\mu}_j - \overline{\boldsymbol{\mu}})^{\mathrm{T}} \tag{2.30}$$

Matrix $W$ is the in-group covariance matrix, while $B$ is the inter-group covariance matrix. Both matrices together constitute a decomposition of the dispersion matrix, or variance-covariance matrix

$$T = \sum_{i=1}^{m} (\mathbf{x}_i - \overline{\boldsymbol{\mu}})(\mathbf{x}_i - \overline{\boldsymbol{\mu}})^{\mathrm{T}} \tag{2.31}$$

i.e.: $T = W + B$.

The typical objective functions, which are used as the criteria of selection of the proper partition, are [173]:

(a) Minimisation of the trace of matrix $W$. This criterion is equivalent to minimisation of the sum of squares of the Euclidean distances between the objects and the centres of clusters, to which these objects belong, that is

$$\begin{aligned} J_1(m,k) &= \sum_{j=1}^{k} \sum_{i=1}^{m} u_{ij} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2 \\ &= \sum_{j=1}^{k} \frac{1}{n_j} \sum_{\mathbf{x}_i, \mathbf{x}_l \in C_j} \|\mathbf{x}_i - \mathbf{x}_l\|^2 \end{aligned} \tag{2.32}$$

In other words, minimisation of the indicator $J_1$ is equivalent to minimisation of the criterion of homogeneity, $h_1(C_j)/n_j$. Such approach favours spherical clusters.

(b) Minimisation of the determinant of matrix $W$. This criterion is useful in the situations, when the natural clusters are not spherical.

(c) Maximisation of the trace of matrix $BW^{-1}$. This is a generalisation of the Mahalanobis distance (2.8) for the case of more than two objects. The shortcoming of this criterion is its sensitivity to scale. Grouping obtained from the raw data may drastically differ from the one obtained after the data are rescaled (e.g. through standardisation or normalisation).

### 2.4.2.1 Minimising the Trace of In-Group Covariance

Despite the limitations, mentioned before, the criterion (2.32) is among the most willingly and most frequently used in practice. We shall soon see that this is not so much the effect of its simplicity, as—surprisingly—its relatively high degree of universality.

Minimisation of the quality index (2.32) leads to the mathematical programming problem of the form

$$\begin{aligned} &\min_{u_{ij}\in\{0,1\}} \sum_{i=1}^{m}\sum_{j=1}^{k} \left\| \mathbf{x}_i - \frac{\sum_{l=1}^{m} u_{lj}\mathbf{x}_l}{\sum_{l=1}^{m} u_{lj}} \right\|^2 \\ \text{subject to } &\sum_{i=1}^{m} u_{ij} > 1,\ j = 1,\ldots,k \\ &\sum_{j=1}^{k} u_{ij} = 1,\ i = 1,\ldots,m \end{aligned} \tag{2.33}$$

It is a 0/1 (binary) programming problem with a nonlinear objective function, [16, 233]. The integer constraints, along with the nonlinear and non-convex objective function make the problem (2.33) $\mathcal{NP}$-hard. For this reason the determination of the minimum of the function $J_1$ is usually performed with the use of the heuristic methods. Yet, attempts have been and are being made, aiming at a satisfactory solution to the problem (2.33). A survey of these attempts can be found, for instance, in [16, 30, 390]. The goal of such studies is not only to find the optimum solution, but also to gain a deeper insight into the very nature of the task of grouping the objects. We provide below several equivalent forms of the problem (2.33), enabling its generalisation in various interesting ways.

Let $F \in \mathbb{R}^{m\times m}$ be a matrix having the elements

$$f_{ij} = \begin{cases} \frac{1}{n_j} & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in C_j \\ 0 & \text{otherwise} \end{cases} \tag{2.34}$$

where, as before, $n_j$ denotes the number of objects in the group $C_j$.

If we number the objects from the set $\mathfrak{X}$ in such a manner that the first $n_1$ objects belong to group $C_1$, the successive $n_2$ of objects belong to group $C_2$, etc., then $F$ is a block-diagonal matrix, $F = \mathrm{diag}(F_1, \ldots, F_k)$. Each block $F_j$ is an $n_j \times n_j$ matrix having elements $1/n_j$, i.e. $F_j = (1/n_j)\mathbf{e}\mathbf{e}^{\mathrm{T}}$, $j = 1, \ldots, k$.

**Lemma 2.4.1** *Matrix $F$ of dimensions $m \times m$, having elements defined as in Eq. (2.34), displays the following properties:*

(a) it is a non-negative symmetric matrix, that is, $f_{ij} = f_{ji} \geq 0$, for $i, j = 1, \ldots, m$,
(b) it is a doubly stochastic matrix, that is, $F\mathbf{e} = F^{\mathrm{T}}\mathbf{e} = \mathbf{e}$,
(c) $FF = F$ (idempotency)

(d) $\mathrm{tr}\,(F) = k$,
(e) spectrum of the matrix $F$, $\sigma(F) = \{0, 1\}$, and there exist exactly $k$ eigenvalues equal 1.

*Proof* Properties (a)–(d) are obvious. We shall be demonstrating only the property (e). Since every block has the form $F_j = \mathbf{e}\mathbf{e}^{\mathrm{T}}/n_j$, then exactly one eigenvalue of this submatrix is equal 1, while the remaining $n_j - 1$ eigenvalues are equal zero. The spectrum of the matrix $F$, $\sigma(F) = \bigcup_{j=1}^{k} \sigma(F_j)$, hence $F$ has exactly $k$ eigenvalues equal 1. □

If $X = (\mathbf{x}_1 \ldots \mathbf{x}_m)^{\mathrm{T}}$ is the matrix of observations, then

$$M = FX$$

is the matrix, whose $i$-th row represents the gravity centre of the group, to which the $i$-th object belongs.

Given the above notation, the quality index (2.32) can be written down in the equivalent matrix form

$$\begin{aligned} J_1(C_1, \ldots, C_k) &= \sum_{j=1}^{k} \sum_{x_i \in C_j} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2 \\ &= \mathrm{tr}\left((X - M)^{\mathrm{T}}(X - M)\right) \\ &= \mathrm{tr}\,(X^{\mathrm{T}}X + M^{\mathrm{T}}M - 2X^{\mathrm{T}}M) \end{aligned}$$

Taking advantage of the additivity and commutativity of the matrix trace, see properties (b) and (c) in p. 321, as well as symmetry and idempotence of matrix $F$, we transform the above expression to the form

$$\begin{aligned} J_1(C_1, \ldots, C_k) &= \mathrm{tr}\,(X^{\mathrm{T}}X + X^{\mathrm{T}}F^{\mathrm{T}}FX - 2X^{\mathrm{T}}FX) \\ &= \mathrm{tr}\,(X^{\mathrm{T}}X + X^{\mathrm{T}}FX - 2X^{\mathrm{T}}FX) \\ &= \mathrm{tr}\,(X^{\mathrm{T}}X - X^{\mathrm{T}}FX) \\ &= \mathrm{tr}\,(X^{\mathrm{T}}X - X^{\mathrm{T}}XF) \end{aligned}$$

Let

$$K = X^{\mathrm{T}}X$$

It is a symmetric and non-negative matrix. The expression $\mathrm{tr}\,(X^{\mathrm{T}}XF) = \mathrm{tr}\,(KF) = \sum_{ij} k_{ij} f_{ij}$ is a linear combination of the elements of matrix $K$, hence, in case of the nonlinear objective function $J_1$ we obtain a linear function! Finally, the task of grouping of objects in the Euclidean space reduces to either *minimisation* of the indicator

$$J_1(C_1, \ldots, C_k) = \mathrm{tr}\left(K(\mathbb{I} - F)\right) \tag{2.35}$$

or, equivalently, if we ignore the constant component $K$, to *maximisation* of the indicator

$$J_1'(C_1, \ldots, C_k) = \operatorname{tr}(KF) \tag{2.36}$$

The first of these forms is used, in particular, by Peng and Wei in [390], while the second, by, for instance, Zass and Shashua in [527]. We concentrate on the task of maximisation[33]

$$\begin{array}{c} \max_{F \in \mathbb{R}^{m \times m}} \operatorname{tr}(KF) \\ \text{subject to } F \geq 0,\ F^{\mathrm{T}} = F,\ F\mathbf{e} = \mathbf{e} \\ F^2 = F,\ \operatorname{tr}(F) = k \end{array} \tag{2.37}$$

The above formulation allows for the generalisation of the task of grouping in a variety of manners:

(a) Minimisation of the indicator (2.32) assumed that the observations are points in $n$-dimensional Euclidean space. In such a case the prototypes are sought, being the gravity centres of the possibly compact groups. The quality index of the form of $\operatorname{tr}(KF)$ makes it possible to replace the distances between the points by a more general kernel function.[34] This shifts our interest in the direction of the so-called relational grouping, where, instead of the Euclidean distance between the pairs of points, a more general measure of similarity is being used, $k_{ij} = s(\mathbf{x}_i, \mathbf{x}_j)$, e.g. the Gaussian kernel function $k_{ij} = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$, where $\gamma > 0$ is a parameter. Thereby we give up the assumption that the set $\mathfrak{X}$ has to have the representation $X \in \mathbb{R}^{m \times n}$.

(b) The objective function, appearing in the problem (2.37) can be replaced by the Bregman divergence (see Definition 2.2.1 on p. 22) $D_\phi$, this leading to the problem, see [492]

$$\begin{array}{c} \max_{F \in \mathbb{R}^{m \times m}} D_\phi(K, F) \\ \text{subject to } F \geq 0,\ F^{\mathrm{T}} = F,\ F\mathbf{e} = \mathbf{e} \\ F^2 = F,\ \operatorname{tr}(F) = k \end{array} \tag{2.38}$$

When $\phi = r^2$, problem (2.38) is reduced to problem (2.37). The generalised variant of the $k$-means algorithm for such a case is considered in the study [50]. It is shown there that, in particular, the prototypes of groups are determined as the (weighted) gravity centres of these groups, and, even more importantly, that such a definition of the prototype is correct only if the dissimilarity of the objects is measured through some Bregman divergence.

---

[33] Peng and Wei note in [390] that the idea of representing the objective function appearing in the problem (2.33) in the form of minimisation of the trace of an appropriate matrix was forwarded first by A.D. Gordon and J.T. Henderson in their paper, entitled "An algorithm for Euclidean sum of squares", which appeared in the 33rd volume of the journal *Biometrica* (year 1977, pp. 355–362). Gordon and Henderson, though, wrote that this idea had been suggested to them by the anonymous referee!.

[34] See Sect. 2.5.7.

Let us also note here that formulation (2.37) turns our attention towards the doubly stochastic matrices, that is—such symmetric and non-negative matrices that the sum of every row and every column is equal 1. One can find interesting remarks on applications of such matrices in, for instance [296].

