Behaviormetrics:
Quantitative Approaches to Human Behavior 15

Sadaaki Miyamoto

Theory of Agglomerative Hierarchical Clustering

Springer

Behaviormetrics: Quantitative Approaches to Human Behavior

Volume 15

Series Editor

Akinori Okada, Professor Emeritus, Rikkyo University,
Tokyo, Japan

This series covers in their entirety the elements of behaviormetrics, a term that encompasses all quantitative approaches of research to disclose and understand human behavior in the broadest sense. The term includes the concept, theory, model, algorithm, method, and application of quantitative approaches from theoretical or conceptual studies to empirical or practical application studies to comprehend human behavior. The Behaviormetrics series deals with a wide range of topics of data analysis and of developing new models, algorithms, and methods to analyze these data.

The characteristics featured in the series have four aspects. The first is the variety of the methods utilized in data analysis and a newly developed method that includes not only standard or general statistical methods or psychometric methods traditionally used in data analysis, but also includes cluster analysis, multidimensional scaling, machine learning, corresponding analysis, biplot, network analysis and graph theory, conjoint measurement, biclustering, visualization, and data and web mining. The second aspect is the variety of types of data including ranking, categorical, preference, functional, angle, contextual, nominal, multi-mode multi-way, contextual, continuous, discrete, high-dimensional, and sparse data. The third comprises the varied procedures by which the data are collected: by survey, experiment, sensor devices, and purchase records, and other means. The fourth aspect of the Behaviormetrics series is the diversity of fields from which the data are derived, including marketing and consumer behavior, sociology, psychology, education, archaeology, medicine, economics, political and policy science, cognitive science, public administration, pharmacy, engineering, urban planning, agriculture and forestry science, and brain science.

In essence, the purpose of this series is to describe the new horizons opening up in behaviormetrics — approaches to understanding and disclosing human behaviors both in the analyses of diverse data by a wide range of methods and in the development of new methods to analyze these data.

More information about this series at https://link.springer.com/bookseries/16001

Sadaaki Miyamoto

Theory of Agglomerative Hierarchical Clustering

Springer

Sadaaki Miyamoto
University of Tsukuba
Tsukuba, Ibaraki, Japan

ISSN 2524-4027 ISSN 2524-4035 (electronic)
Behaviormetrics: Quantitative Approaches to Human Behavior
ISBN 978-981-19-0419-6 ISBN 978-981-19-0420-2 (eBook)
https://doi.org/10.1007/978-981-19-0420-2

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

This small book is aimed for introducing the theory of agglomerative hierarchical clustering. Prospective readers are researchers in clustering as well as users interested in applications of this method. Readers in the first class, including Ph.D. course students interested in theoretical aspects, may find the contents are quite different from those in other textbooks. Readers in the second class may think it unnecessary to read this book, as it is theoretically oriented. However, the author's fundamental belief is that theory and applications should not be separated.

Generally far more methodological works have been done in the area of non hierarchical than hierarchical clustering, although the usefulness of agglomerative hierarchical clustering techniques is more clearly appreciated in a variety of academic fields. This means that it seems difficult, unpromising, and even useless to carry out theoretical studies in agglomerative hierarchical clustering, after the collection of standard techniques has been established in early studies and user-friendly computer programs have been developed. However, to ignore or to overlook theoretical aspects will result in wrong use that may be found in many application studies. Hence to emphasize theory is important in any fields, and agglomerative hierarchical clustering is not exceptional.

This book thus has a number of novel features. First, the style of mathematics, e.g., propositions/theorems and their proofs are described. This style necessarily leads us to the central topic of the single linkage together with the introduction of the concept of *fuzziness* that is not found in other books. Moreover some more recent algorithms are introduced as variations of the single linkage, uncovering their theoretical properties.

Second feature is the description of algorithms of the whole procedure of agglomerative hierarchical clustering including the output of the dendrogram. The author took the classical style of the algorithm, since the key points are data structure as a tree and the recursive call in drawing a dendrogram together with the tree traversal technique. Thus reference to works in computer algorithms was necessary, although no advanced knowledge is needed to understand the text herein. Thus a general procedure of agglomerative hierarchical clustering is introduced as **AHC** algorithm and each clustering method such as the single linkage and the complete linkage is

described as a selection in **AHC** procedure. Theoretical results such as reversals in dendrograms are handled by referring to **AHC**.

More recent methods are also described. They include the use of positive-definite kernels, agglomerative hierarchical clustering with constraints, and handling of asymmetric similarity measures. The use of positive-definite kernels is usual in the support vector machine, one of the most popular technique in supervised classification, but its application to the present topic of hierarchical clustering is generally unknown. In particular, the well-known Ward method is sometimes used in applications with indefinite kernel which appears to be *ad hoc* and theoretically not justified. This book shows, however, that the use of indefinite kernel can still be justified by introducing the concepts of similar dendrograms and a regularized kernel. Moreover old methods such as Lance-Williams formula are discussed within the present framework.

Overall, the studies of clustering situate in between statistics and machine learning. This means that researchers in clustering should refer to literature in both fields. At the same time, both new and old literature should be referred to. It's a pity, however, that old studies in clustering are often overlooked, partly due to inaccessibility to those old papers and books. The author would like to emphasize that there are old works that should be remarked in new views. Indeed, some old studies, maybe forgotten, greatly inspired the author, who is thankful to former researchers in this area.

The author expresses his deep appreciation to Dr. Akinori Okada, the chief editor of this series for his encouragement to finish this small book. He is grateful to an anonymous reviewer for her/his useful comments. Moreover, he is thankful to Mr. Yutaka Hirachi and Ms. Sridevi Purushothaman for their help in preparing the manuscript.

Tsukuba, Japan
July 2021

Sadaaki Miyamoto
Professor Emeritus

Contents

Chapter 1
Introduction

The word of *cluster analysis* alias *clustering* has become popular nowadays in the field of data analysis. It means a family of methods of grouping objects. It contrasts with a still more popular method of *supervised classification* that means an external classification rule on a set of objects (but not for the entire space) is given and the problem is how to extend the rule onto the entire space. On the other hand, a method of clustering generates a family of groups called *clusters* using a *similarity* or *dissimilarity measure* without an external rule. (The meaning of similarity/dissimilarity measure will be given in the sequel.) In this sense clustering is sometimes referred to as *unsupervised classification*, since it does not have an external supervisor.

A *cluster* is a group of objects that are mutually similar. When a set of objects are given with a similarity/dissimilarity measure, we usually find multiple clusters. Hence two objects in different clusters should not be similar. This means that there are two criteria for clustering. One divides clusters, each of which has similar objects, and the other uses a criterion by which objects in different clusters are not similar.

Methods of clustering can be divided into a number of categories: there are *agglomerative hierarchical clustering*, *K-means* and related methods, *mixture of distributions*, and so on. The subject of this book is the agglomerative hierarchical clustering, except that a basic algorithm of K-means is also shown in relation to the main subject. The agglomerative hierarchical clustering is also called the hierarchical agglomerative clustering, and the former is used throughout this book.

This book moreover focuses on the theoretical aspect of the agglomerative hierarchical clustering, some of which are not found in other books. Why the author selected this subject and this aspect is that the agglomerative hierarchical clustering is more frequently used than other methods, and theoretical properties have often been overlooked, as researchers in applications often regarded them unnecessary or cumbersome. Methods of agglomerative clustering which is referred to as *linkage methods* were studied in 1950 and 1960s. We can mention Sokal and Sneath [1], Anderberg [2], and Everitt [3] as representatives of early literature. After

S. Miyamoto, *Theory of Agglomerative Hierarchical Clustering*,
Behaviormetrics: Quantitative Approaches to Human Behavior 15,
https://doi.org/10.1007/978-981-19-0420-2_1

that researchers thought the theory was already finished and further theoretical development is difficult. However, readers will see new facts in this subject which will give new perspectives in the agglomerative hierarchical clustering.

Basic notations for clustering and basic concepts are first given that are necessary for discussions throughout this book.

1.1 Notations

We begin with basic notations. $X = \{x_1, \ldots, x_N\}$ is the finite set of objects for clustering. A cluster G_i is a nonempty subset of objects: $G_i \subseteq X$. There are multiple clusters and the number of clusters is denoted by K. Unless stated otherwise, clusters form a *partition* of X:

$$\bigcup_{i=1}^{K} G_i = X, \qquad G_i \cap G_j = \emptyset \ \ (i \neq j), \tag{1.1}$$

i.e., the union of all clusters is equal to the entire set X and there is no common element between arbitrary two clusters. Moreover the collection of G_i $(1 \leq i \leq K)$ is denoted by $\mathcal{G} = \{G_1, \ldots, G_K\}$.

Note 1 There are methods generating clusters that do not satisfy (1.1) such as overlapping clusters, fuzzy clusters, and probabilistic clusters, but we do not consider them in this book.

The above definition of clusters is not sufficient for defining *hierarchical clusters*. We introduce an integer or real parameter α and consider a case when clusters are dependent on this parameter: $G_i(\alpha)$ and $\mathcal{G}(\alpha) = \{G_1(\alpha), \ldots, G_{K(\alpha)}(\alpha)\}$, assuming these clusters satisfy (1.1). Note also that the number of clusters $K(\alpha)$ is also dependent on the parameter.

We moreover define that the family of clusters $\mathcal{G}(\alpha)$ is hierarchical. For this purpose we define the refinement of two clusters.

For given two clusters $\mathcal{G}$ and $\mathcal{G}'$, $\mathcal{G}$ is called a *refinement* of $\mathcal{G}'$ if for every $G_i \in \mathcal{G}$, there exists $G'_j \in \mathcal{G}'$ such that $G_i \subseteq G'_j$.

If we use a symbol $\prec$ for refinement: $\mathcal{G} \prec \mathcal{G}'$, then $\prec$ is reflexive and transitive.

Now a family of clusters $\mathcal{G}(\alpha)$ is called *hierarchical* if for every two parameters α and α' such that $\alpha < \alpha'$, $\mathcal{G}(\alpha)$ is a refinement of $\mathcal{G}(\alpha')$. The order of the parameter can be reversed: we also call $\mathcal{G}(\alpha)$ hierarchical when $\mathcal{G}(\alpha)$ is a refinement of $\mathcal{G}(\alpha')$ for every $\alpha > \alpha'$.

A method of hierarchical clustering is not necessarily agglomerative: there are *divisive hierarchical clustering* and other methods. A method of agglomerative hierarchical clustering which is the main subject herein means a specific algorithm and linkage methods which will be introduced below.

Vectors and Matrices

Special notations for vectors and matrices are used in standard textbooks. Many readers know, for example, $\mathbf{x} = (x_1, \ldots, x_M)$ and $\mathbf{A} = (a_{ij})$, respectively, for a vector and a matrix, or in other cases, $\boldsymbol{x} = (x_1, \ldots, x_M)$ and $A = (a_{ij})$. The former *bold face* symbols are used in statistics, and the latter *bold mathematics* symbols are used in other fields of mathematics. In spite of these conventional usages, we simply use the italic symbols for the both: $x = (x^1, \ldots, x^M)$ (the components are shown by superscript here instead of subscript) and $A = (a_{ij})$ for simplicity. Our aim is to save symbols as well as to make descriptions clearer. Specifically, an object is sometimes a vector and sometimes not. We need to use a common symbol x for the both cases, in other words, we do not need to distinguish x and $\mathbf{x}$. Moreover, a matrix in this book corresponds to a relation, say S (see Sect. 1.6 for the definition of a relation). Thus we use the same symbol S for the relation and the matrix, since we do not need to distinguish the both. Thus if we use the bold face notations, the descriptions will become more complicated, which is far from the intention of the author.