In order to enhance the flexibility of the formulation (2.37), let us replace the matrix $F$ by the product $GG^{\mathrm{T}}$, where $G$ is the matrix of the dimensions $m \times k$, having the elements

$$g_{ij} = \begin{cases} \frac{1}{\sqrt{n_j}} & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in C_j \\ 0 & \text{otherwise} \end{cases}$$

Matrix $G$ fulfils the following conditions: (a) it is non-negative, $g_{ij} \geq 0$, $i = 1, \ldots, m$, $j = 1, \ldots, k$, (b) $G^{\mathrm{T}}G = \mathbb{I}$, (c) $GG^{\mathrm{T}}\mathbf{e} = \mathbf{e}$. The non-negativity of the matrix $G$ means that $F$ is a completely positive matrix, and its cp-rank[35] equals $k$.

By referring to the fact that $\mathrm{tr}\,(AB) = \mathrm{tr}\,(BA)$, provided both products do exist, we can turn the task of maximisation (2.37) into the one of the form

$$\begin{array}{ll} & \max_{G \in \mathbb{R}^{m \times k}} \mathrm{tr}\,(G^{\mathrm{T}} K G) \\ \text{subject to} & G \geq 0, \\ & G^{\mathrm{T}}G = \mathbb{I}, \\ & GG^{\mathrm{T}}\mathbf{e} = \mathbf{e} \end{array} \tag{2.39}$$

Taking into account the formulation (2.39) we can treat the problem of grouping as a search for such a matrix $F$, for which $\mathrm{tr}\,(KF)$ attains the maximum in the set of all matrices $\mathbb{R}^{m \times m}$ fulfilling two additional conditions: (a) $F$ is a doubly stochastic matrix, (b) $F$ is a completely positive matrix, and its cp-rank = $k$, that is: $F = GG^{\mathrm{T}}$, where $G$ is a non-negative matrix of the dimensions $m \times k$. Zass and Shashua propose in [527] a two-stage procedure:

(i) The given matrix $K$ is replaced by the doubly stochastic matrix $\widetilde{F}$. In order to do this, one can use the Sinkhorn-Knopp method, which consists in the intermittent normalisation of rows and columns of matrix $K$. Other, more effective methods of turning a matrix into the doubly stochastic form are commented upon by Knight in [296].

(ii) In the set of the non-negative matrices of dimensions $m \times k$ such a matrix $G$ is sought, which minimises the error $\|\widetilde{F} - GG^{\mathrm{T}}\|_F$, where $\|A\|_F = \sqrt{\mathrm{tr}\,A^{\mathrm{T}}A} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{m} a_{ij}^2}$ denotes the Frobenius norm.

The concepts here barely outlined have been developed into the advanced methods of cluster analysis. A part of them makes use of the so-called semi-definite programming; we can mention here the studies, reported in [107, 308, 390], or in [512]. They concern not only the typical problems of minimising the trace of an appropriate matrix, but also more advanced methods of grouping, which are considered in the further parts of this book.

[35] See Definition B.2.4 in p. 320.

#### 2.4.2.2 Approximating the Data Matrix

The quality index (2.32) can be transformed to

$$J_1 = \|X - UM\|_F^2 \tag{2.40}$$

where $M \in \mathbb{R}^{k\times n}$ is the matrix with group centroids being its rows, $M = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k)^{\mathrm{T}}$, while $U \in \mathbb{R}^{m\times k}$ is the matrix indicating the assignment of the $i$-th object to the $j$-th group, $U = (\mathbf{u}_1, \ldots, \mathbf{u}_m)^{\mathrm{T}}$.

Minimisation of the indicator (2.40) allows for taking a different perspective on the task of grouping: we look for a possibly good approximation of the data matrix by the product of two matrices, $U$ and $M$. If $u_{ij} \in \{0, 1\}$, then this task is being carried out with the help of the following procedure:

(a) If $\widehat{M} = (\widehat{\boldsymbol{\mu}}_1, \ldots, \widehat{\boldsymbol{\mu}}_k)^{\mathrm{T}}$ is the current approximation of the matrix $M$, then the elements $\widehat{u}_{ij}$ of the matrix $\widehat{U}$, constituting the approximation of $U$, have the form

$$\widehat{u}_{ij} = \begin{cases} 1 \text{ if } j = \underset{1\le t\le k}{\arg\min} \|\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_t\|_F^2 \\ 0 \text{ otherwise} \end{cases} \tag{2.41}$$

(b) If $\widehat{U}$ is the current approximation of the matrix $U$, then determination of the matrix $\widehat{M}$ minimising the indicator $J_1$ is a classical problem of regression. From the condition[36] $\partial J_1/\partial \widehat{M} = \widehat{U}^{\mathrm{T}}(\widehat{U}\widehat{M} - X) = 0$ we get

$$\widehat{M} = (\widehat{U}^{\mathrm{T}}\widehat{U})^{-1}\widehat{U}^{\mathrm{T}}X \tag{2.42}$$

(c) Steps (a) and (b) are repeated until the terminal condition is fulfilled, this condition consisting in the performance of a given number of repetitions, or in stabilisation of the elements of the matrix $\widehat{U}$.

The algorithm is initiated by specifying either an approximation $\widehat{U}$, or the matrix $\widehat{M}$. A better variant, ensuring faster convergence, is to start from the matrix $\widehat{M}$. The methods of its initialisation are considered in Sect. 3.1.3.

Minimisation of the indicator (2.40) constitutes, in its essence, the problem of the so-called non-negative factorisation, playing an important role in machine learning [497], bioinformatics [145], text analysis [67, 516], or in recommender systems [301]. Yet, the problem (2.40) differs somewhat from the classical formulation [319], where it is required to have both matrices non-negative. In the case of grouping, matrix $M$ does not have to be non-negative, while matrix $U$ must satisfy certain additional constraints, like, e.g., $\sum_{j=1}^{k} u_{ij} = 1$. The thus formulated task of factorisation of matrix $X$ is a subject of intensive studies, e.g. [151, 153, 264, 326, 327].

[36] We take advantage here of the fact that $\|A\|_F^2 = \mathrm{tr}\,(A'A)$.

By generalising the indicator (2.40) to the form

$$J_1 = \|X - U^{\alpha} M^{\mathrm{T}}\|_F^2 \tag{2.43}$$

where $\alpha > 1$, we obtain the formulation leading to fuzzy grouping that we consider in Sect. 3.3. An example of application of this technique in bioinformatics shall be presented in Sect. 2.5.8.

#### 2.4.2.3 Iterative Algorithm of Finding Clusters

The algorithms of determination of the partition of objects into $k$ classes are usually iterative procedures, which converge at a local optimum [230]. An instance thereof has been presented in the preceding section. Its weak point is the necessity of performing operations on matrices in step (b). Note, though, that due to a special structure of the matrix of assignments $U$:

(a) Product $\widehat{U}^{\mathrm{T}}\widehat{U} = \widetilde{U}$ is a diagonal matrix having elements

$$\widetilde{u}_{jj} = \sum_{i=1}^{m} \widehat{u}_{ij} = n_j, \quad j = 1, \ldots, k$$

where $n_j$ denotes the number of elements of the $j$-th group. Hence, $\widetilde{U}^{-1}$ is a diagonal matrix, as well, having elements $\widetilde{u}_{jj}^{-1} = 1/n_j$.

(b) Matrix $\widetilde{M} = \widehat{U}^{\mathrm{T}} X$ has the dimensions $k \times n$, and its $i$-th row is the sum of rows of the matrix $X$, corresponding to the elements of the $i$-th cluster. So, the $i$-th row of the matrix $\widetilde{U}^{-1}\widetilde{M}$ is the arithmetic mean of the coordinates of objects, assigned to the $i$-th group.

The general form of the iterative procedure of assigning objects to clusters is shown in the pseudocode 2.3. Here, the weights are additionally used, indicating the contribution of a given object to the relocation of the gravity centres. This mechanism was introduced by Zhang [530], and was applied, in particular, by Hamerly and Elkan, [230]. When all weights are equal 1, the Algorithm 2.3 corresponds to the algorithm from the preceding section and represents the classical Lloyd's heuristics [334].

The essence of the algorithm is the iterative modification of the assignment of objects to clusters. It is most common to assign an object to the cluster with the closest gravity centre, i.e.

$$u_{ij} = \begin{cases} 1 \text{ if } j = \underset{1 \le t \le k}{\arg\min} \|\mathbf{x}_j - \boldsymbol{\mu}_t\| \\ 0 \text{ otherwise} \end{cases} \tag{2.45}$$

**Algorithm 2.3** Iterative algorithm of cluster analysis (generalised Lloyd's heuristics)

**Require:** Data set $X$ and number of groups $k$.

**Ensure:** Gravity centres of classes $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k\}$ along with the assignment of objects to classes $U = [u_{ij}]_{m \times k}$.

1: *Initialisation*. Select the gravity centres of clusters and assign weights to objects $w(\mathbf{x}_i)$.

2: Updating of assignments: for each object determine its assignment to a cluster and, possibly, also its weight.

3: Updating of the gravity centres of clusters:

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{m} u_{ij} w(\mathbf{x}_i) \mathbf{x}_i}{\sum_{i=1}^{m} u_{ij} w(\mathbf{x}_i)} \tag{2.44}$$

4: Repeat steps 2 and 3 until the stopping condition is fulfilled, usually assumed to be the lack of changes in the assignment of objects to clusters.

where $\boldsymbol{\mu}_t$ denotes the gravity centre of cluster $C_t$ (this rule was applied in step (a) of the algorithm from the preceding section). It is the rule *the winner takes all*—known also from the theory of competitive supervised learning.[37] The clusters, determined in this manner, are called Voronoi clusters.[38] Formally, if $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ is a set of prototypes, then the Voronoi cluster $W_j$ is the set of points, for which $\boldsymbol{\mu}_j$ is the closest prototype, that is:

$$W_j = \{\mathbf{x} \in \mathbb{R}^n | j = \arg\min_{1 \le l \le k} \|\mathbf{x} - \boldsymbol{\mu}_l\|\} \tag{2.46}$$

These clusters are convex sets, i.e.

$$[(\mathbf{x}' \in W_j) \wedge (\mathbf{x}^* \in W_j)] \Rightarrow [\mathbf{x}' + \alpha(\mathbf{x}^* - \mathbf{x}')] \in W_j, \quad 0 \le \alpha \le 1$$

The division of the space $\mathbb{R}^n$ into Voronoi sets is called Voronoi tessellation (tiling) or Dirichlet tessellation. Their nature is illustrated in Fig. 2.6. In the two-dimensional case the lines, separating the regions, belonging to different clusters, are the lines of symmetry of the segments, linking the neighbouring gravity centres. Although effective algorithms for tiling are known for the two-dimensional case [397], the very notion remains useful also in the $n$-dimensional case.