1.2 Informal Procedure

Let us introduce a simple procedure of agglomerative hierarchical clustering. A similarity measure $S(x, y)$ or dissimilarity measure $D(x, y)$ between two objects x and y is used of which the details are described below. For the moment we should know that a large $S(x, y)$ means that x and y are similar, whereas a small $D(x, y)$ means x and y are similar. These measures are generalized into measures between two clusters G and G': $S(G, G')$ and $D(G, G')$ using a linkage method which will be described after the procedure in this section.

We now state an informal procedure for agglomerative hierarchical clustering.

Step 0: Make initial clusters: each cluster has each object. There are hence N clusters: the number of clusters K is equal to the number of objects: $K = N$.
Step 1: Search the pair of clusters having the maximum similarity from the set of all similarity values (in the case of dissimilarity, search the pair of minimum dissimilarity).
Step 2: Merge the pair of clusters found in Step 1 and make a new cluster.
Step 3: Keep data records of the merging in Step 2. If the number of clusters is one, output the whole merging process as a *dendrogram* and stop.
Step 4: Since similarities between the new (merged) cluster and other clusters are not yet defined, define (in other words, update) new similarity values (or dissimilarity values).

The above procedure is imperfect as an algorithm, i.e.,

- how to output a dendrogram with certain data records is not described, and
- how to define new similarity values is not shown.

The first part is not shown in many textbooks, but we will show an algorithm to output a dendrogram. The second part is known as the selection of a linkage method, which is written in all textbooks; we will describe them using new notations which will make the situation clearer.

1.3 *K*-means as a Method of Non Hierarchical Clustering

Although this book is for agglomerative hierarchical clustering, we should describe an important *non hierarchical method*. This method is called *K-means* and known very well.

The method of K-means (MacQueen [4] first called the method k-means, but the algorithm itself is known to be older than his proposal.) simply divides a given set of objects into K clusters. We assume that an object is a real-valued vector: $x_k = (x_k^1 \ldots, x_k^M)$ $(1 \leq k \leq N)$.

Although there are variations, the basic procedure is as follows.

KM: K-means algorithm.

KM0: Generate randomly a partition $G_1, \ldots, G_K$ of X (or alternatively, give randomly K centers $v_1, \ldots, v_K$; in this case skip **KM1**).

KM1: Calculate centers v_i for G_i $(i = 1, \ldots, K)$:

$$v_i = \frac{1}{|G_i|} \sum_{x_k \in G_i} x_k, \tag{1.2}$$

where $|G_i|$ is the number of objects in G_i.

KM2: Allocate every object x_k $(K = 1, \ldots, N)$ to the cluster of nearest center:

$$x_k \in G_i \iff i = \arg \min_{1 \leq j \leq K} \|x_k - v_j\|^2. \tag{1.3}$$

and make new $G_1, \ldots, G_k$.

KM3: If clusters are convergent, stop. Else go to **KM1**.

End KM.

1.4 Similarity or Dissimilarity Measures

We already used the words of similarity and dissimilarity measures. There are standard textbooks [2, 5] discussing these measures in detail and interested readers would refer to these literature. In this section the discussion of similarity/dissimilarity is limited to those which are just needed in this book.

Let us remind that we already introduced the set of objects $X = \{x_1, \ldots, x_N\}$ and a similarity $S(x, y)$ or dissimilarity $D(x, y)$ without showing how we can concretely define them. Moreover we used the symbol of the *squared Euclidean distance*:

$$\|x - y\|^2 = \sum_{j=1}^{M} (x^j - y^j)^2, \tag{1.4}$$

where $x = (x^1 \ldots x^M)$ and $y = (y^1, \ldots y^M)$ are objects as points of M-dimensional space.

The *squared Euclidean distance* (and not the Euclidean distance itself) is the most common dissimilarity measure encountered in clustering and other areas of data analysis. We thus use dissimilarity

$$D(x, y) = \|x - y\|^2 \tag{1.5}$$

in this case.

Another important space is ℓ_1*-metric* when objects are points: $x = (x^1 \ldots x^M)$ and $y = (y^1, \ldots y^M)$ with the same symbol as above. The ℓ_1 metric is defined by

$$\|x - y\|_1 = \sum_{j=1}^{M} |x^j - y^j|, \tag{1.6}$$

where $|\cdot|$ shows the absolute value. In the latter case we put

$$D(x, y) = \|x - y\|_1. \tag{1.7}$$

The ℓ_1 metric is less frequently used than the squared Euclidean distance.

It is easy to see that these two measures are *symmetric*:

$$D(x, y) = D(y, x)$$

and

$$D(x, y) \geq 0$$

where $D(x, y) = 0$ if and only if $x = y$.

These two are dissimilarity measures. In contrast, a well-known similarity measure in the case of Euclidean space is the *cosine correlation coefficient*, i.e.,

$$S(x, y) = \frac{\sum_{j=1}^{M} x^j y^j}{\|x\|\|y\|}, \tag{1.8}$$

for two non zero vectors x and y. It is not difficult to see that

$$-1 \leq S(x, y) \leq 1$$

and $S(x, y) = 1$ if and only if $x = const.y$, where $const.$ is a positive constant. The proof uses the well-known Schwarz inequality:

$$|\sum_{j=1}^{M} x^j y^j| \leq \|x\|\|y\|. \tag{1.9}$$

In general, a similarity measure is normalized ($0 \leq S(x, y) \leq 1$ or $-1 \leq S(x, y) \leq 1$), while a dissimilarity measure not ($0 \leq D(x, y) \leq \infty$). An object should be the most similar to itself: $S(x, x) = 1$ and $D(x, x) = 0$ and they should be symmetric ($S(x, y) = S(y, x)$ and $D(x, y) = D(y, x)$), although there are asymmetric measures and also clustering techniques handling those *asymmetric measures* [6]. We consider a method for an asymmetric measure in a later chapter.

Note also that the metric property including the *triangular inequality*:

$$m(x, z) \leq m(x, y) + m(y, z) \tag{1.10}$$

is unnecessary for a dissimilarity measure. Indeed, the squared Euclidean distance, which is not the Euclidean distance itself, does not satisfy the last property.

The above three measures assume the existence of an underlying continuous space. In contrast, there are other two types of data:

1. The underlying space is discrete: results of mathematical continuity cannot be applied.
2. The underlying space is not referred to.

The best known case of the first type is 0/1-*valued data* (also called *binary data*): $x^j = 0$ or $x^j = 1$ in $x = (x^1, \ldots, x^j, \ldots, x^M)$, where 2×2 *table* is often considered. The symbol a in Table 1.1 means the number of j's where $x^j = 0$ and $y^j = 0$, b means the number of j's where $x^j = 0$ and $y^j = 1$, c means the number of j's where $x^j = 1$ and $y^j = 0$, and d means the number of j's where $x^j = 1$ and $y^j = 1$. From the assumption $a + b + c + d = M$.

Table 1.1 A 2×2 table

		y	
		0	1
x	0	a	b
	1	c	d

A well-known measure using this table is called the *simple matching coefficient*:

$$S_{smc}(x, y) = \frac{a+d}{M}. \tag{1.11}$$

We have

$$1 - S_{smc}(x, y) = M\|x - y\|_1 = M\|x - y\|^2 \tag{1.12}$$

when all data are $0/1$-valued. Thus, this coefficient is a particular case of the former two measures. Another well-known measure called *Jaccard coefficient* is defined by

$$S_{jac}(x, y) = \frac{a}{a+b+c}, \tag{1.13}$$

which is not justified using the Euclidean or ℓ_1 space.

If we apply (1.8) to $0/1$-valued data, we have

$$S_{cc}(x, y) = \frac{a}{\sqrt{(a+b)(a+c)}}. \tag{1.14}$$

It is easy to see that $0 \leq S_{cc}(x, y) \leq 1$ and $S_{cc}(x, y) = 1$ if and only if $x = y$. Note that $S_{cc}(x, y)$ cannot be applied to the zero vector.

Nominal scale with the meaning that the data are just symbols and without a numerical implication has also been discussed in literature (see, e.g., [5]), but we include this case into the third type below.

Let us consider the third type where $D(x, y)$ or $S(x, y)$ is simply given on the set $X = \{x_1, \ldots, x_N\}$ without reference to the way how the measure is defined. Such a case is accepted in agglomerative hierarchical clustering. Moreover this type can be put into the *network* (X, D) or (X, S), which means that a *graph* with weights on the edges is assumed: the nodes are X, the edges make the complete graph (edge (x, y) is present for every pair $x, y \in X$), and the weight $D(x, y)$ (or $S(x, y)$) is put on the edge (x, y).

For later use, we classify these into three types:

(i) squared Euclidean distance,
(ii) other dissimilarity/similarity based on a space such as the ℓ_1 space or cosine correlation, etc., and
(iii) the case of a network (X, D) or (X, S).

Next chapter discusses linkage methods: some of them can handle all of these types, while some others are based on type (i). Type (ii) needs a special way of handling which will be discussed in later chapters.

Conversion Between Similarity and Dissimilarity Measures

The last topic of this section is how we can convert a similarity measure into a dissimilarity measure and vice versa. We sometimes need to use similarity instead of dissimilarity, or other users may prefer to use dissimilarity more than similarity.

Let us introduce two symbols of $D_{\max}$ and $D_{\min}$:

$$D_{\max} = \max_{x,y \in X} D(x, y), \tag{1.15}$$

$$D_{\min} = \min_{x,y \in X} D(x, y). \tag{1.16}$$

It appears straightforward to have a similarity measure from a given dissimilarity and vice versa as follows:

$$S(x, y) = \frac{D_{\max} - D(x, y)}{D_{\max} - D_{\min}}, \tag{1.17}$$

$$D(x, y) = 1 - S(x, y). \tag{1.18}$$

There are points to be noted. First, a similarity is normalized into the unit interval ($S(x, y) \in [0, 1]$) and $S(x, x) = 1$ while $D(x, y)$ not. Hence the last two equations are not symmetrical.

Second and more important note is that we should not apply such a conversion without reflection to a linkage method in the next section. Some linkage methods like the single linkage and the complete linkage (see the next chapter for the definition) accept such conversion while others not (Ward method and the centroid method).

A more general way of the conversion is to use a strictly monotone decreasing function F defined on $[0, 1]$ with non negative real values such that

$$F(1) = 0, \tag{1.19}$$

$$F(x) < F(y) \iff x > y. \tag{1.20}$$

Thus if we set

$$D(x, y) = F(S(x, y)), \tag{1.21}$$

then $D(x, y)$ is a dissimilarity measure derived from a similarity measure and F. We can define a similarity measure from a dissimilarity and a monotone function in a similar way, which is omitted here.

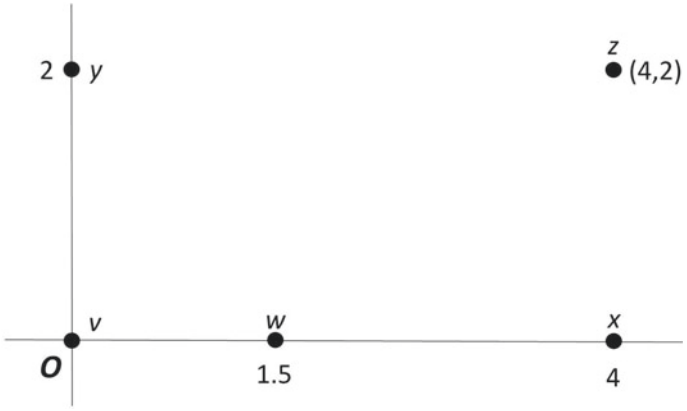

Fig. 1.1 Five points $v = (0, 0)$, $w = (1.5, 0)$, $x = (4, 0)$, $y = (2, 0)$, $z = (4, 2)$ on a plane for a set of objects $X = \{v, w, x, y, z\}$

1.5 Simple Examples

Let us see how clustering algorithms work on a small set of objects as points on a plane. Figure 1.1 shows five points of $\boldsymbol{R}^2$ with coordinates $v = (0, 0)$, $w = (1.5, 0)$, $x = (4, 0)$, $y = (2, 0)$, $z = (4, 2)$ for a set of objects $X = \{v, w, x, y, z\}$.