[37] See, e.g., J. Hertz, A. Krogh, R.G. Palmer: *Introduction to the Theory of Neural Computation. Santa Fe Institute Series*, Addison-Wesley, 1991.

[38] Georgiy Fedosiyevich Voronoi, whose name appears in the Voronoi diagrams, was a Russian mathematician of Ukrainian extraction. He lived in the years 1868–1908. Interesting information on this subject can be found in 17th chapter of popular book by Ian Stewart, entitled *Cows in the Maze. And other mathematical explorations*, published in 2010 by OUP Oxford.

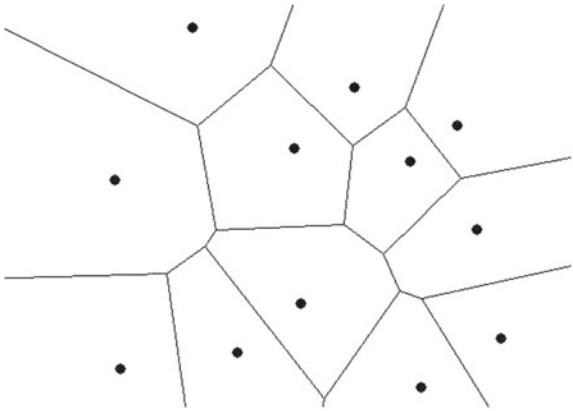


**Fig. 2.6** Voronoi tessellation, that is, the boundaries of clusters determined by the gravity centres marked by dark dots

### 2.4.3 Grouping According to Cluster Volume

Minimisation of the trace or the determinant of the matrix $W$ leads to clusters having similar numbers of elements. Besides, minimisation of $\mathrm{tr}\,(W)$ favours spherical clusters.

Report [429] presents the MVE algorithm (*Minimum Volume Ellipsoids*), in which clusters are represented by the hyper-ellipsoids having minimum volume. In distinction from the algorithms outlined in the preceding section, the MVE algorithm is independent of scale and allows for the determination of clusters having different numbers of elements. The dissimilarity measure, applied in this algorithm, is related to Mahalanobis distance. A similar issue is also discussed by Kumar and Orlin in [310].

The quality criterion adopted has the following form:

$$\begin{array}{l} \min \sum_{j=1}^{k} \mathrm{vol}\,(C_j) \\ \text{subject to } \sum_{j=1}^{k} |C_j| \geq (1-\alpha)m,\ 0 \leq \alpha < 1 \\ C_j \subset X \end{array}$$

The first limiting condition means that existence of at most $\alpha m$ outliers in the data set is allowed, while the remaining observations have to be assigned to $k$ clusters.

The hyper-ellipsoid $E_j$, containing the objects from the group $C_j$ is defined by its centre $\mathbf{c}_j$ and the symmetric and positive definite matrix $Q_j$, e.g. the covariance matrix, characterising this group of data. So,

$$E_j = \{\mathbf{x} \in \mathbb{R}^n \colon (\mathbf{x} - \mathbf{c}_j)^{\mathrm{T}} Q_j^{-1} (\mathbf{x} - \mathbf{c}_j) \leq 1\} \tag{2.47}$$

Its volume is equal $\sqrt{\det(Q_j)}$. Taking advantage of this fact, we can reduce the task of partitioning the set $X$ into $k$ groups to the problem of semi-definite programming of the form (see, e.g., [390])

$$\begin{aligned} & \min \sum_{j=1}^{k} \sqrt{\det(Q_j)} \\ \text{subject the } & (\mathbf{x}_i - \mathbf{c}_j)^{\mathrm{T}} Q_j^{-1} (\mathbf{x}_i - \mathbf{c}_j), \forall (\mathbf{x}_i \in C_j),\ j = 1, \ldots, k \\ & \sum_{j=1}^{k} |C_j| \geq (1-\alpha)m,\ 0 \leq \alpha < 1 \\ & C_j \subset X \\ & C_j \succ 0,\ j = 1, \ldots, k \end{aligned} \tag{2.48}$$

Symbol $C_j \succ 0$ means that $C_j$ is a symmetric and positive definite matrix.

Publication [310] presents two algorithms that solve the problem formulated above: (1) `kVolume`, an iterative algorithm, which divides up the set of observations into the predefined number of groups, and (2) `hVolume`, that is—a hierarchical grouping algorithm, which generates a monotonic family of clusters. In both cases a fast algorithm of calculating volumes is made use of, as presented in [450].

### 2.4.4 Generalisations of the Task of Grouping

In many instances, such as: grouping of documents, social network analysis, or bioinformatics, particular objects may belong simultaneously to several groups. In other words, instead of partitioning the data set, we look for the covering of this set. One of the ways to deal with such situations is to apply the algorithms of fuzzy clustering, for instance the FCM algorithm from Sect. 3.3. Yet, in recent years, emphasis is being placed on the algorithms allowing not only for an explicit reference to simultaneous assignment of objects to various groups, but also making it possible to identify the optimum coverings of the given set of objects. Their detailed consideration exceeds the assumed framework of this book. We shall only mention a couple of interesting solutions, encouraging the reader to an own study of this matter:

– Banerjee et al. presented in [49] `MOC`–*Model based Overlapping Clustering*, which can be seen as the first algorithm, which produces the optimum covering of the data set. The authors mentioned make use of the probabilistic relational model,[39] proposed for purposes of analysis of the microarrays. While the original solution concentrates on the normal distributions, `MOC` operates on arbitrary distributions from the exponential family. Besides, by introducing Bregman divergence, one can apply here any of the distances, discussed in Sect. 2.2.1. In the opinion of the respective authors, this new algorithm can be applied in document analysis, recommender systems, and in all situations, in which we deal with highly dimensional and sparse data.

[39] E. Segal, A. Battle, and D. Koller. Decomposing gene expression into cellular processes. In: *Proc. of the 8th Pacific Symp. on Biocomputing* (PSB), 2003, pp. 89–100.

– Cleuziou[40] proposed OKM–*Overlappingk-Means*, which is a generalisation of the $k$-means algorithm. This idea was then broadened to encompass the generalised $k$-medoids algorithm.
– In other approaches to the search for the coverings, graph theory and neural networks are being applied. In the case of the graph theory methods, first the similarity graph is constructed (analogously as in the spectral data analysis, considered in Sect. 5), and then all the cliques contained in it are sought.[41]

A survey of other algorithms is provided also in [391].

Grouping of objects in highly dimensional spaces on the basis of distances between these objects gives rise to problems discussed in Sect. 2.2.1.1. A classical solution consists in projecting the entire data set on a low dimensional space (by applying, for instance, multidimensional scaling, random projections, or principal component analysis) and using some selected algorithm to cluster the thus obtained transformed data. In practice, though, it often turns out that various subsets of data ("clusters") may be located in different subspaces.

That is why the task of grouping is defined somewhat differently. Namely, such a breakdown $C_1, \ldots, C_k$ of the data set is sought, along with the corresponding subsets of features, $F_i \subset \{1, \ldots, n\}$, that points assigned to $C_j$ be sufficiently close one to another in the $F_j$ dimensional space. It can be said that $C_j$ is composed of points which, after projection on the $F_j$ dimensional space, constitute a separate cluster. A survey of methods meant to solve the thus formulated task is provided in [304, 386].

### 2.4.5 *Relationship Between Partitional and Hierarchical Clustering*

Similarly as for hierarchical clustering, we may consider a theoretical partitioning of a probability distribution underlying the actual samples. But contrary to the concept of clustering three (see Definition 2.3.2), we need a more elaborate concept of a clustering. It is generally agreed that a clustering can be viewed as a cut of the clustering tree that is a subset of the clusters of clustering tree (that is for each cluster of the cluster tree there exists either a subset or superset of it as a cluster of our clustering) such that no two clusters intersect. This particular view has been adopted e.g. in the HDBSCAN algorithm in which the dendrogram is the starting point of flat partitioning the data set.[42]

---

[40] G. Cleuziou. A generalization of $k$-means for overlapping clustering. Université d'Orléans, LIFO, Rapport No RR-2007-15.

[41] See, e.g., W. Didimo, F. Giordano, G. Liotta. Overlapping cluster planarity. In: Proc. APVIS 2007, pp. 73–80, M. Fellows, J. Guo, C. Komusiewicz, R. Niedermeier, J. Uhlmann. Graph-based data clustering with overlaps, *COCOON*, 2009, pp. 516–526.

[42] Also one can proceed in the reverse direction: One may use a partitional algorithm to create a hierarchical clustering of data, simply by applying recursively the partitional algorithm to clusters obtained previously, like in case of bi-sectional $k$-means algorithm in Sect. 3.1.5.2.

In particular, a clustering may be the set of clusters emerging for a given value of $\lambda$. This would correspond to a cut on the same level of the cluster tree. But for various practical reasons, also other criteria are used, either jointly or separately, like balanced size (in terms of enclosing tight ball), balanced probability mass, sufficient separation for a given sample size etc. Hence each cluster may be obtained for a different density threshold value $\lambda_j$. In this case the cluster tree would be cut at different levels at different branches.

Like in case of hierarchical clustering, upon obtaining a clustering, its consistency with the clustering tree, $\sigma$, $\epsilon$-separation and other statistical properties with respect to the underlying distribution should be checked. Such checks may fail due to at least one of the following problems (1) no structure in the data exists, (2) no structure compatible with the class of structures sought by the applied algorithm exists, (3) the sample size is too small to reveal the underlying structure.

## 2.5 Other Methods of Cluster Analysis

Methods, which have been outlined in the two preceding sections belong to two essential streams developing within cluster analysis. In both cases a cluster is understood as a set of objects that are mutually much more similar than any two objects selected from two different clusters.

### *2.5.1 Relational Methods*

It has been assumed till now that each object is represented by a vector of feature values. An alternative description is constituted by the relation of similarity or dissimilarity for the pairs of objects. We encounter such situation in social sciences or in management [138, 242]. By operating on the relation of similarity / dissimilarity we can conceal the values of the attributes, which may be of significance in such domains as, for instance, banking. It also is simpler to deal with mixed attribute types (both quantitative and qualitative). The sole problem to be solved at this level is the choice of the function measuring similarity / dissimilarity of objects.

The notion of "relational methods" is sometimes used in a wider context. Thus, for instance, a set of relations may be given, $S_i$, defined on different subsets $\mathfrak{X}_i$ of objects [27], or relations may have a more complex character. In particular, when grouping documents, it is worthwhile to consider relations between subject groups and keywords.

Graph-based methods mentioned below can be viewed as special cases of relational methods understood in this way. Notably, the discussion of graph clustering in Chap. 5 is not restricted to classical graphs alone, but addresses at least partially also weighted graphs that can well represent the problems of relational clustering.

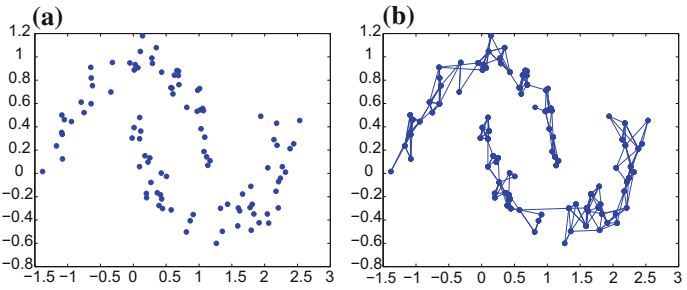



**Fig. 2.7** Identification of clusters by examining mutual similarity between objects: **a** a set of points. In illustrations like this we follow the intuitive convention: The points are meant to lie in a plane, the horizontal axis reflects the first coordinate and the vertical one the second coordinate, **b** a graph obtained by joining five nearest neighbors of each node

### *2.5.2 Graph and Spectral Methods*

The set $X$ is often identified with the set of vertices of a certain graph $\Gamma$, whose edges represent the connections between the objects; e.g. the pair $\{\mathbf{x}_i, \mathbf{x}_j\}$ is an edge in $\Gamma$, if the two objects are similar in the degree not lower than $s_\tau$. In such a context a cluster becomes equivalent to, for instance, a clique, that is, a connected subgraph of the graph $\Gamma$, i.e. such a one that every two vertices of the subgraph are connected by an edge. In another definition it is assumed that a cluster is such a subgraph $\Gamma_i$, whose vertices communicate exclusively among themselves, and do not communicate with other vertices, outside this subgraph. This means that when extracting clusters we take into account their connectivity and not only their compactness. This is illustrated on Fig. 2.7. Using pairwise distances between the points from the left panel we construct a graph shown on the right panel. Here, each node is linking with other five nearest nodes.

The progenitor of graph cut clustering is Zahn's[43] approach, consisting of two steps. First, using similarity matrix describing relationships among the objects, a maximum spanning tree is constructed, and then the edges with small weights are removed from this tree to get a set of connected components. This method is successful in detecting clearly separated clusters, but if the density of nodes is changed, its performance will deteriorate. Another disadvantage is that the cluster structure must be known in advance.

The understanding of clusters, as described above, places the problem of their extraction in the context of graph cutting. In particular, when we assign to every edge $\{v_i, v_j\}$ in the graph $\Gamma$ a similarity degree $s_{ij}$, the problem boils down to removing the edges with small weights in order to decompose $\Gamma$ into connected components. Such a procedure ensures that the sum of weights of the edges, connecting vertices from different groups is small in comparison with the sum of weights linking the vertices,

[43] C.T. Zahn, Graph-theoretic methods for detecting and describing gestalt clusters. *IEEE Trans. Comput.*, 20(1):68–86, 1971.

belonging to the same subgraph.[44] An exhaustive survey of techniques, which are applied in the partitioning of graphs, can be found e.g. in [179, 418].

A particularly important domain of application of this group of algorithms is parallel computing. Assume that solving a certain problem requires performing of $m$ tasks, each of which constitutes a separate process, program or thread, realised by one of $c$ processors. When the processors are identical, and all tasks are of similar complexity, then we can assign to every processor the same number of tasks. Usually, though, realisation of a concrete task requires the knowledge of partial results, produced by other tasks, which implies the necessity of communication between these tasks. In this manner we obtain a graph, with vertices corresponding to individual tasks, while edges—correspond to their communication needs. Communication between the processors, which form a parallel computing machine is much slower than data movement within one single processor. In order to minimise communication to a necessary level, the processes (vertices of the graph) ought to be partitioned into groups (i.e. assigned to processors) in such a way that the number of edges, linking different groups, be minimal. This is the formulation of a problem, whose solutions are considered, in particular, in [179].

Another group of problems is constituted by image segmentation. Here, an image is modelled as an undirected weighted graph, its vertices being pixels, or groups of pixels, while the weights of edges correspond to the degree of similarity (or dissimilarity) between the neighbouring pixels. This graph (image) is to be segmented conform to a certain criterion, defining "good" clusters.

An interesting offer, allowing for identification of complex cluster structures is spectral graph theory—its application in clustering is considered in detail in this book in Chap. 5. Various methods of spectral cluster analysis are reviewed in [176, 418, 482]. It is worth noting that spectral methods play nowadays a significant role in practical tasks of data analysis and mining, such as information retrieval [68], bioinformatics [251], or recommender systems [1, 311].

### 2.5.3 *Relationship Between Clustering for Embedded and Relational Data Representations*

Let us recall once again the cluster tree concept representing the dense and less dense regions of the data. If we sample objects from a space with various densities, then it is obvious that more objects will be clustered from dense regions and if we construct a similarity graph, then nodes corresponding to densier regions will be more strongly interconnected. Hence finding denser regions will correspond to cutting a graph in such a way as to remove the lowest number of edges (a kind of minimum cut). Due to these analogies, as we will see later in this book, various algorithmic ideas are adopted across the boundary of data representation.

[44] which means that objects, belonging to the same class are mutually sufficiently similar, while objects, belonging to different groups are sufficiently mutually dissimilar.

### 2.5.4 Density-Based Methods

In a different perspective, a cluster is conceived as a dense area in the space of objects, surrounded by low density areas. This kind of definition is convenient in situations, when clusters have irregular shapes, and the data set contains outliers and noisy observations—see Fig. 2.8. A typical representative of this group of algorithms is DBSCAN [171] and its later modifications [26, 170, 414].

Before presenting the essential idea of this algorithm, we shall introduce the necessary definitions.

(a) Let $d(\mathbf{x}, \mathbf{y})$ denote distance between any two points $\mathbf{x}, \mathbf{y} \in X$ and let $\epsilon > 0$ be a parameter. The $\epsilon$-neighbourhood of the object $\mathbf{x}$ is the set

$$N_\epsilon(\mathbf{x}) = \{\mathbf{x}' \in X : d(\mathbf{x}, \mathbf{x}') \leq \epsilon\}$$

(b) An object $\mathbf{x} \in X$ is called the internal point of a cluster, if its $\epsilon$-neighbourhood contains at least $minPts$ objects, i.e. when $|N_\epsilon(\mathbf{x})| \geq minPts$, where $minPts$ is a parameter.
(c) An object $\mathbf{x} \in X$ is called a border point, if $|N_\epsilon(\mathbf{x})| < minPts$, but this neighbourhood contains at least one internal point.
(d) If $\mathbf{x} \in X$ is neither an internal point, nor a border point, then it is treated as a disturbance (*outlier*).

In construction of the particular clusters use is made of the notion of density-reachability. Namely, a point $\mathbf{y} \in X$ is *directly* density-reachable from the point $\mathbf{x} \in X$ if $\mathbf{y} \in N_\epsilon(\mathbf{x})$, and, besides, $\mathbf{x}$ is an internal point, that is, it is surrounded by a sufficiently high number of other points. Note that the relationship of direct density-reachability is asymmetric. Then, $\mathbf{y}$ is called density-reachable from the point $\mathbf{x}$ if there exists a sequence $\mathbf{x}_1, \ldots, \mathbf{x}_n$ of points such that $\mathbf{x}_1 = \mathbf{x}$, $\mathbf{x}_n = \mathbf{y}$, and each

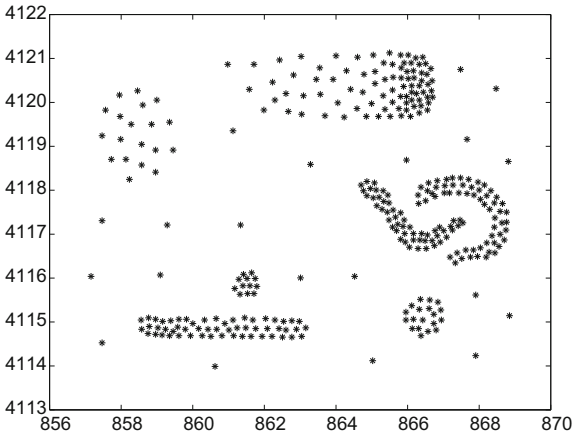



**Fig. 2.8** An example of clusters with irregular shapes and various densities. The data set contains also *outliers*

point $\mathbf{x}_{i+1}$ is directly density-reachable from $\mathbf{x}_i$, $i = 1, \ldots, n-1$. Note that the thus defined relation of reachability is asymmetric ($\mathbf{y}$ may be a border point). That is why the notion of density-connectedness is introduced: points $\mathbf{x}, \mathbf{y} \in X$ are density-connected, if there exists such a point $\mathbf{z} \in X$ that both $\mathbf{x}$ and $\mathbf{y}$ are density-reachable from $\mathbf{z}$. Clusters, generated by the DBSCAN algorithm have the following properties:

(i) All points, belonging to a cluster are mutually density-connected.
(ii) If an internal point is density-connected with another point of a cluster, then it is also an element of this cluster.

Generation of clusters, having such properties, is outlined here through the pseudocode 2.4. Note that we make use here only of the internal and border points. Each two internal points, whose mutual distance does not exceed the value of $\epsilon$ are put in the same cluster. On the other hand, the border points are classified in any cluster, provided a neighbour of this point is an internal point of the cluster.

**Algorithm 2.4** DBSCAN algorithm

**Require:** Data $X = (\mathbf{x}_1, \ldots, \mathbf{x}_m)^{\mathrm{T}}$, maximal distance between internal points $\epsilon$.
**Ensure:** A partition $\mathcal{C} = \{C_1, \ldots, C_k\}$, the number of clusters $k$.
1: *Initialisation*. Mark the points from the set $X$ as internal, border or noisy points.
2: Remove the noisy points.
3: Connect with an edge the neighbouring internal points (i.e. the internal points situated at a distance not bigger than $\epsilon$).
4: Form a cluster out of the neighbouring internal points.
5: Assign the border points to one of the clusters, upon which they neighbour.

Choice of the appropriate value of the radius $\epsilon$ influences significantly the results of the algorithm. If $\epsilon$ is too big—density of each point is identical and equal $m$ (that is—the cardinality of the set $X$). If, however, the value of the radius is too small, then $|N_\epsilon(\mathbf{x})| = 1$ for any $\mathbf{x} \in X$. The value of the parameter $\epsilon$ is often assumed as the so-called $k$-distance, $k\text{-}dist(\mathbf{x})$, namely the distance between the point $\mathbf{x}$ and its $k$-th nearest neighbour. In the case of the two-dimensional sets the value of $k = 4$ is often assumed, although there is also a suggestion of taking $k = n + 1$.