Example 1 We first apply the informal procedure in this chapter in which the dissimilarity is the Euclidean distance:

$$D(x, y) = \|x - y\|. \tag{1.22}$$

The linkage method in Step 4 of the informal procedure is the *single linkage* (alias the *nearest neighbor linkage*) which uses the rule:

The distance between two clusters is the nearest distance between two objects out of the two clusters
(an object is from a cluster and the other object is from the other cluster.)

The initial clusters are each object: $\{v\}$, $\{w\}$, $\{x\}$, $\{y\}$, $\{z\}$. The above rule implies that the first cluster is formed for the pair of objects with the minimum distance in Fig. 1.1. Hence the first cluster is $\{v, w\}$ as shown by an oval in Fig. 1.2. We next need distances between $\{v, w\}$ and other three objects, which is calculated by the above nearest distance rule. Hence

$$\begin{aligned}
D(\{v, w\}, x) &= \min\{D(v, x), D(w, x)\} = D(w, x) = 2.5, \\
D(\{v, w\}, y) &= \min\{D(v, y), D(w, y)\} = D(v, y) = 2, \\
D(\{v, w\}, z) &= \min\{D(v, z), D(w, z)\} = D(w, z) = \sqrt{2.5^2 + 2^2}.
\end{aligned}$$

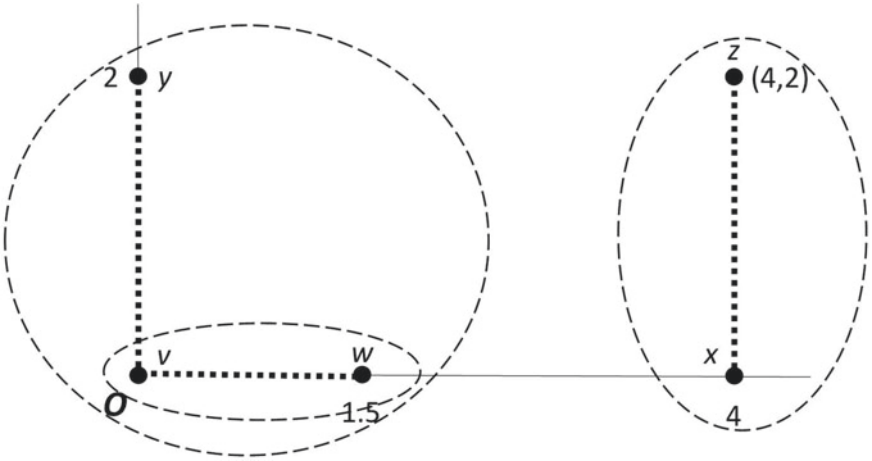

Fig. 1.2 Clusters obtained from the five points in Fig. 1.1 using the single linkage. The clusters are shown as ovals except the final cluster of the five points

Table 1.2 List of clusters using the single linkage for Example 1

Clusters	Level whereby two clusters are merged
$\{v\}, \{w\}, \{x\}, \{y\}, \{z\}$	–
$\{v, w\}, \{x\}, \{y\}, \{z\}$	1.5
$\{v, w, y\}, \{x\}, \{z\}$	2
$\{v, w, y\}, \{x, z\}$	2
$\{v, w, x, y, z\}$	2.5

The second cluster is either $\{v, w, y\}$ or $\{x, z\}$ as the distance is 2 for the both. For such a case of *tie*, we select one of the two (or more in general). If we select $\{v, w, y\}$, we calculate the new distances:

$$D(\{v, w, y\}, x) = D(w, x) = 2.5,$$
$$D(\{v, w, y\}, z) = D(w, z) = \sqrt{2.5^2 + 2^2}.$$

The third cluster is apparently $\{x, z\}$, and we calculate

$$D(\{v, w, y\}, \{x, z\}) = D(w, x) = 2.5.$$

Finally, the two clusters $\{v, w, y\}$ and $\{x, z\}$ are merged at 2.5 into the whole set of one cluster. The clusters are shown in Fig. 1.2 as ovals except the final cluster of the five points. Dotted lines imply distances between clusters.

This process can be summarized in Table 1.2 as a list of clusters and the levels whereby those clusters are formed. The essence of agglomerative hierarchical clus-

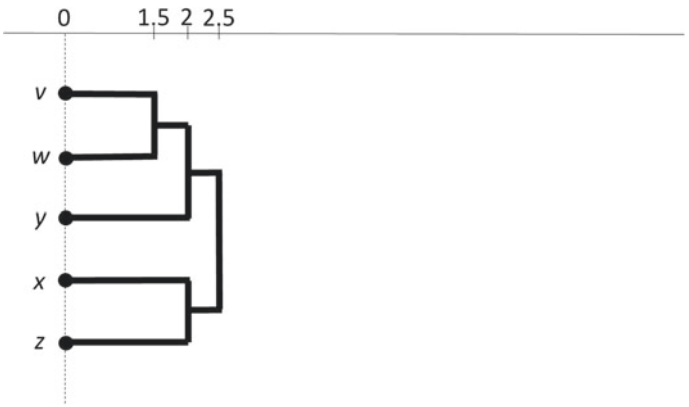

Fig. 1.3 Dendrogram obtained from the five points in Fig. 1.1 using the single linkage

tering is to show such a list of clusters, but this list is too cumbersome to observe, and therefore the *dendrogram* is used.

The dendrogram for Example 1 is shown in Fig. 1.3. As shown in this figure, a dendrogram is a *tree* in which a *leaf* (an end point) is an object and a *branch* shows a link of merging two clusters. Thus clusters $\{v\}$ and $\{w\}$ are first linked at a branch and then the latter is linked with $\{y\}$. On the other hand, $\{x\}$ and $\{z\}$ are linked at the lowest branch and then they are linked to form the one cluster of the whole objects.

The clusters in Table 1.2 are fully shown in this dendrogram, and moreover the dendrogram in Fig. 1.3 is easier to observe clusters than the list of clusters in Table 1.2.

Example 2 The second example uses the same set of points in Fig. 1.1 but with another linkage method called the *centroid method*. The precise description of the centroid method is given in the next chapter, but the rule of calculating distance between clusters is as follows.

The distance between two clusters is the squared Euclidean distance between the two centroids (centers of gravity) of those clusters.

Note that the distance is not the Euclidean distance itself but the squared distance. Thus for the initial clusters of $\{v\}$, $\{w\}$, $\{x\}$, $\{y\}$, $\{z\}$, $D(\{v\}, \{w\}) = 1.5^2$, $D(\{v\}, \{x\}) = 4^2$, and so on.

The first merging occurs between $\{v\}$ and $\{w\}$ which is shown as a small oval in Fig. 1.4. Then $m(\{v, w\})$ in this figure shows the centroid of cluster $\{v, w\}$. The squared distance between $m(\{v, w\})$ and other points are then calculated. The second cluster is $\{x, z\}$, since $\|x - z\|^2 = 4$ is smaller than $\|y - m(\{v, w\})\|^2 \approx 4.56$. The third cluster is $\{v, w, y\}$.

Thus the list of clusters are in Table 1.3, and the dendrogram is shown in Fig. 1.5.

Examples 1 and 2 show how different linkage methods behave and the distances are different for the same set of points on the plane.

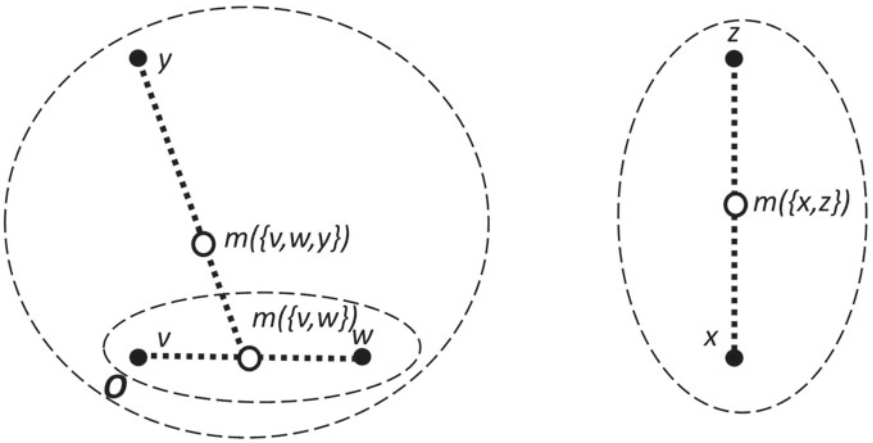

Fig. 1.4 Clusters by the centroid method are shown as ovals. Centroids are shown by small white circles

Table 1.3 List of clusters using the centroid method for Example 2

Clusters	Level whereby two clusters are merged
$\{v\}, \{w\}, \{x\}, \{y\}, \{z\}$	–
$\{v, w\}, \{x\}, \{y\}, \{z\}$	2.25
$\{v, w\}, \{y\}, \{x, z\}$	4
$\{v, w, y\}, \{x, z\}$	4.56
$\{v, w, x, y, z\}$	12.36

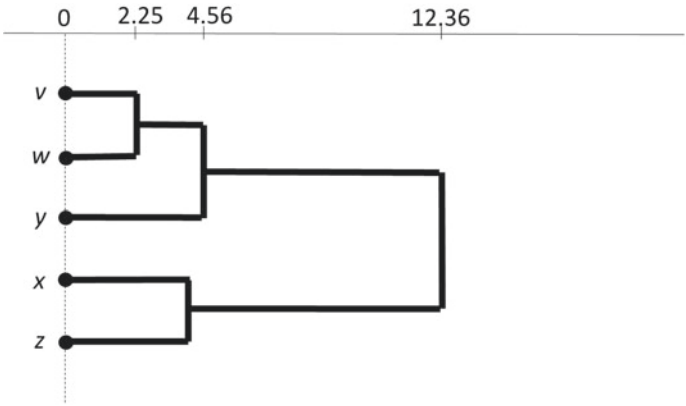

Fig. 1.5 Dendrogram obtained from the points in Fig. 1.1 using the centroid method

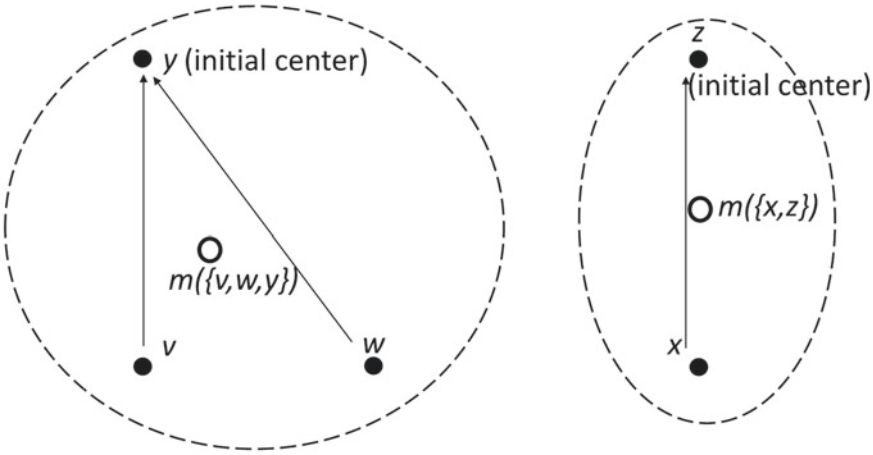

Fig. 1.6 Two clusters by K-means applied to the points in Fig. 1.1

In the next chapter we will show well-known linkage methods including these two with formal precise definitions.

Example 3 The last example in this chapter is not for agglomerative clustering, but for *K-means*. The purpose for considering K-means is for comparison. The set of objects is again the five points in Fig. 1.1.

Let us assume that the number of clusters is $K = 2$; the number of clusters is needed as a parameter for K-means but unnecessary in agglomerative clustering. Initial clusters are also needed and we take two cluster centers as $v_1 = y$ and $v_2 = z$. Then the nearest prototype allocation rule is applied and each object is allocated to the cluster of the nearest center: $v \to v_1$, $w \to v_1$, and $x \to v_2$. Thus the clusters at this moment are $\{v, w, y\}$ and $\{x, z\}$. New cluster centers are calculated as those centroids: $v_1 = m(\{v, w, y\})$ and $v_2 = m(\{x, z\})$ in Fig. 1.6. The nearest prototype allocation rule is applied again, and apparently cluster members are unchanged. Hence the iteration algorithm of K-means stops, as clusters are converged.

1.6 Hierarchical Partitions, Relations, and Dissimilarity Measures

A *relation* R of X is a subset of $X \times X$: $R \subseteq X \times X$. Hence a point $(x, y) \in X \times X$ satisfies $(x, y) \in R$ or $(x, y) \notin R$, and not both. We moreover use special notations for a relation: $(x, y) \in R$ is expressed as $x\,R\,y$ or $R(x, y) = 1$; $(x, y) \notin R$ is expressed as $x \not R y$ or $R(x, y) = 0$. The last expression of $R(x, y) = 1$ or 0 called *prefix notation* is convenient for our purpose and we use this in this section.

Basic four properties for a relation is the following.

(i) A relation R is called *reflexive* if $R(x, x) = 1,\ \forall x \in X$.

(ii) A relation R is called *symmetric* if $R(x, y) = 1 \Rightarrow R(y, x) = 1,\ \forall x, y \in X$.

(iii) A relation R is called *transitive* if $R(x, y) = 1, R(y, z) = 1 \Rightarrow R(x, z) = 1,\ \forall x, y, z \in X$.

(iv) A relation R is called *equivalence relation* if R is reflexive, symmetric, and transitive.

The following theorem is well known.

Theorem 1 *Let R be an equivalence relation on X. Assume that*

$$[x]_R = \{\, y \in X : xRy \,\}. \tag{1.23}$$

Then, $[x]_R$ for all $x \in X$ forms a partition (1.1) of X. Conversely, let $\mathcal{G} = \{G_1, \ldots, G_K\}$ be a partition of X. Define R by

$$R(x, y) = 1 \iff x \text{ and } y \text{ are in a same set } G_i \text{ of } \mathcal{G}. \tag{1.24}$$

In contrast, $R(x, y) = 0$ if and only if x and y are in different sets of $\mathcal{G}$. Then, R is an equivalence relation.

Proof For the first part, note the obvious equation:

$$\bigcup_{x \in X} [x]_R = X. \tag{1.25}$$

We next prove either

$$[x]_R = [y]_R \text{ or } [x]_R \cap [y]_R = \emptyset. \tag{1.26}$$

Suppose $[x]_R \cap [y]_R \neq \emptyset$. Then there exists $z \in [x]_R \cap [y]_R$. Take an arbitrary $w \in [x]_R$. Since $R(w, x) = 1$ and $R(x, z) = 1$, we have $R(w, z) = 1$ from the transitivity. We moreover have $R(w, y) = 1$ from the last equation and $R(y, z) = 1$ and using the transitivity again. The equation $R(w, y) = 1$ is rewritten as $w \in [y]_R$, which in turn implies $[x]_R \subseteq [y]_R$. If we interchange x and y in the above argument, we have $[y]_R \subseteq [x]_R$. We have $[x]_R = [y]_R$ and thus the first part is proved.

For the second part, assume that $\mathcal{G}$ is given and R is defined in the theorem. The reflexive property $R(x, x) = 1$ is trivial. The symmetric property is also trivial, since the definition of R is symmetric for x and y. Let us suppose $R(x, y) = 1$ and $R(y, z) = 1$. This means that all x, y, z are in the same set, say G_i of $\mathcal{G}$. Thus we have $R(x, z) = 1$, which implies the transitivity. □

Fuzzy Relation

Fuzzy relations [7, 8] have been defined in terms of fuzzy sets [9] and their properties have been considered in different ways. We limit ourselves to a generalization of equivalence relation.

A *fuzzy relation* R on X is a $[0, 1]$-valued function defined on $X \times X$: (R: $X \times X \to [0, 1]$), which means that $R(x, y)$ can take 0 and 1, and also any real value between 0 and 1. For the value $R(x, y)$, we attach the meaning of the degree of relatedness of x to y. Thus if $R(x, y) = 1$, x is surely related to y, while if $R(x, y) = 0$, x is not related to y at all. Moreover if $0 < R(x, y) < 1$, the relatedness of x to y is not sure as its value shows. Generally, the interpretation of a fuzzy relation is subjective, but what we consider in this book is more objective after a similarity measure is defined.

Let us define fuzzy versions of the reflexivity, symmetry, and transitivity.

Definition 1 A fuzzy relation R is called *reflexive* if

$$R(x, x) = 1, \ \forall x \in X. \tag{1.27}$$

A fuzzy relation R is called *symmetric* if

$$R(x, y) = R(y, x), \ \forall x, y \in X. \tag{1.28}$$

A fuzzy relation R is called *transitive* if

$$R(x, z) \geq \min\{R(x, y), R(y, z)\}, \ \forall x, y, z \in X. \tag{1.29}$$

A fuzzy relation R is called *fuzzy equivalence relation* if R is reflexive, symmetric, and transitive.

It seems that fuzzy reflexivity and symmetry are simple and natural, but fuzzy transitivity is much more complicated and difficult to check. Note moreover that (1.29) is equivalent to the following:

$$R(x, z) \geq \max_{y \in X} \min\{R(x, y), R(y, z)\}, \ \forall x, z \in X. \tag{1.30}$$

We proceed to observe how a fuzzy equivalence relation is related to non fuzzy (0/1-valued) equivalence relation. We need to define α-*cuts* of fuzzy relations for this purpose.

Definition 2 Assume that a fuzzy relation R on X is given. For an arbitrary $\alpha \in [0, 1]$, an α-cut for R, denoted $[R]_\alpha$, is a non fuzzy (0/1-valued) relation defined as follows:

$$[R]_\alpha(x, y) = \begin{cases} 1 & (R(x, y) \geq \alpha) \\ 0 & (R(x, y) < \alpha) \end{cases} \tag{1.31}$$

We have the next result.

Proposition 1 *For a fuzzy relation R on X, the following properties hold.*

1. *R is reflexive if and only if $[R]_\alpha$ is reflexive for all $\alpha \in [0, 1]$.*
2. *R is symmetric if and only if $[R]_\alpha$ is symmetric for all $\alpha \in [0, 1]$.*
3. *R is transitive if and only if $[R]_\alpha$ is transitive for all $\alpha \in [0, 1]$.*

Proof The proofs for the reflexivity and symmetry are easy and omitted here. Next, suppose that R is transitive, i.e., (1.29) is satisfied, which means that if $R(x, y) \geq \alpha$ and $R(y, z) \geq \alpha$, then $R(x, z) \geq \alpha$. Thus $[R]_\alpha$ is transitive. Conversely, suppose that $[R]_\alpha$ is transitive for all $\alpha \in [0, 1]$. Let $\alpha = \min\{R(x, y), R(y, z)\}$. Then the transitivity of $[R]_\alpha$ implies

$$R(x, z) \geq \alpha = \min\{R(x, y), R(y, z)\}. \tag{1.32}$$

Since $y \in X$ is arbitrary, the above inequality means (1.29). Thus the transitivity of R is proved. □

The next theorem obviously holds.

Theorem 2 *A fuzzy relation R is a fuzzy equivalence relation if and only if all of its α-cuts $[R]_\alpha$ ($\forall \alpha \in [0, 1]$) are equivalence relations. Assume moreover that $\alpha < \alpha'$, then for an arbitrary $x \in X$,*

$$[x]_{[R]_{\alpha'}} \subseteq [x]_{[R]_\alpha}. \tag{1.33}$$

Proof The former half of the theorem (of the if and only if part) is an immediate consequence of Proposition 1.

For the latter half, suppose $\alpha < \alpha'$ and take any $y \in [x]_{[R]_{\alpha'}}$. Then $R(x, y) \geq \alpha'$ and hence $R(x, y) > \alpha$, which means $y \in [x]_{[R]_\alpha}$. The theorem is thus proved. □

Remind that we have defined a hierarchical cluster as a hierarchical partition $\mathcal{G}(\alpha)$. What we have shown above is that a fuzzy equivalence relation R produces a hierarchical cluster $\mathcal{G}(\alpha)$ for $0 \leq \alpha \leq 1$ with the property that for $\alpha < \alpha'$, $\mathcal{G}(\alpha')$ is a refinement of $\mathcal{G}(\alpha)$.

Conversely, assume $\mathcal{G}(\alpha)$ for $0 \leq \alpha \leq 1$ is a hierarchical cluster such that for $\alpha < \alpha'$, $\mathcal{G}(\alpha')$ is a refinement of $\mathcal{G}(\alpha)$. Let us define R by $R(x, y) = \alpha$ if there exists $G \in \mathcal{G}(\alpha)$ such that $x, y \in G$ and for every $\alpha' > \alpha$, there are two different subsets $G', G'' \in \mathcal{G}(\alpha')$ ($G' \cap G'' = \emptyset$) such that $x \in G'$ and $y \in G''$. Then R is a fuzzy equivalence relation.

In summary, a fuzzy equivalence relation is essentially a hierarchical cluster and vice versa. It is difficult to have a fuzzy equivalence relation directly in real applications. Hence we should consider how we can generate a fuzzy equivalence from a given non transitive relation or similarity measure. This topic is considered in a later chapter of the theory of the single linkage.

Dissimilarity and Ultrametric

The previous section is related to a similarity measure, which will be discussed in more detail in a later chapter. There is, however, another version of consideration of the same or similar properties in terms of dissimilarities. The discussion of this section may be more familiar to some readers, since it is related to *ultrametric* [10]. Indeed, ultrametric is the dissimilarity version of fuzzy equivalence, or in other words, fuzzy equivalence is the similarity version of ultrametric.

Let us remind that a dissimilarity measure satisfies the property $D(x, y) = 0$ if $x = y$ and $D(x, y) = D(y, x)$.

Assume, on the other hand, $S(x, y)$ be a given similarity measure and $S(x, x) = 1$ for all $x \in X$. Since $S(x, y) = S(y, x)$, the similarity measure can be regarded as a fuzzy reflexive and symmetric relation. Assume moreover that $S(x, y)$ is transitive, i.e., (1.29) holds for $S(x, y)$.

We now assume that $D(x, y)$ is derived from $D(x, y) = F(S(x, y))$ using (1.21) with a monotone function $F(\cdot)$ with the property (1.19) and (1.20).