In order to select the proper value of $\epsilon$ a diagram is developed of the increasingly ordered values of $k\text{-}dist(\mathbf{x})$ for all $\mathbf{x} \in X$. It is characterised by the appearance of a value, following which an abrupt increase of distance takes place. This is the value, which is chosen as $\epsilon$. The procedure is illustrated in Fig. 2.9. Its left part shows the data set,[45] while the right part shows the diagram of $k\text{-}dist(\mathbf{x})$ for $k = 4$. It is usually assumed that $minPts = k$. One can also read out of the diagram that $\epsilon \approx 1.0$.

In a general case, instead of $k$-distance, one can use a certain function $g(\mathbf{x})$, characterising density at point $\mathbf{x}$. In the case of the DENCLUE algorithm [252] this is the sum of the components of the function

[45] It arose from adding 14 randomly generated points to the set, described in Chap. 7, namely `data3_2`.

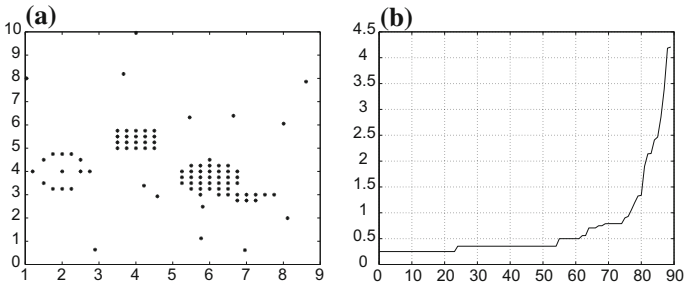



**Fig. 2.9** Diagram of $k$-distances between the points of an exemplary data set $\mathfrak{X}$: **a** An exemplary data set composed of 90 points, **b** increasingly ordered values of the average distance between each of the 90 points of the data set $\mathfrak{X}$ and their $k = 4$ nearest neighbours

$$g(\mathbf{x}) = \sum_{\mathbf{x}' \in X} f(\mathbf{x}, \mathbf{x}')$$

where $f(\mathbf{x}, \mathbf{x}')$ is, in principle, an arbitrary function, which describes the influence exerted by the object $\mathbf{x}'$ on the object $\mathbf{x}$, e.g.

$$f(\mathbf{x}, \mathbf{x}') = \exp\left( - \frac{d(\mathbf{x}, \mathbf{x}')}{2\sigma^2} \right)$$

with $\sigma > 0$ being a parameter. In this context, clusters are described as local maxima of the function $g(\mathbf{x})$.

The fundamental qualities of the DBSCAN algorithm are as follows:

1. Knowledge of the number of clusters existing in the set $X$ is not required.
2. Clusters may have arbitrary shapes. Owing to the parameter *minPts* the effect of a single connection is reduced, as manifested by the appearance of thin lines, composed of points, belonging to different clusters.
3. Data are allowed to contain noise and outliers.
4. The algorithm requires specification of just two parameters: the radius $\epsilon$, and the number *minPts*, used in classification of points.
5. The algorithm is only slightly sensitive to the order, in which the individual objects from the set $X$ are considered.

Yet, there is a significant requirement that clusters have similar densities. This shortcoming is done with in the variants of the algorithm—GDBSCAN [414] and LDBSCAN [162]. Another essential weak point is the fact that the quality of the algorithm

strongly depends upon the definition of distance $d(\mathbf{x}, \mathbf{y})$. We already know that, for instance, Euclidean distance loses on its usefulness in the case of analysis of highly dimensional data.

Moore [361] proposed the so-called anchor algorithm to analyse highly dimensional data—see the pseudocode 2.5. This algorithm belongs to the class of algorithms grouping on the basis of the minimum class diameter.

**Algorithm 2.5** Anchor grouping algorithm, [361]

**Require:** Data set $X$, number of clusters $K$, minimum number of objects in a cluster $n_{min}$.
**Ensure:** A partition $\mathcal{C} = \{C_1, \ldots, C_K\}$.
1: Select randomly a point. Initiate $k = 1$, $k' = 0$. Take as the anchor $a^1$ the point from the set $X$ which is most distant from the initial point.
2: Assign to the group $C_k$, initiated by the anchor $a^k$, these objects, which are closer to $a^k$ than to other anchors. The objects form a list, ordered decreasingly with respect to their distance from the anchor.
3: Check, whether the anchor $a^k$ has the sufficient number of objects assigned to it. If $|C_k| < n_{min}$, then $k' = k' + 1$.
4: Substitute $k = k + 1$. For $a^k$ substitute the point that is most distant from the remaining anchors.
5: If $k - k' < K$, then go to step 2.
6: **return** partition of the set $X$ into disjoint groups.

It should be noted that the algorithm may produce more than $K$ groups. It may also happen that the algorithm shall not produce sets containing more than $n_{min}$ elements.

Still another path for resolving DBSCAN shortcomings was the HDBSCAN algorithm [98]. It produces a flat set of clusters by first creating a hierarchy of clusters (like in the agglomerative methods) and then cutting this hierarchy at varying levels of hierarchy optimising some quality criteria. In this way clusters of varying density may be extracted. As a first step the minimum spanning tree is constructed where however the distances between elements are not the original ones but are modified. First a neighbourhood size parameter $k$ is defined (like in robust single link on p. 33). Then the distances are updated in such a way that it is the maximum of the distance between elements, the original distance to its $k$-th closest neighbour of the first element and of the second element. When building the tree, a dendrogram is created as in agglomerative methods. Once the dendrogram is finished, it is "pruned" to mark as "non-clusters" those clusters that were too small. For this purpose the user has to define the minimum cluster size. All clusters that are smaller in the dendrogram, are marked as "non-clusters". Afterwards the persistence of clusters is computed. In the dendrogram, if a cluster $A$ splits into a cluster $B$ and a non-cluster $C$, then the cluster $A$ is deemed to be the same as $B$. When the cluster identities are fixed in this way, the birth of a cluster is defined as the first occurrence of this cluster from the top of the dendrogram, and its death—the last occurrence. For each cluster $\lambda_{birth}$ and $\lambda_{death}$ are computed as inverses of the edge lengths added to dendrogram upon birth and death resp. Additionally, for each data point $x$ that ever belonged to the cluster its $\lambda_{x,left}$ is computed as the inverse of the distance when it left the cluster. Cluster stability is defined as the sum of $(\lambda_{x,left} - \lambda_{birth})$ over all data points ever belonging to it. We

first select all leaf nodes as "candidate flat clusters". Then working from the bottom of the tree, we check if the sum of stabilities of child clusters of a node is lower or equal the stability of a given node, then we deselect its children and select it as a candidate flat cluster. Otherwise we substitute the original stability of the node with the sum of stabilities of its child nodes. Upon reaching the top of the dendrogram the current candidate flat clusters become our actual set of clusters, being the result of HDBSCAN.

Summarizing, HDBSCAN has two parameters that need to be set by the user: the number of neighbours $k$ and the minimal cluster size. They substitute the $\epsilon$ radius of the neighbourhood that needed to contain at least $minPts$ objects in DBSCAN. By replacing $\epsilon$ with $k$, the flexibility of having clusters of varying density is achieved. Setting of $minPts$ in DBSCAN was a bit artificial from the point of view of the user, as he had to estimate cluster density in advance, prior to looking into the data. Whereas the choice of minimal cluster size in advance is a bit easier because it may be derived from some business requirements and is easy to adjust upon look in the final clustering. HDBSCAN provides also with the insight into degree of obscureness of some points in the clusters, if one looks through the dendrogram and analyses the "non-clusters".

### *2.5.5 Grid-Based Clustering Algorithms*

Grid-based approaches are recommended for large multidimensional spaces where clusters are regarded as regions that are denser than their surroundings. In this sense they can be deemed as an extension of density-based approaches.

The grid based methods handle the complexity of huge data sets by not dealing with the data points themselves but rather with the value space surrounding data points. A typical grid-based clustering algorithm proceeds as follows [210]

1. Creating the grid, that is partitioning the data space into a finite number of cells
2. Calculating the cell density for each cell
3. Sorting of the cells according to their densities.
4. Identifying cluster centres.
5. Merging information from neighbouring cells.

One example of such an algorithm, STatistical INformation Grid-based clustering method (STING), was proposed by Wang et al. [496] for clustering spatial databases. The goal of the algorithm was to enable spatial queries in two-dimensional space. The data is organised hierarchically: four lower level cells are combined into one higher level cell.

### 2.5.6 Model-Based Clustering

Model-based clustering assumes that a parametric model is underlying the cluster structure of the data. Typical approach here is to consider data as being generated by a mixture of Gaussian models. Clustering is reduced to identifying these models by detecting the parameters, e.g. mean, standard deviations, covariance matrices of the elements of the mixture etc.

### 2.5.7 Potential (Kernel) Function Methods

These methods originate from the work of Ajzerman, Braverman and Rozonoer [12], where these authors used the term of potential function. A short introduction to this approach can be found in Sect. 5.6 of the monograph [463]. Along with the development of the theory of support vector machines (SVM) [477] the term "kernel function" replaced the earlier term of the "potential function". We present below, following [176], the fundamental assumptions of cluster analysis methods using the notion of kernel function.

We assume (for simplicity and in view of practical applications), that we deal with $n$-dimensional vectors having real-valued components (and not the complex numbers, as this is assumed in the general theory). Hence, as until now, $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ denotes the non-empty set of objects, with $\mathbf{x}_i \in \mathbb{R}^n$.

**Definition 2.5.1** A function $\mathsf{K}\colon X \times X \to \mathbb{R}$ is called the positive definite kernel function (Mercer kernel or simply kernel) if: (i) $\mathsf{K}(\mathbf{x}_i, \mathbf{x}_j)$ is a symmetric function, and (ii) for any vectors $\mathbf{x}_i, \mathbf{x}_j \in X$ and any real-valued constants $c_1, \ldots, c_m$ the following inequality holds: [46]

---

[46] An introduction of Kernel functions should start with definition of a *vector space*. Let $V$ be a set, $\oplus\colon V \times V \to V$ be so-called inner operator (or vector addition) and $\odot\colon \mathbb{R} \times V \to V$ be so-called outer operator (or scalar vector multiplication). Then $(V, \oplus, \odot)$ is called vector space over the real numbers if the following properties hold: $u \oplus (v \oplus w) = (u \oplus v) \oplus w$, there exists $0_v \in V$ such that $v \oplus 0_V = 0_V \oplus v = v$, $v \oplus u = u \oplus v$, $\alpha \odot (u \oplus v) = (\alpha \odot u) \oplus (\alpha \odot v)$, $(\alpha + \beta) \odot v = (\alpha \odot v) \oplus (\beta \odot v)$, $(\alpha \cdot \beta) \odot v = \alpha \odot (\beta \odot v)$, $1 \odot v = v$.