What we have from this transformation is a dissimilarity measure that satisfies $D(x, x) = 0$ from the reflexive property of $S(x, x) = 1$, $D(x, y) = D(y, x)$ from the symmetry $S(x, y) = S(y, x)$. Moreover from the transitivity of $S(x, y)$: we have

$$D(x, z) \leq \max\{D(x, y), D(y, z)\}, \quad \forall x, y, z \in X, \tag{1.34}$$

which is known as the *ultrametric inequality* [10].

Using the above transformation, all results concerning fuzzy equivalence relations hold for ultrametrics.

For stating results concerning ultrametrics, we introduce a λ-*cut* for $\lambda \in [0, +\infty)$, i.e., $[D]^{\lambda}$ is a relation:

$$[D]^{\lambda}(x, y) = \begin{cases} 1, & (D(x, y) \leq \lambda), \\ 0, & (D(x, y) > \lambda). \end{cases} \tag{1.35}$$

Using the λ-cut, we have the following.

Theorem 3 *Dissimilarity $D(x, y)$ on X is an ultrametric if and only if its all λ-cuts $[D]^{\lambda}$ (for all $\lambda \in [0, +\infty)$) are equivalence relations. Moreover, for $\lambda < \lambda'$, $[x]_{[D]^{\lambda}} \subseteq [x]_{[D]^{\lambda'}}$ for all $x \in X$.*

The proof is omitted, as the conclusions are easily seen by using the corresponding theorem and the transformation $D(x, y) = F(S(x, y))$.

Thus an ultrametric provides us again a hierarchical clusters $\mathcal{G}(\lambda)$ with the property that for $\lambda < \lambda'$, $\mathcal{G}(\lambda)$ is a refinement of $\mathcal{G}(\lambda')$. The converse is also true, but we omit the details.

Note the difference between those results for a fuzzy equivalence relation and those for ultrametric, and note also that the fundamental parts are same for the both.

References

1. R.R. Sokal, P.H.A. Sneath, *Principles of Numerical Taxonomy* (W.H. Freeman, San Francisco, 1963)
2. M.R. Anderberg, *Cluster Analysis for Applications* (Academic Press, New York, 1973)
3. B. Everitt, *Cluster Analysis*, 2nd edn. (Heinemann, London, 1974)
4. J.B. MacQueen, Some Methods for classification and Analysis of Multivariate Observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1 (Berkeley, University of California Press, 1967), pp. 281–297
5. B.S. Everitt, S. Landau, M. Leese, D. Stahl, *Cluster Analysis*, 5th edn. (Wiley, Chichester, UK, 2011)
6. T. Saito, H. Yadohisa, *Data Analysis of Asymmetric Structures* (Marcel Dekker, New York, 2005)
7. L.A. Zadeh, Similarity relations and fuzzy orderings. Inf. Sci. **3**(2), 177–200 (1971)
8. D. Dubois, H. Prade, *Fuzzy Sets and Systems* (Academic Press, New York, 1988)
9. L.A. Zadeh, Fuzzy sets. Inf. Control **8**, 338–353 (1965)
10. G.W. Milligan, Ultrametric hierarchical clustering algorithms. Psychometrika **44**, 343–346 (1979)

Chapter 2
Linkage Methods and Algorithms

Two basic aspects of agglomerative hierarchical clustering are linkage methods, also called clustering methods, and algorithms for generating clusters. We first describe a formal and abstract procedure of agglomerative hierarchical clustering common to all linkage methods. Second, well-known linkage methods and their properties are discussed. Third, the abstract procedure is made into a concrete algorithm by introducing data records for drawing a *dendrogram*. Finally, theoretical topics are discussed.

2.1 Formal and Abstract Procedure

Let us first review notations in the first chapter and moreover introduce new symbols. The set $X = \{x_1, \ldots, x_N\}$ is the *objects* for clustering. The objects are not necessarily in a real space in general, but for some linkage methods, they need to be points in the Euclidean space. In the latter case, we assume the coordinates are $x_j = (x_j^1, \ldots, x_j^M)$ for M-dimensional real space denoted by $\boldsymbol{R}^M$.

Clusters are denoted by $\mathcal{G} = \{G_1, \ldots, G_K\}$ and hierarchical clusters are with parameter α: $\mathcal{G}(\alpha) = \{G_1, \ldots, G_{K(\alpha)}\}$ as introduced in the first chapter. The parameter α is replaced by an integer K itself which shows the number of clusters in the agglomerative procedure. Given the initial clusters $\mathcal{G}(N_0) = \{G_1, \ldots, G_{N_0}\}$ with $K = N_0$, the procedure finds and merges two clusters, e.g., G_p and G_q into one cluster $G_r = G_p \cup G_q$, and the number of clusters are reduced: $K = K - 1$.

The two clusters to be merged have maximum similarity or minimum dissimilarity among all cluster pairs:

S. Miyamoto, *Theory of Agglomerative Hierarchical Clustering*,
Behaviormetrics: Quantitative Approaches to Human Behavior 15,
https://doi.org/10.1007/978-981-19-0420-2_2

$$S(G_p, G_q) = \max_{i,j(i\neq j)} S(G_i, G_j), \tag{2.1}$$

$$D(G_p, G_q) = \min_{i,j(i\neq j)} D(G_i, G_j). \tag{2.2}$$

Or the same property can be rewritten as

$$(G_p, G_q) = \arg \max_{i,j(i\neq j)} S(G_i, G_j), \tag{2.3}$$

$$(G_p, G_q) = \arg \min_{i,j(i\neq j)} D(G_i, G_j). \tag{2.4}$$

The merging is repeated until there is only one cluster $G_1 = X$ with $K = 1$ when the procedure outputs the process of clustering and stops.

The typical input of the initial clusters is objects themselves: $N_0 = N$ and $G_i = \{x_i\}$ $(i = 1, 2, \ldots, N)$ and we do not consider other cases when $N_0 < N$ until a later chapter, but we do not need to assume $N = N_0$ throughout this book.

The output of the procedure should be a dendrogram that shows the process of merging from the initial N_0 clusters to the final one cluster X. It will be described later and for the moment it is simply represented as a subprocedure **FORM_DENDROGRAM**.

Thus the formal procedure is given in Fig. 2.1 as **AHC** which is the abbreviation of Agglomerative Hierarchical Clustering.

Input: initial clusters $\mathcal{G}(N_0) = \{G_1, \ldots, G_{N_0}\}$
Output: Dendrogram produced in **FORM_DENDROGRAM**
begin AHC
  $K = N_0$
  Find a pair (G_p, G_q) of minimum dissimilarity (or maximum similarity):
    $D(G_p, G_q) = \min_{i,j(i\neq j)} D(G_i, G_j)$
    (or $S(G_p, G_q) = \max_{i,j(i\neq j)} S(G_i, G_j)$)
  % If there are more than one pairs than attain the minimum/maximum,
  % we need a *tie breaking rule*. See below.
    Put $L_K = D(G_p, G_q)$ (or $L_K = S(G_p, G_q)$)
  Merge clusters:
    $G_r = G_p \cup G_q$
  Update $\mathcal{G}(K)$:
    $\mathcal{G}(K-1) = \mathcal{G}(K) \cup \{G_r\} - \{G_p, G_q\}$
    $K = K - 1$
  If $K = 1$ call **FORM_DENDROGRAM** and stop
  call **KEEP_DATA** of $(L_K, \mathcal{G}(K))$
  Update dissimilarity $D(G_r, G_j)$ (or similarity $S(G_r, G_j)$) for all other $G_j \in \mathcal{G}(K)$
  go to **Find a pair**
end AHC.

Fig. 2.1 **AHC**: Formal abstract procedure of agglomerative hierarchical clustering: comment lines begin from %

It seems that subprocedure **KEEP_DATA** needs huge resource to store $\mathcal{G}(K)$. Actually we do not require such huge resource, as we see later in this chapter.

Updating dissimilarity/similarity of $D(G_r, G_j)$ (or $S(G_r, G_j)$) is a crucial part of the procedure. We have different ways for updating called *linkage methods*, which are also called *clustering methods* [1, 2], described in the next section.

Note 2 There may be more than one pair of (G_p, G_q) of minimum dissimilarity (or maximum similarity) in **AHC**. Hence we need a *tie breaking rule* to select one of such pairs. A usual rule is the *lexicographic order* that selects the minimum of (p, q) of those pairs. Note that $p < q$ is assumed, since dissimilarity and similarity measures are symmetric. For example, if $D(G_2, G_5) = D(G_3, G_4)$ and they are the minimum of all dissimilarity values at a particular step of **AHC**, we select (G_2, G_5) for the next merging. This issue of more than one pair of minimum dissimilarity value will be considered again in a later section of this chapter.

2.2 Linkage Methods

Well-known linkage methods are the single linkage, the complete linkage, the average linkage (alias the group average method), the centroid method, and Ward method, although there are other possibilities to choose a non standard method or a novel method. We discuss these well-known methods and their properties in this section. To this end, we first consider similarity or dissimilarity measures between clusters.

2.2.1 Similarity or Dissimilarity Measures Between Clusters

We have various types of similarity measures $S(x, y)$ or dissimilarity measures $D(x, y)$ defined between a pair of objects, as seen in the previous chapter. In the methods of the single linkage, the complete linkage, and the average linkage, we can choose any measure as we consider appropriate. In contrast, the centroid method and Ward method use the squared Euclidean distance.

What we call *dissimilarity/similarity between clusters* or *inter-cluster dissimilarity/similarity* is denoted by $D(G, G')$ (or $S(G, G')$) for $G, G' \in \mathcal{G}$. We give the definition of a dissimilarity/similarity measure between clusters from a dissimilarity/similarity between objects.

We then give, what we call in this book, an *updating formula* that describes $D(G_r, G_j)$ in terms of $D(G_p, G_j)$ and $D(G_q, G_j)$ without referring to $D(x, y)$. When a similarity measure is used, an *updating formula* describes $S(G_r, G_j)$ in terms of $S(G_p, G_j)$ and $S(G_q, G_j)$ without referring to $S(x, y)$.

The abbreviations such as SL, CL, *etc.* after the titles of the linkage methods are used for simplicity, and also we sometimes use them like $D_{SL}(G, G')$, $D_{CL}(G, G')$,

etc. (or $S_{CL}(G, G')$, $S_{CL}(G, G')$, *etc.*) to express that the single linkage, the complete linkage, *etc.*, is used to update the dissimilarity (or similarity).

We describe definitions of inter-cluster dissimilarity and inter-cluster similarity in parallel and then give updating formulas of inter-cluster dissimilarity and inter-cluster similarity for each linkage method.

Single linkage (SL):
Definition of inter-cluster dissimilarity/similarity:

$$D(G, G') = \min_{x \in G, y \in G'} D(x, y) \tag{2.5}$$

$$S(G, G') = \max_{x \in G, y \in G'} S(x, y). \tag{2.6}$$

Updating formula:

$$D(G_r, G_j) = \min\{D(G_p, G_j), D(G_q, G_j)\}, \tag{2.7}$$

$$S(G_r, G_j) = \max\{S(G_p, G_j), S(G_q, G_j)\}, \tag{2.8}$$

where $G_r = G_p \cup G_q$ and G_j is another cluster as noted in **AHC** algorithm.

The updating formulas can be proved using the definitions of inter-cluster dissimilarity/similarity for linkage methods herein; the proofs will be given after this section.

Note 3 A natural question is whether or not an updating formula is needed, after we have the definition of an inter-cluster dissimilarity/similarity for a particular linkage method. This question is discussed in [2]: it is possible to use a linkage method without an updating formula. The calculation of an inter-cluster dissimilarity/similarity using the updating formula is, however, more efficient than the definition itself. Hence an updating formula can be regarded as a way to make calculation more efficiently.

Complete linkage (CL):
Definition of inter-cluster dissimilarity/similarity:

$$D(G, G') = \max_{x \in G, y \in G'} D(x, y) \tag{2.9}$$

$$S(G, G') = \min_{x \in G, y \in G'} S(x, y). \tag{2.10}$$

Updating formula:

$$D(G_r, G_j) = \max\{D(G_p, G_j), D(G_q, G_j)\}, \tag{2.11}$$

$$S(G_r, G_j) = \min\{S(G_p, G_j), S(G_q, G_j)\}. \tag{2.12}$$

Average linkage (alias group average method) (AL):
Definition of inter-cluster dissimilarity/similarity:

$$D(G, G') = \frac{1}{|G||G'|} \sum_{x \in G, y \in G'} D(x, y) \tag{2.13}$$

$$S(G, G') = \frac{1}{|G||G'|} \sum_{x \in G, y \in G'} S(x, y). \tag{2.14}$$

Updating formula:

$$D(G_r, G_j) = \frac{1}{|G_r|}\{|G_p| D(G_p, G_j) + |G_q| D(G_q, G_j)\}, \tag{2.15}$$

$$S(G_r, G_j) = \frac{1}{|G_r|}\{|G_p| S(G_p, G_j) + |G_q| S(G_q, G_j)\}. \tag{2.16}$$

Note that $|G_r| = |G_p| + |G_q|$ holds.

The next two linkage methods assume that an object is a point in an Euclidean space, i.e., $x = (x^1, \ldots, x^M)$. They use the dissimilarity of the squared Euclidean distance.

Centroid method (CNT):
Definition of inter-cluster dissimilarity:
Let the centroid of G be

$$m(G) = \frac{1}{|G|} \sum_{x_k \in G} x_k. \tag{2.17}$$

More specifically, let $G = \{x_{i_1}, \ldots, x_{i_l}\}$ and $x_{i_k} = (x_{i_k}^1, \ldots, x_{i_k}^j, \ldots, x_{i_k}^M)$, then

$$m^j(G) = \frac{1}{|G|} \sum_{k=1}^{l} x_{i_k}^j, \tag{2.18}$$

and $m(G) = (m^1(G), \ldots, m^M(G))$. Then

$$D(G, G') = \|m(G) - m(G')\|^2. \tag{2.19}$$

Similarity $S(G, G')$ cannot be used for this method.

Updating formula:

$$D(G_r, G_j) = \frac{|G_p|}{|G_r|} D(G_p, G_j) + \frac{|G_q|}{|G_r|} D(G_q, G_j) - \frac{|G_p||G_q|}{|G_r|^2} D(G_p, G_q). \tag{2.20}$$

Note again $|G_r| = |G_p| + |G_q|$.

Note moreover that dissimilarity between objects x, y is defined after inter-cluster dissimilarity $D(\{x\}, \{y\})$, i.e.,

$$D(x, y) = D(\{x\}, \{y\}) = \|x - y\|^2. \tag{2.21}$$

Ward method (WRD) [3]:
Definition of inter-cluster dissimilarity:
We first define $E(G)$ as the squared error sum in G with the center $m(G)$:

$$E(G) = \sum_{x_k \in G} \|x_k - m(G)\|^2. \tag{2.22}$$

The increase of the squared error sum $\Delta E(G, G')$ when G and G' are merged is then defined:

$$\Delta E(G, G') = E(G \cup G') - E(G) - E(G'). \tag{2.23}$$

Note that $G \cap G' = \emptyset$ and $\Delta E(G, G') > 0$, since

$$m(G) = \arg \min_{y \in \boldsymbol{R}^M} \sum_{x_k \in G} \|x_k - y\|^2.$$

We now take the dissimilarity in this method to be ΔE:

$$D(G, G') = \Delta E(G, G'). \tag{2.24}$$

Similar to the centroid method, the dissimilarity between two objects is defined in terms of $\Delta E(\{x\}, \{y\})$, i.e.,

$$D(x, y) = \Delta E(\{x\}, \{y\}) = \|x - \frac{x + y}{2}\|^2 + \|y - \frac{x + y}{2}\|^2 = \frac{1}{2}\|x - y\|^2. \tag{2.25}$$

Updating formula:

$$\begin{aligned} D(G_r, G_j) = \frac{1}{|G_r| + |G_j|}\{&(|G_p| + |G_j|)D(G_p, G_j) + (|G_q| + |G_j|)D(G_q, G_j) \\ &- |G_j|D(G_p, G_q)\}, \end{aligned} \tag{2.26}$$

where $|G_r| = |G_p| + |G_q|$.

2.2.2 Proof of Updating Formulas

Proofs of updating formulas given the definitions of the linkage methods in the previous section are described below.

Single linkage:
It is enough to note the following:

$$\begin{aligned} D(G_r, G_j) &= \min_{x \in G_r, y \in G_j} D(x, y) \\ &= \min\{\min_{x \in G_p, y \in G_j} D(x, y), \min_{x \in G_q, y \in G_j} D(x, y)\} \\ &= \min\{D(G_p, G_j), D(G_q, G_j)\}. \end{aligned}$$

For $S(G_r, G_j)$, the above equation can be rewritten by replacing min by max.

Complete linkage:
It is enough to note the following:

$$\begin{aligned} D(G_r, G_j) &= \max_{x \in G_r, y \in G_j} D(x, y) \\ &= \max\{\max_{x \in G_p, y \in G_j} D(x, y), \max_{x \in G_q, y \in G_j} D(x, y)\} \\ &= \max\{D(G_p, G_j), D(G_q, G_j)\}. \end{aligned}$$

For $S(G_r, G_j)$, the above equation can be rewritten by replacing max by min.

Average linkage (alias group average method):
Note

$$\begin{aligned} |G_r| D(G_r, G_j) &= \sum_{x \in G_r, y \in G_j} D(x, y) \\ &= \sum_{x \in G_p, y \in G_j} D(x, y) + \sum_{x \in G_q, y \in G_j} D(x, y) \\ &= |G_p| D(G_p, G_j) + |G_q| D(G_q, G_j), \end{aligned}$$

whence the updating formula follows. For $S(G_r, G_j)$ it is enough to replace D by S in the above equations.

Centroid method:
Let us introduce a symbol of inner product:

$$\langle x, y \rangle = x^1 y^1 + \cdots + x^M y^M \tag{2.27}$$

and note

$$2\langle x, y \rangle = \|x\|^2 + \|y\|^2 - \|x - y\|^2 \tag{2.28}$$

Moreover note the following:

$$m(G_r) = \frac{|G_p|}{|G_r|} m(G_p) + \frac{|G_q|}{|G_r|} m(G_q). \tag{2.29}$$

Put $\alpha_p = \frac{|G_p|}{|G_r|}$ and $\alpha_q = \frac{|G_q|}{|G_r|}$, we then have $\alpha_p + \alpha_q = 1$. Now, it is easy to see the following:

$$\begin{aligned}
D(G_r, G_j) &= \|\alpha_p(m(G_p) - m(G_j)) + \alpha_q(m(G_q) - m(G_j))\|^2 \\
&= \alpha_p^2 \|m(G_p) - m(G_j)\|^2 + \alpha_q^2 \|m(G_q) - m(G_j)\|^2 \\
&\quad + 2\alpha_p\alpha_q \langle m(G_p) - m(G_j), m(G_q) - m(G_j)\rangle \\
&= \alpha_p^2 \|m(G_p) - m(G_j)\|^2 + \alpha_q^2 \|m(G_q) - m(G_j)\|^2 \\
&\quad + \alpha_p\alpha_q \|m(G_p) - m(G_j)\|^2 + \alpha_p\alpha_q \|m(G_q) - m(G_j)\|^2 \\
&\quad - \alpha_p\alpha_q \|m(G_p) - m(G_q)\|^2 \\
&= \alpha_p \|m(G_p) - m(G_j)\|^2 + \alpha_q \|m(G_q) - m(G_j)\|^2 \\
&\quad - \alpha_p\alpha_q \|m(G_p) - m(G_q)\|^2.
\end{aligned}$$

Thus the updating Eq. (2.20) is proved.

Ward method:
Let G and G' be two disjoint subsets of X: $G \cap G' = \emptyset$. Note the well-known inequality:

$$\sum_{x_k \in G} \|x_k - m(G)\|^2 = \sum_{x_k \in G} \|x_k\|^2 - |G| \|m(G)\|^2.$$

Using the symbol $\Delta E(G, G')$ in this proof and noting $|G \cup G'| = |G| + |G'|$, we have

$$\begin{aligned}
\Delta E(G, G') &= E(G \cup G') - E(G) - E(G') \\
&= \sum_{x_k \in G \cup G'} \|x_k\|^2 - |G \cup G'| \|m(G \cup G')\|^2 \\
&\quad - \Big(\sum_{x_k \in G} \|x_k\|^2 - |G| \|m(G)\|^2\Big) - \Big(\sum_{x_k \in G'} \|x_k\|^2 - |G'| \|m(G')\|^2\Big) \\
&= |G| \|m(G)\|^2 + |G'| \|m(G')\|^2 - |G \cup G'| \|m(G \cup G')\|^2.
\end{aligned}$$

Substituting

$$|G \cup G'| \|m(G \cup G')\|^2 = (|G| + |G'|) \Big\| \frac{|G| m(G) + |G'| m(G')}{|G| + |G'|} \Big\|^2$$

into the above equation of $\Delta E(G, G')$, we have

$$\begin{aligned}\Delta E(G, G') &= |G|\|m(G)\|^2 + |G'|\|m(G')\|^2 \\ &\quad - \frac{|G|^2\|m(G)\|^2 + |G'|^2\|m(G')\|^2 + 2|G||G'|\langle m(G), m(G')\rangle}{|G| + |G'|} \\ &= \frac{|G||G'|}{|G| + |G'|}\|m(G) - m(G')\|^2. \end{aligned} \tag{2.30}$$

We now use the updating formula of the centroid method, i.e., substituting

$$\frac{|G| + |G'|}{|G||G'|}\Delta E(G, G') = \|m(G) - m(G')\|^2$$

into (2.20) and noting $|G_r| = |G_p| + |G_q|$, we have

$$\begin{aligned}\Delta E(G_r, G_j) &= \frac{|G_r||G_j|}{|G_r| + |G_j|}\|m(G_r) - m(G_j)\|^2 \\ &= \frac{|G_r||G_j|}{|G_r| + |G_j|}\{\frac{|G_p|}{|G_r|}\|m(G_p) - m(G_j)\|^2 + \frac{|G_q|}{|G_r|}\|m(G_q) - m(G_j)\|^2 \\ &\quad - \frac{|G_p||G_q|}{|G_r|^2}\|m(G_p) - m(G_q)\|^2\} \\ &= \frac{|G_r||G_j|}{|G_r| + |G_j|}\{\frac{|G_p|}{|G_r|}\frac{|G_p| + |G_j|}{|G_p||G_j|}\Delta E(G_p, G_j) \\ &\quad + \frac{|G_q|}{|G_r|}\frac{|G_q| + |G_j|}{|G_q||G_j|}\Delta E(G_q, G_j) - \frac{|G_p||G_q|}{|G_r|^2}\frac{|G_p| + |G_q|}{|G_p||G_q|}\Delta E(G_p, G_q)\} \\ &= \frac{1}{|G_r| + |G_j|}\{(|G_p| + |G_j|)\Delta E(G_p, G_j) + (|G_q| + |G_j|)\Delta E(G_q, G_j) \\ &\quad - |G_j|\Delta E(G_p, G_q)\}. \end{aligned}$$

The updating formula (2.26) is thus proved.