Given the vector space, one can define the inner product space as a vector space $V$ over real numbers in which the scalar product $\langle \cdot, \cdot \rangle\colon V \times V \to \mathbb{R}$ $\langle v, v \rangle \geq 0$, $\langle v, v \rangle = 0 \Leftrightarrow v = 0$, $\langle u, v \rangle = \langle v, u \rangle$, $\langle u, \lambda \odot v \rangle = \lambda \cdot \langle u, v \rangle$, $\langle u, v + w \rangle = \langle u, v \rangle + \langle u, w \rangle$,

Now let $X$ be a the space of our input data. A mapping $K : X \times X \to \mathbb{R}$ is called a *kernel*, if there exists an inner product space $(F, \langle \cdot, \cdot \rangle)$ ($F$ being called the feature space) and a mapping $\Phi : X \to F$ such that in this inner product space $K(x, y) = \langle \Phi(x), \Phi(y) \rangle$ for all $x, y \in X$. As for inner product $\langle \Phi(x), \Phi(y) \rangle = \langle \Phi(y), \Phi(x) \rangle$ holds, obviously $K(x, y) = K(y, x)$.

Mercel has shown that if $\int_{x \in X} \int_{y \in X} K^2(x, y) dx dy < +\infty$ (compactness of $K$) and for each function $f : X \to \mathbb{R}$ $\int_{x \in X} \int_{y \in X} K(x, y) f(x) f(y) dx dy \geq 0$ (semipositive-definiteness of $K$) then there exists a sequence of non-negative real numbers (eigenvalues) $\lambda_1, \lambda_2, \ldots$ and a sequence of functions $\phi_1, \phi_2, \ldots : X \to \mathbb{R}$ such that $K(x, y) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(y)$, where the sum on the right side is absolute convergent. Moreover $\int_{x \in X} \phi_i(x) \phi_j(x) dx$ is equal 1 if $i = j$ and equal 0 otherwise.

$$\sum_{i=1}^{m}\sum_{j=1}^{m} c_i c_j \mathsf{K}(\mathbf{x}_i, \mathbf{x}_j) \geq 0$$

□

If $\mathsf{K}_1(\mathbf{x}, \mathbf{y})$, $\mathsf{K}_2(\mathbf{x}, \mathbf{y})$ are kernel functions, then their sum, product, and $a\mathsf{K}(\mathbf{x}, \mathbf{y})$, where $a > 0$, are also kernel functions. The typical kernel functions, which are used in machine learning are:

(a) Linear kernel $\mathsf{K}_l(\mathbf{x}, \mathbf{y}) = \mathbf{x}^{\mathrm{T}}\mathbf{y} + c$. In the majority of cases, the algorithms, which use linear kernel functions, are close to equivalence with their "non-kernel" counterparts (thus, e.g., the kernel-based variant of the principal component analysis with the linear kernel is equivalent to the classical PCA algorithm).

(b) Polynomial kernel $\mathsf{K}(\mathbf{x}, \mathbf{y}) = (\alpha\mathbf{x}^{\mathrm{T}}\mathbf{y} + c)^d$, where $\alpha, c$ and the degree of the polynomial, $d$, are parameters. The polynomial kernel functions are applied, first of all, in the situations, in which normalised data are used.

(c) Gaussian kernel $\mathsf{K}(\mathbf{x}, \mathbf{y}) = \exp\Big(-\|\mathbf{x} - \mathbf{y}\|^2/(2\sigma^2)\Big)$, where $\sigma > 0$ is a parameter, whose choice requires special care. If its value is overestimated, the exponent will behave almost linearly, and the very nonlinear projection will lose its properties. In the case of underestimation, function $\mathsf{K}$ loses the regularisation capacities and the borders of the decision area become sensitive to the noisy data. The Gaussian kernel functions belong among the so-called radial basis functions of the form

$$\mathsf{K}(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\sum_{j=1}^{n} |x_j^a - y_j^a|^b}{2\sigma^2}\right),\ b \leq 2$$

The strong point of the Gaussian kernel is that (for a correctly chosen value of the parameter $\sigma$) it filters out effectively the noisy data and the outliers.

Every Mercer kernel function can be represented as a scalar product

$$\mathsf{K}(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^{\mathrm{T}}\Phi(\mathbf{x}_j) \tag{2.49}$$

where $\Phi\colon X \to \mathcal{F}$ is a nonlinear mapping of the space of objects into a highly dimensional space of features $\mathcal{F}$. An important consequence of this representation is the possibility of calculating the Euclidean distance in the space $\mathcal{F}$ without knowledge of the explicit form of the function $\Phi$. In fact,

---

(Footnote 46 continued)
Obviously, the function $\phi$ may be of the form of an infinite vector $\Phi = (\sqrt{\lambda_1}\phi_1, \sqrt{\lambda_2}\phi_2, \ldots)$.

The above kernel definition constitutes a special application of this general formula to the case of a finite set $X$. The function $K$ over a finite set $X$ can be represented in such a case as a matrix, which must be therefore semipositive definite and the function $\phi$ can be expressed for each $x \in X$ as the vector of corresponding eigenvector components multiplied with square root of the respective eigenvalues.

$$\begin{aligned}\|\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)\|^2 &= \big(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)\big)^{\mathrm{T}}\big(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)\big) \\ &= \Phi(\mathbf{x}_i)^{\mathrm{T}}\Phi(\mathbf{x}_i) + \Phi(\mathbf{x}_j)^{\mathrm{T}}\Phi(\mathbf{x}_j) - 2\Phi(\mathbf{x}_i)^{\mathrm{T}}\Phi(\mathbf{x}_j) \\ &= \mathsf{K}(\mathbf{x}_i, \mathbf{x}_i) + \mathsf{K}(\mathbf{x}_j, \mathbf{x}_j) - 2\mathsf{K}(\mathbf{x}_i, \mathbf{x}_j)\end{aligned} \tag{2.50}$$

In view of the finiteness of the set $X$ it is convenient to form a matrix $K$ having elements $k_{ij} = \mathsf{K}(\mathbf{x}_i, \mathbf{x}_j)$. Since $k_{ij} = \Phi(\mathbf{x}_i)^{\mathrm{T}}\Phi(\mathbf{x}_j)$, then, from the formal point of view, $K$ is a Gram matrix, see the Definition B.2.3. Given this notation, we can write down the last equality in the form

$$\|\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)\|^2 = k_{ii} + k_{jj} - 2k_{ij} \tag{2.51}$$

The kernel functions are being used in cluster analysis in three ways, referred to through the following terms, see [176, 515]:

(a) kernelisation of the metrics,
(b) clustering in feature space $\mathcal{F}$,
(c) description via support vectors.

In the first case we look for the prototypes in the space $X$, but the distance between objects and prototypes is calculated in the space of features, with the use of Eq. (2.50). The counterpart to the criterion function (2.32) is now constituted by

$$\begin{aligned}J_1^{\Phi} &= \sum_{j=1}^{k}\sum_{i=1}^{m} u_{ij}\|\Phi(\mathbf{x}_i) - \Phi(\boldsymbol{\mu}_j)\|^2 \\ &= \sum_{j=1}^{k}\sum_{i=1}^{m} u_{ij}\Big(\mathsf{K}(\mathbf{x}_i, \mathbf{x}_i) + \mathsf{K}(\boldsymbol{\mu}_j, \boldsymbol{\mu}_j) - 2\mathsf{K}(\mathbf{x}_i, \boldsymbol{\mu}_j)\Big)\end{aligned} \tag{2.52}$$

If, in addition, $\mathsf{K}(\mathbf{x}_i, \mathbf{x}_i) = 1$, e.g. $\mathsf{K}$ is a Gaussian kernel, the above function simplifies to the form

$$J_1^{\Phi} = 2\sum_{j=1}^{k}\sum_{i=1}^{m} u_{ij}\Big(1 - \mathsf{K}(\mathbf{x}_i, \boldsymbol{\mu}_j)\Big) \tag{2.53}$$

In effect, in this case the function $d(\mathbf{x}, \mathbf{y}) = \sqrt{1 - \mathsf{K}(\mathbf{x}, \mathbf{y})}$ is a distance, and if, in addition, $\mathsf{K}$ is a Gaussian kernel, then $d(\mathbf{x}, \mathbf{y}) \to \|\mathbf{x} - \mathbf{y})\|$ when $\sigma \to \infty$.

An example of such an algorithm is considered in deeper details in Sect. 3.3.5.6. The idea of calculating distances in the space of features was also made use of in the kernelised and effective algorithm of hierarchical grouping, as well as in the kernelised version of the mountain algorithm,[47] see also [288].

[47] The mountain algorithm is a fast algorithm for determining approximate locations of centroids. See R.R. Yager and D.P. Filev. Approximate clustering via the mountain method. *IEEE Trans. on Systems, Man and Cybernetics*, 24(1994), 1279–1284.

In the second case we operate with the images $\Phi(\mathbf{x}_i)$ of the objects and we look for the prototypes $\boldsymbol{\mu}_j^{\Phi}$ in the space of features. The criterion function (2.32) takes now on the form

$$J_2^{\Phi} = \sum_{j=1}^{k} \sum_{i=1}^{m} u_{ij} \|\Phi(\mathbf{x}_i) - \boldsymbol{\mu}_j^{\Phi}\|^2 \tag{2.54}$$

where $\boldsymbol{\mu}_j^{\Phi} \in \mathcal{F}$. In Sect. 3.1.5.5 we show how this concept is applied to the classical $k$-means algorithm, and in Sect. 3.3.5.6.2.—to the $k$-fuzzy-means algorithm (FCM).

Finally, the description based on the support vectors refers to the single-class variant of the support vector machine (SVM), making it possible to find in the space of features the sphere of minimum radius, containing *almost* all data, that is—the data with exclusion of the *outliers* [63]. By denoting the centre of the sphere with the symbol $\mathbf{v}$, and its radius with the symbol $R$, we obtain the constraint of the form

$$\|\Phi(\mathbf{x}_i) - \mathbf{v}\|^2 \leq R^2 + \xi_i, \ i = 1, \ldots, m \tag{2.55}$$

where $\xi_i$ are artificial variables. More extensive treatment of this subject is presented in Sect. 3.5 of the book [176].

The basic characteristics of the kernel-based clustering algorithms are as follows:

(a) They enable formation and description of clusters having shapes different from spherical or ellipsoidal.
(b) They are well adapted to analysing the incomplete data and the data containing *outliers* as well as disturbances (noise).
(c) Their shortcoming consists in the necessity of estimating additional parameters, e.g. the value of $\sigma$ in the case of the Gaussian kernel.