2.3 Examples

We observe the results of clustering for the points in Fig. 1.1 using other linkage methods than those in Sect. 1.5.

Before describing these examples, let us briefly review the two examples in Chap. 1. The single linkage uses the *nearest neighbor rule*, which is formulated by (2.5) and (2.7). The centroid method apparently uses the rule of (2.19) or (2.20), and the results of the dendrograms are respectively shown in Figs. 1.3 and 1.5.

We proceed to observe the results of the complete linkage, the average linkage, and Ward method.

Example 4 We apply the complete linkage method to the example in Fig. 1.1 where the dissimilarity is the Euclidean distance: $D(x, y) = \|x - y\|$. The updating rule is

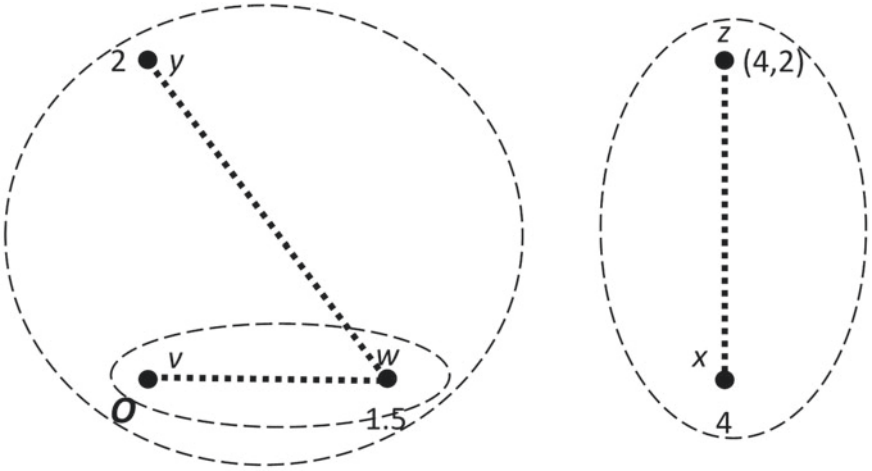

Fig. 2.2 Clusters obtained from the points in Fig. 1.1 using the complete linkage

(2.9) or (2.11). The initial clusters are each object: $\{v\}$, $\{w\}$, $\{x\}$, $\{y\}$, $\{z\}$. The first cluster is again $\{v, w\}$ as shown by an oval in Fig. 2.2. Distances between $\{v, w\}$ and other three objects are calculated as follows:

$$D(\{v, w\}, x) = \max\{D(v, x), D(w, x)\} = D(v, x) = 4,$$
$$D(\{v, w\}, y) = \max\{D(v, y), D(w, y)\} = D(w, y) = \sqrt{2^2 + 1.5^2},$$
$$D(\{v, w\}, z) = \max\{D(v, z), D(w, z)\} = D(v, z) = \sqrt{4^2 + 2^2}.$$

The second cluster is hence $\{x, z\}$.We then calculate the new distances:

$$D(\{v, w\}, \{x, z\}) = D(v, z) = \sqrt{4^2 + 2^2},$$
$$D(y, \{x, z\}) = D(y, x) = \sqrt{4^2 + 2^2}.$$

The third cluster is $\{v, w, y\}$ which is formed at the level $\sqrt{2^2 + 1.5^2} = 2.5$. Finally, the two clusters $\{v, w, y\}$ and $\{x, z\}$ are merged at $\sqrt{4^2 + 2^2} \approx 4.47$ into the whole set of one cluster. The clusters are shown in Fig. 2.2 as ovals except the final cluster of the five points. Dotted lines show distances between clusters.

The list of hierarchical clusters is shown in Table 2.1 and the corresponding dendrogram for this example is shown in Fig. 2.3. The interpretation of this dendrogram is easy and omitted.

Example 5 We see the result from the average linkage applied to the same example using the Euclidean distance. The updating rule is (2.13) or (2.15).

The first cluster is again $\{v, w\}$ shown by an oval in Fig. 2.4. Distances between $\{v, w\}$ and other three objects are calculated as follows:

Table 2.1 List of clusters obtained from the points in Fig. 1.1 using the complete linkage

Clusters	Level whereby two clusters are merged
$\{v\}, \{w\}, \{x\}, \{y\}, \{z\}$	–
$\{v, w\}, \{x\}, \{y\}, \{z\}$	1.5
$\{v, w\}, \{y\}, \{x, z\}$	2
$\{v, w, y\}, \{x, z\}$	2.5
$\{v, w, x, y, z\}$	4.47

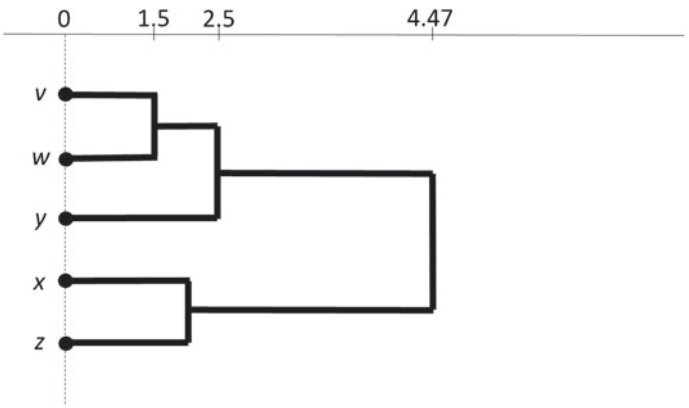

Fig. 2.3 Dendrogram obtained from the points in Fig. 1.1 using the complete linkage

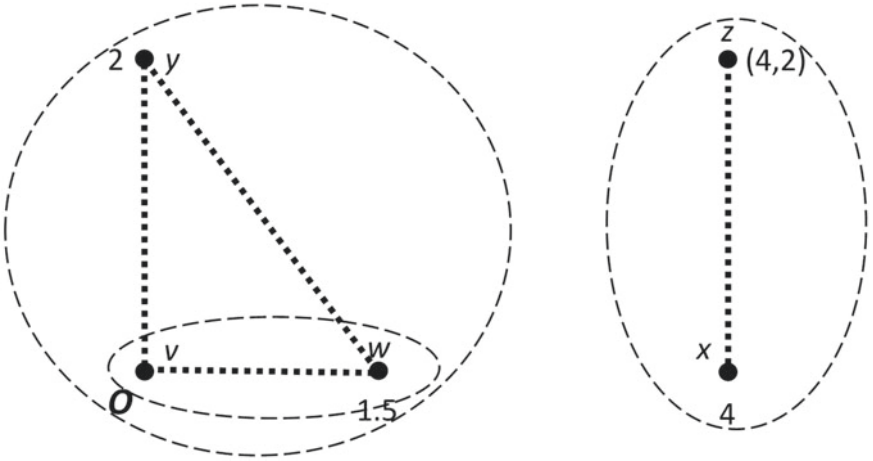

Fig. 2.4 Clusters obtained from the points in Fig. 1.1 using the average linkage

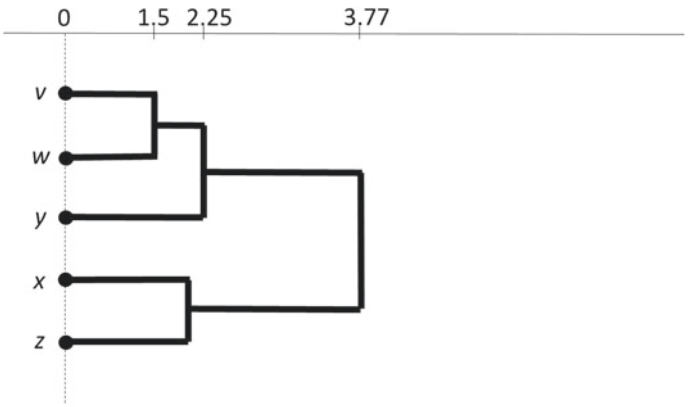

Fig. 2.5 Dendrogram obtained from the points in Fig. 1.1 using the average linkage

$$D(\{v, w\}, x) = \frac{1}{2}(D(v, x) + D(w, x)) = \frac{1}{2}(2.5 + 4),$$
$$D(\{v, w\}, y) = \frac{1}{2}(D(v, y) + D(w, y)) = \frac{1}{2}(2 + 2.5),$$
$$D(\{v, w\}, z) = \frac{1}{2}(D(v, z) + D(w, z)) = \frac{1}{2}(\sqrt{4^2 + 2^2} + \sqrt{2.5^2 + 2^2}).$$

The second cluster is $\{x, z\}$.We then update the distances:

$$D(\{v, w\}, \{x, z\}) = \frac{1}{4}(D(v, x) + D(w, x) + D(v, z) + D(w, z)),$$
$$D(y, \{x, z\}) = \frac{1}{2}(D(y, x) + D(y, z)).$$

The third cluster is $\{v, w, y\}$ which is formed at the level $\frac{1}{2}(2 + 2.5) = 2.25$. Finally, the two clusters $\{v, w, y\}$ and $\{x, z\}$ are merged at 3.77 into the whole set of one cluster. The clusters are shown in Fig. 2.4 as ovals except the final cluster of the five points. Dotted lines in this case imply distances used in calculating averages.

The list of clusters is omitted and the dendrogram for this example is shown in Fig. 2.5. The interpretation of this dendrogram is almost trivial and omitted.

The last example for the same set of points uses Ward method which uses the squared Euclidean distance and the centroids, but still different from the centroid method. The formula (2.30) shows the relation and difference of these two linkage methods. The detailed description for Ward method is omitted and the dendrogram is shown in Fig. 2.6. We note the first cluster is formed at

$$D(v, w) = \frac{1}{2}\|v - w\|^2 = 1.125$$

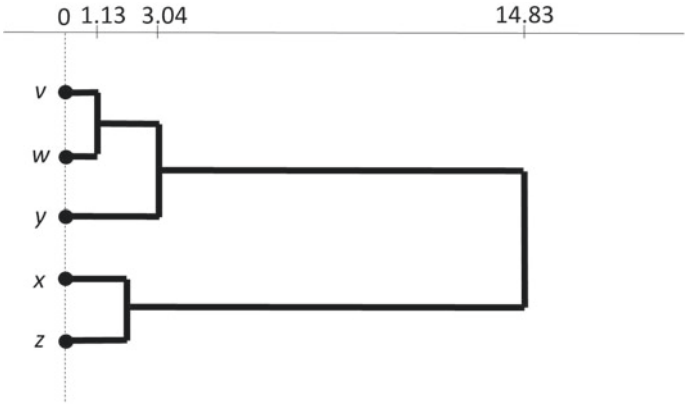

Fig. 2.6 Dendrogram obtained from the points in Fig. 1.1 using ward method

and the final cluster is formed at

$$D(\{v, w, y\}, \{x, z\}) = \frac{6}{5}\|m(\{v, w, y\}) - m(\{x, z\})\|^2 \approx 14.83.$$

2.4 Output of Dendrogram

Usually how a dendrogram is printed is not described in books of clustering. The reason why we discuss this subject is that the dendrogram is essential in agglomerative hierarchical clustering regardless whether one is interested in this or not.

The description is divided into two parts. In the first part we observe a tree as a basic and abstract data structure for a dendrogram. In the second part to actually draw a dendrogram using a *tree traversal* technique is discussed. The second part is more technical and some readers can skip this part. In both parts we use the simple example of the five points in Fig. 1.1.