Even though the characteristic (a) sounds highly encouraging, it turns out that the classical partitional algorithms from Sect. 2.4 may also be applied in such situations. We deal with this subject at greater length below.

### 2.5.8 Cluster Ensembles

Similarly as in machine learning, where the so-called families of classifiers are used in classification (see Sects. 4.5 and 4.6 in [303]), in data grouping attempts are made to enhance the effectiveness of grouping by applying the families of groupings (*cluster ensembles* [257]). This kind of approach is also referred to as aggregation of clusterings or consensus partitioning . The data set $X$ is analysed from various points of view, and the conclusions, resulting therefrom, are used in the construction of the final partition. As noted by Strehl and Ghosh [447] it is, in a way, the problem of the so-called *knowledge reuse*, with which we deal in, for instance, marketing or banking. Thus, for instance, a company disposes of various profiles, describing

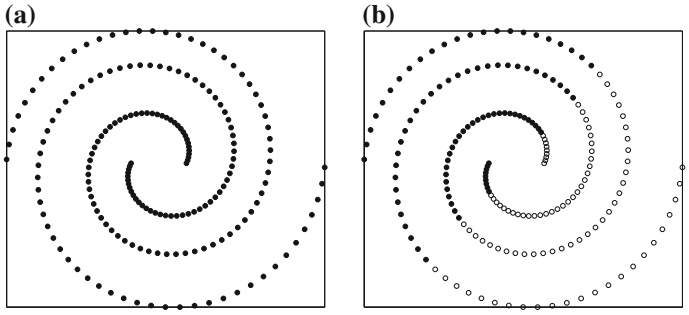



**Fig. 2.10** **a** The set `2spirals` is composed of two spirals, situated one inside the other. **b** Grouping produced by the $k$-means algorithm

the behaviour of customers in terms of demographic and geographical aspects, the history of purchases done, etc. Aggregation of such descriptions allows for formulating of composite judgements, which support the design of effective trade strategies, addressed at well selected groups of customers. It is essential that in formulation of such judgements the entire analysis does not have to be repeated from scratch, but knowledge, originating from various sources is used and creatively processed.

*Example 2.5.1* Consider a simple problem, represented by the data set,[48] which is shown in Fig. 2.10a. Data points are here located along two spirals, of which one is situated inside the other one. The classical $k$-means algorithm produces the output, which is shown in Fig. 2.10b.
In order to obtain the correct partition, Fred and Jain [187] ran $N$ times the $k$-means algorithm, assuming a different number of classes at each time. They aggregated the partial results, establishing the so-called co-association matrix, composed of the elements $w_{ij} = r_{ij}/N$, where $r_{ij}$ is the number of cases, in which the pair of objects $(i, j)$ was assigned to the same class. In order to determine the ultimate partition on the basis of this new matrix, the single link hierarchical algorithm was used, i.e. variant (a) from Sect. 2.3. □

The above way of proceeding can be formalised as follows: Let $\mathfrak{C} = \{\mathcal{C}^1, \ldots, \mathcal{C}^N\}$ be a family of partitions of the data set $X$, with $\mathcal{C}^i = \{C_1^i, \ldots, C_{k_i}^i\}$, $i = 1, \ldots, N$, where $k_i$ is the number of groups, proposed in the $i$-th partition. The problem consists in finding such a partition $\mathcal{C}^*$ of the set $X$, which has the following properties [188]

(i) Conformity with the family of partitions $\mathfrak{C}$, i.e. the partition $\mathcal{C}^*$ ought to reflect the essential features of each partition $\mathcal{C}^i \in \mathfrak{C}$.
(ii) Robustness with respect to small disturbances in $\mathfrak{C}$, namely the number of clusters and their content ought not undergo drastic changes under the influence of slight disturbances of the partitions, forming the family $\mathfrak{C}$.

[48] Information on this data set is provided in Chap. 7.

(iii) Conformity with the additional information on the elements of the set $X$, provided this information is available. Thus, e.g., if the assignment of (all or some) objects to classes is known, the partition $\mathcal{C}^*$ ought to be to the maximum degree in agreement with this assignment.

To measure the degree of agreement between the partitions, forming the family $\mathfrak{C}$, Fred and Jain applied in [188], similarly, anyway, as Strehl and Ghosh in [447], the normalised measure of mutual information $NMI(\mathcal{C}^\alpha, \mathcal{C}^\beta)$, where $\mathcal{C}^\alpha, \mathcal{C}^\beta$ denote the partitions compared. The form and the properties of this measure are considered in detail in Sect. 4.4.3. We only note here that $NMI(\mathcal{C}^\alpha, \mathcal{C}^\beta)$ is a number from the interval [0, 1].

The degree of agreement of the partition $\mathcal{C}^*$ with the partitions from the family $\mathfrak{C}$ is calculated as

$$NMI(\mathcal{C}^*, \mathfrak{C}) = \frac{1}{N} \sum_{i=1}^{N} NMI(\mathcal{C}^*, \mathcal{C}^i) \tag{2.56}$$

Let, further on, $\mathfrak{P}(k) = \{\mathcal{P}_1(k), \ldots, \mathcal{P}_{\vartheta(m,k)}(k)\}$ denote all the possible partitions of the set $X$ into $k$ disjoint classes. It should be remembered that $\vartheta(m, k)$ is the number, defined by the Eq. (2.23), of all the possible partitions of an $m$-element set $X$ into $k$ disjoint classes. Hence, as $\mathcal{C}^*$ we can take the partition

$$\mathcal{C}^* = \underset{1 \le and \le \vartheta(m,k)}{\arg\max} \; NMI(\mathcal{P}_i(k), \mathfrak{C}) \tag{2.57}$$

This partition satisfies the first of the postulates, formulated before. In order to account for the requirements of flexibility, the authors quoted here form with a bootstrap method an $M$-element family of partitions $\mathbb{B} = \{\mathfrak{B}_1, \ldots, \mathfrak{B}_M\}$, randomly assigning objects from the set $X$, with repetitions, to the appropriate sets from the family $\mathfrak{C}$. A more extensive treatment of the subject, along with a description of the performed experiments, is provided in [188, 189].

Alternative methods of aggregating multiple partitions of the data set are considered by Strehl and Ghosh [447]. Hore, Hall and Goldgof [257] formulate the procedure that ensures scalability of aggregation of partitions (represented by the gravity centres of classes), corresponding to a sparse data set. These latter authors consider two situations: (a) in each portion of data the same number of classes is distinguished, or (b) the numbers of classes are different. In the first case these authors obtain the so-called BM (*Bipartire Merger*) algorithm, and in the second case—the MM (*Metis Merger*) algorithm. An additional strong point of this work is the rich bibliography, concerning the families of partitions.

Then, Thangavel and Visalakshi [458] describe the application of the families of partitions in the $k$-harmonic means algorithm.[49] Finally, Kuncheva and Vetrov wonder in [313] whether the outputs from cluster ensembles are more stable than partitions obtained from the a single clustering algorithm. They understand stability as

[49] This algorithm is presented in Sect. 3.1.5.4 of this book.

sensitivity (or, more precisely, lack of sensitivity) with respect to small disturbances in the data or in the parameters of the grouping algorithm. The considerations therein allowed for the formulation of certain recommendations, concerning the selection of the number of clusters, resulting from the analysed family of partitions.

Let us mention, at the end, one more method of aggregating the partial groupings, which is used in microarray analysis.

*Example 2.5.2* One of the most popular applications of clustering in bioinformatics is microarray analysis. Suppose we treat $X$ as a matrix with rows corresponding to genes, and columns—to experiments or samples. The value of $x_{ij}$ corresponds to the level of expression of the $i$-th gene in the sample (experiment) $j$. In typical applications from this domain the matrix $X$ is exceptionally "slender": it has thousands of rows and not more than 100 columns.

The study [92] presents the problem of formation of meta-genes, being the linear combinations of $n$ genes. In solving this problem, the non-negative factorisation of matrices is used, mentioned here in Sect. 2.4.2.1, i.e. finding of such matrices $W$ and $H$, whose product $WH^{\mathrm{T}}$ is an approximation of the matrix $X$. In this concrete case columns of the matrix $W$ correspond to meta-genes (or to diagnostic classes), while the number $w_{ij}$ defines the value of the coefficient of contribution from the $i$-th gene in the $j$-th meta-gene. Then, the elements $h_{ij}$ of the matrix $H$ indicate the levels of expression of the meta-gene $j$ in sample $i$. Matrix $H$ is made use of for grouping of samples: $i$-th sample is assigned to this meta-gene $j^*$, which corresponds to the maximum value of $h_{ij}$.

In the general case, by performing the decomposition of the matrix $X$ many times over, we obtain the set of matrices $(W^t, H^t)$, where $t = 1, \ldots, t_{max}$ denotes the successive number of the NMF decomposition, while $t_{max}$ is the total number of the decompositions performed.

In the study, reported in [92], the following manner of aggregating the partial results was applied. Let $C^t$ denote the concordance matrix, obtained in experiment $t$, having dimensions $m \times m$. Its element $c^t_{ij} = 1$ if genes $i$ and $j$ belong to the same class, and $c^t_{ij} = 0$ in the opposite case. Let, further, $\overline{C} = (C_1 + \cdots + C_{t_{max}})/t_{max}$ be the aggregate (averaged) concordance matrix. The numbers $\overline{c}_{ij}$ can be treated as the degrees of similarity of the gene pairs $(i, j)$. By turning similarities into distances, that is—by forming the matrix $D$, having elements $d_{ij} = 1 - \overline{c}_{ij}$, we construct the dendrogram and calculate the cophenetic correlation coefficient (see Example 2.3.1 in p. 31), indicating the degree of agreement between the distances, contained in matrix $D$, and the distances, resulting from the dendrogram developed. If the results of grouping, obtained in each run of the algorithm are similar (meaning that the grouping obtained has a stable character), then the elements of matrix $\overline{C}$ (and of matrix $D$) will have values close to 0 or 1, and the calculated correlation coefficient will be close to 1. In case, when the data set analysed does not represent a clear $k$-group structure, the correlation coefficient shall have value well below 1. In addition, the resulting dendrogram serves in the ordering of column and rows of the concordance matrix. The use is made here of the order, in which the leaves of the dendrogram are marked. In the left hand part of Fig. 2.11 the unordered matrices of concordance,

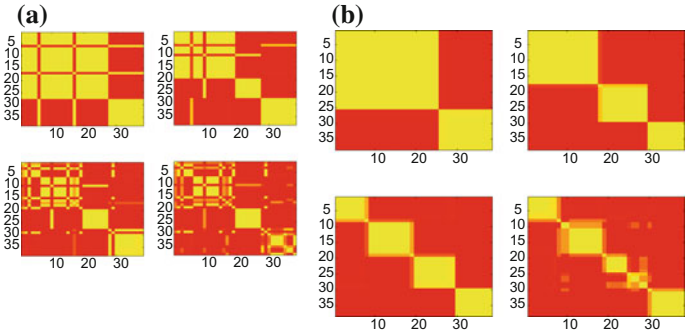



**Fig. 2.11** Unordered (**a**) and ordered (**b**) concordance matrices, generated with the use of the method, described in the text. Experiments were carried out for $k = 2, 3, 4, 5$ groups

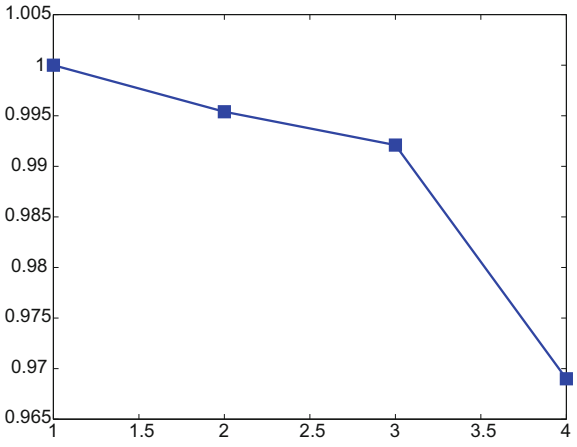



**Fig. 2.12** Values of the cophenetic correlation coefficient for the partitions obtained. The horizontal axis represents the number of groups, the vertical axis shows the cophenetic correlation coefficient

$\overline{C}$, are shown, as obtained for $k = 2, 3, 4, 5$, while in the right-had part—the same matrices with the appropriately ordered rows and columns. Matrix $X$ represents, in this case, the levels of expression of 5000 genes, registered in 38 samples, taken from the bone marrow.[50] Conform to the claim from the authors of [92], the method here outlined allows for a precise distinction between two types of leukaemia (myeloid and lymphoblastic leukaemia), this being indicated in the upper matrix in part (b) of Fig. 2.11. A reader, interested in the interpretation of the remaining figures is kindly referred to the publication [92].

It should be noted that the quality of partitions, as measured by the cophenetic correlation coefficient, decreases with the number of groups—see Fig. 2.12. This corresponds to the existence of less distinct structures in the data set, so that the algorithm is not capable of determining them sufficiently precisely. □

[50]The data, as well as the MATLAB code, are available at http://www.broadinstitute.org/cgi-bin/cancer/publications/pub_paper.cgi?mode=view\&paper_id=89.

## 2.6 Whether and When Grouping Is Difficult?

When treated as an optimisation task, grouping is a "hard" problem, if we consider it in the context of pessimistic complexity (the worst case analysis). This means that with the increase of the number of observations there is a dramatic increase in the pessimistic time complexity, associated with finding the global optimum of the criterion function. If, however, the data represent the true clusters, and the number of observations is sufficiently high, then use can be made of a number of methods of local search, allowing for the correct identification of groups. This may lead to the conviction that "grouping is not difficult; it is either easy or not interesting" [441].

It turns out, in fact, that if the data originate from a (well separable) mixture of normal distributions, and the number of observations is sufficiently high, then the task of grouping is easy. There exists an algorithm, having polynomial time complexity, which identifies—with high probability—the correct division into groups. In particular, this algorithm locates sufficiently precisely (with an assumed error) the gravity centres of groups. This allows for formulating the upper bound on the computational conditioning of the grouping algorithm, that is—the minimum gap between groups and the minimum number of observations in the sample, ensuring correct classification. So, for instance, for an arbitrary Gaussian mixture, if only an appropriate number of observations exist, then the maximum likelihood estimates tend to the true parameters, provided that the local maxima of the likelihood function are lower than the global maximum [404].

One can, of course—and, in fact, should—ask what the "sufficiently high" number of observations means, that is—what is the information limit for the task of grouping. This issue is discussed also in the already cited work [441].

Dasgupta and Schulman [134] concentrate on the mixture of $n$-dimensional spherical normal distributions $N(\boldsymbol{\mu}, \sigma \mathbb{I}_n)$—see Fig. 2.13. The data, which come from a spherical normal distribution, can be enclosed inside the hypersphere with the

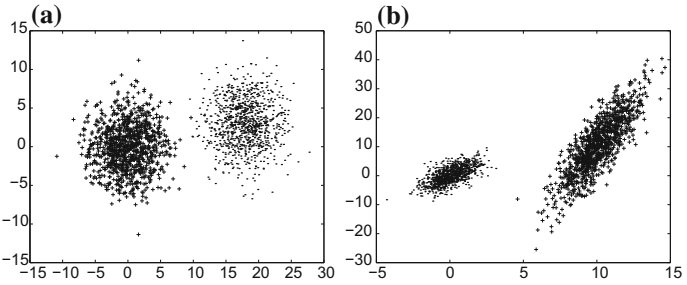



**Fig. 2.13** Exemplary data having: **a** spherical and **b** non-spherical normal distribution

radius[51] $r = \sigma\sqrt{n}$. Therefore, it can be assumed that the data, originating from two distributions, $N(\boldsymbol{\mu}_1, \sigma_1\mathbb{I}_n)$ and $N(\boldsymbol{\mu}_2, \sigma_2\mathbb{I}_n)$ are $c$-separable, if [134]

$$\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\| \geq c\max(\sigma_1, \sigma_2)\sqrt{n} = c\max(r_1, r_2) \tag{2.58}$$

We have endowed, in this manner, the notion of separability of distributions, with a precise meaning.[52] In particular, when we deal with anisotropic multi-normal distributions $N(\boldsymbol{\mu}, \Sigma)$, where $\Sigma$ denotes the covariance matrix, then $r = \sqrt{tr(\Sigma)}$. In the case of a mixture of $k$ distributions, we denote by $c_{ij}$ the separability of the $i$-th and $j$th distribution, and by $c = \min_{i\neq j} c_{ij}$ the separability of the mixture. So, e.g., the 2-separable mixture of distributions represents almost exclusively the disjoint clusters of $n$ dimensional points, and with the increasing $n$ lower and lower value of $c$ is required to secure the disjointness of clusters, this being due to the specific properties of the Euclidean distance (see Sect. 2.2).

If the mixture of $k$ spherical normal distributions is sufficiently separable (at the order of $\Omega(n^{1/4})$), and the sample contains $O(k)$ observations, then it suffices to perform two iterations of the EM algorithm.[53] Further advance is achieved by the application of the spectral projection methods (considered in Chap. 5). Vempala and Wang [479] show that if $c$ is a constant of the order $\Omega(n^{1/4}\log^{1/4} nk)$ and the dimension of the sample is of the order $\Omega(n^3k^2\log ckn/\delta)$, then the $k$-dimensional spectral projection allows for identification of the group centres with probability $1-\delta$.

Kanungo et al. adopt, as the measure of separation of clusters, in [281], the quotient

$$sep = \frac{r_{min}}{\sigma_{max}} \tag{2.59}$$

where $r_{min}$ is half of distance between the closest centres of classes, while $\sigma_{max}$ denotes the maximum of the standard deviations, characterising clusters. The authors quoted show, see Theorem 1 in [281], that if the class centres are sufficiently close to the gravity centres of clusters, then as the value of $sep$ increases, the time of execution of the appropriately implemented $k$-means algorithm improves.

A similar conclusion was formulated by Zhang in the report [531]. Call *clusterability* a measure characterising the partition $\mathcal{C}$ of the set $X$. For a given partition $\mathcal{C} = \{X_1, \ldots, X_k\}$ it is, for instance, possible to define the within-group variance, $W_C(X) = \sum_{j=1}^{k} p_j\sigma^2(X_i)$, and the between-group variance, $B_C(X) = \sum_{j=1}^{k} p_j\|\boldsymbol{\mu}_j - \boldsymbol{\mu}\|$, where $p_i = |X_i|/m$, $\boldsymbol{\mu}_j$ is the gravity centre of the $j$-th group,

[51] It is, actually, the approximate average length of the random vector, having exactly this distribution. If $\mathbf{x}$ is a vector, having as coordinates random numbers distributed according to $N(0, \sigma)$, then its expected length is $\mathbb{E}(\|\mathbf{x}\|) = \sigma\sqrt{2}\Gamma((n+1)/2)/\Gamma(n/2)$. In an approximation, $\mathbb{E}(\|\mathbf{x}\|) \approx \sigma\sqrt{2}[1 - 1/(4n) + 1/(21n^2)]$, and so $\mathbb{E}(\|\mathbf{x}\|) \to \sigma\sqrt{2}$ when $n \to \infty$.

[52] Recall, that on p. 12 we have listed a number of requirements for the separation notion in general, and on pp. 35, 28, and 33 we have already mentioned a couple of other separation measures, intended for more general types of probability distributions.

[53] This algorithm is commented upon in Sect. 3.2.

and $\boldsymbol{\mu}$ is the gravity centre of the entire set $X$. Then, the measure of clusterability is the quotient [531]

$$C(X) = \max_{\mathcal{C} \in \mathfrak{C}} \frac{B_C(X)}{W_C(X)}$$

where $\mathfrak{C}$ is the set of all possible clusterings of the set $X$. The higher the value of $C(X)$, the more separate are individual groups. One of the results, presented in the report quoted, proposes that the higher the clusterability (corresponding to the existence of natural clusters), the easier it is to find the appropriate partition [2].

For other notions of clusterability, less dependent on a particular form of probability distribution, see Definition D.1.5 and its subsequent discussion on p. 348.

Last not least, when choosing an algorithm and evaluating its results, we shall keep in mind the ultimate goal—the consumption of clustering results. If we seek a way to create a taxonomy, a catalogue of e.g. a large collection of products, we would apply a hierarchical clustering algorithm and test how well the hierarchy represents distances between products. If we look for covering an area with points of sales, we will apply a partitioning algorithm and verify whether or not our consumers have a reasonable distance to the closest POS and weight the loss of consumers against our investment budget. If we want to develop effective marketing strategies, we will probably seek well separated clusters.

Let us terminate this section with the following statement. The assessment of quality of a concrete tool, in this case—an algorithm of grouping—remains, actually, in the competence of the person, using the tool. Thus, instead of looking for the "best" tool, one should rather consider the available algorithms in the categories of their *complementarity*: the capacity of compensating for the weak points of one algorithm with the qualities of the other, or the capacity of strengthening the positive qualities of an algorithm by some other one.

http://www.springer.com/978-3-319-69307-1

Modern Algorithms of Cluster Analysis

Wierzchon, S.; Klopotek, M.

2018, XX, 421 p. 51 illus., Hardcover
ISBN: 978-3-319-69307-1