2.4.1 Tree as Data Structure for Dendrogram

The *tree* used here to represent dendrogram is precisely a *rooted binary tree* that is a special type of directed graph. A *directed graph* is a pair (V, E) in which $V = \{v_1, \ldots, v_n\}$ is a finite set of *nodes*, and $E = ((v_i, v_j), \ldots, (v_h, v_l))$ is a set of directed pairs. Thus, $E \subseteq V \times V$. An element (v_i, v_j) is called here a link from v_i to v_j and represented as an arrow $v_i \to v_j$, incoming to v_j and outgoing of v_i.

The definition of a *rooted binary tree* is given below.

A *rooted binary tree* $T = (V, E)$ is a directed graph that satisfies the next two properties:

(i) There exists one and only one node v_0, called the root node, having no incoming arrow: there is no $v \in V$ such that $(v, v_0) \in E$. The root node has just two outgoing arrows.
(ii) Other nodes $v \in V$ $(v \neq v_0)$ has one incoming arrow. They are divided into two types:

(ii-1) A leaf node (or simply leaf) has no outgoing arrow.
(ii-2) A non leaf node has just two outgoing arrows.

Figure 2.7 is an example of a rooted binary tree in which nodes are represented by circles and numbers. The root is node 9 and the leaves are $1 - 5$. The other nodes $6 - 8$ are non leaves.

A tree is often with associated data, an example of which is shown in Fig. 2.8. A node i is thus with the data of (label, level). Thus node 1 is with label $= v$ and level $= 0$; node 6 is with label $= -$ and level $= 1.5$. The symbol '$-$' implies no value (alias null value).

It is now clear that this figure is an abstracted form of the dendrogram in Fig. 1.3 of the single linkage applied to the points in Fig. 1.1.

To represent a link in a rooted tree, we introduce two data items of left_child and right_child at each node.

Thus the data records for the tree in Fig. 2.8 is in Table 2.2. For example, record of node 6 has left_child of node 1 and right_child of node 2. When two clusters are merged, we make a new node and the merged clusters are made to be the two children of the new node that represents the merged cluster. In this example, cluster $\{v, w\}$ is node 6: $\{v\}$ is left_child and $\{w\}$ is right_child. Then $\{v, w, y\}$ is node 7: $\{v, w\}$ is left_child and $\{y\}$ is right_child. After $\{x, z\}$ as node 8 is made with $\{x\}$ as left_child

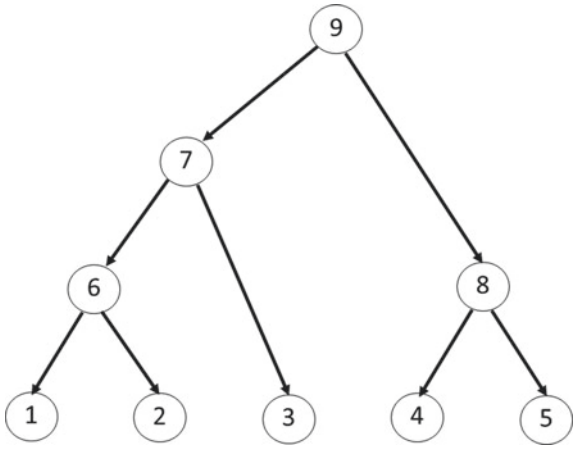

Fig. 2.7 An example of a rooted binary tree

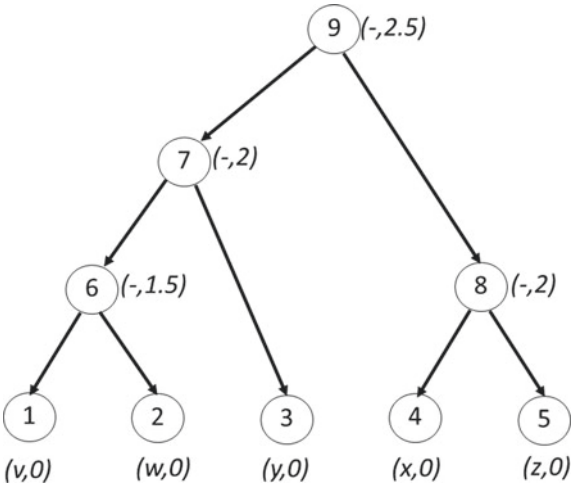

Fig. 2.8 A rooted tree with labels and levels

Table 2.2 Data records for the tree in Fig. 2.8. Symbol '–' means null value

	Id	Label	Level	Left_child	Right_child	
[	1:	v	0	–	–	]
[	2:	w	0	–	–	]
[	3:	y	0	–	–	]
[	4:	x	0	–	–	]
[	5:	z	0	–	–	]
[	6:	–	1.5	1	2	]
[	7:	–	2	6	3	]
[	8:	–	2	4	5	]
[	9:	–	2.5	7	8	]

and $\{z\}$ as right_child, the final cluster is made as node 9 . The levels are used when actual dendrogram is drawn.

Thus the subprocedure to keep data needed for drawing dendrograms is as follows.

KEEP_DATA for $(L_K, \mathcal{G}(K))$
% COMMENT:
% uses records of [id , label , level , left_child , right_child] with initial values
% and set all labels of objects x_i with id $= i$ and level $= 0$
% before merging clusters starts
% cluster_ list of initial value of $\{1, 2, \ldots, N\}$ is also used.
% Let initial idcount= N.
Find $P = \mathrm{id}(p)$ and $Q = \mathrm{id}(q)$ such that
$D(G_p, G_q) = \min_{i,j\,(i \neq j)} D(G_i, G_j)$
idcount=idcount+1
make new_record with id=idcount
put level= $D(G_p, G_q)$, left_child=P , right_child=Q of new_record
delete P and Q from cluster_list, and add idcount to cluster_list
end KEEP_DATA

This procedure requires the updated dissimilarity $D(G_r, G_j)$ that should be stored into an array of

$$D(\text{idcount, j}) \quad \text{for all j in cluster\_list}$$

where 'idcount' represents G_r.

Note 4 In the algorithms herein, we do not strictly distinguish *left* and *right* of the two children of the rooted tree. For example, we can put left_child= node 1 or right_child = node 1 as well. Readers who wish to actually develop computer programs have to determine which is left or right. The author recommends to try lexicographic order.

2.4.2 Drawing Dendrograms

The final step to realize the algorithm of agglomerative hierarchical clustering is to draw dendrograms. For this purpose what we need is to specify the positions of nodes in the rooted tree. The position of a node is specified by two coordinates of which one is the level of merging, which is also called *height*. The other coordinate is called here x_position and hence the position is specified by (x_position, level). Thus a record for a node has the fields of

$$[\text{id, x\_position, label, level, left\_child, right\_child}]$$

which has the field of x_position in addition to those in Table 2.2.

We proceed to describe the way to determine x_position. What is essential is

do not place the objects so that lines of the dendrogram cross.

For example, the dendrogram in Fig. 2.2 and also in other figures of those dendrograms of the same example has the order of v, w, y, x, z. If we change the order into

the alphabetical order of v, w, x, y, z, some lines of those dendrograms will cross without altering clustering results. If we have many more objects for clustering, such crossing of lines will produce terribly entangled output.

The solution is obtained by applying a basic technique of *recursive programming*.

```
FORM_DENDROGRAM
% This procedure simply calls a recursive procedure
% of OUTPUT_DENDROGRAM(·)
% which will be described next.
  % Set initial value of x_position
  x_position = 0
call OUTPUT_DENDROGRAM(idcount)
  % note idcount refers to the last record
  % of the root of the tree
  % which is set in KEEP_DATA
end FORM_DENDROGRAM
```

```
OUTPUT_DENDROGRAM(id)
% The recursive procedure to output the dendrogram
% If there is no child, simply output the label
% at (x_position, level)
  if left_child = NULL_VALUE then
    x_position = x_position + x_increment
    PRINT LABEL(label) at (x_position, level)
  % We do not specify the detail of PRINT LABEL(label)
  % as that is almost trivial
  else
    call OUTPUT_DENDROGRAM(left_child(id))
    call OUTPUT_DENDROGRAM(right_child(id))
  endif
  % Set x_position of 'id' as the mid point of those children
  x_position = 0.5*(x_position(left_child)+ x_position(right_child) )
  call DRAW_LINES that connects those two children and the node of id
  % details of DRAW_LINES is omitted as it is easy
end OUTPUT_DENDROGRAM
```

As stated in the last procedure, a node is placed in the mid of those nodes of the two children, whereby no cross of lines occurs.

Let us apply this procedure to the example of five points in Fig. 1.1.

Example 6 Let us consider the single linkage: the records are shown in Table 2.2 except the item of x_position.

- **FORM_DENDROGRAM** calls **OUTPUT_DENDROGRAM**(idcount) where idcount = 9,
- which first calls **OUTPUT_DENDROGRAM**(7) with left_child = 7,

Table 2.3 The data in Table 2.2 with the additional field of x_position, where x_position = 1. Symbol '−' means null value

	Id	X_position	Label	Level	Left_child	Right_child	
[	1:	1	v	0	−	−	]
[	2:	2	w	0	−	−	]
[	3:	3	y	0	−	−	]
[	4:	4	x	0	−	−	]
[	5:	5	z	0	−	−	]
[	6:	1.5	−	1.5	1	2	]
[	7:	2.25	−	2	6	3	]
[	8:	4.5	−	2	4	5	]
[	9:	3.375	−	2.5	7	8	]

- then **OUTPUT_DENDROGRAM**(6) with left_child = 6 is called.
- When **OUTPUT_DENDROGRAM**(1) with left_child = 1 is called as the next step, x_position of 1 is determined as x_position = x_increment, since its left_child = − (NULL_VALUE). If we assume x_increment=1, then x_position = 1 for object label 1.
- Object x_position with label 2 is then determined as x_position = 2 ∗ *x*_increment (= 2), as it is right_child of object 6.
- Next, x_position with node label 6 is determined: x_position = 0.5∗(x_increment +2∗ x_increment) = 1.5∗x_increment.

In this way, the positions of nodes are determined in the following order:

$$\begin{aligned}
\text{object label 3} &= 3 * \text{ x_increment},\\
\text{node label 7} &= 2.25 * \text{ x_increment},\\
\text{object label 4} &= 4 * \text{ x_increment},\\
\text{object label 5} &= 5 * \text{ x_increment},\\
\text{node label 8} &= 4.5 * \text{ x_increment},\\
\text{node label 9} &= 3.375 * \text{ x_increment}
\end{aligned}$$

The table after the node positions are determined is shown in Table 2.3. Note again that the order of referring to the node labels is $\rightarrow 1 \rightarrow 2 \rightarrow 6 \rightarrow 3 \rightarrow 7 \rightarrow 4 \rightarrow 5 \rightarrow 8 \rightarrow 9$.

2.5 Problems and Theoretical Issues

In order to start methodological considerations of agglomerative hierarchical clustering, we should discuss some typical and inherent problems in this method. We use a small example in Fig. 2.9 for the discussion in this section. This figure shows four

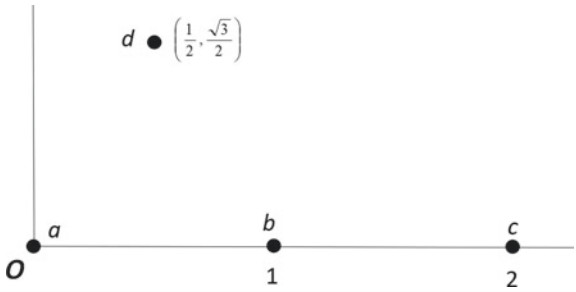

Fig. 2.9 Four points on the plane. Points a, b, c are on the horizontal axis with the equal interval. Points a, b, d forms an equilateral triangle

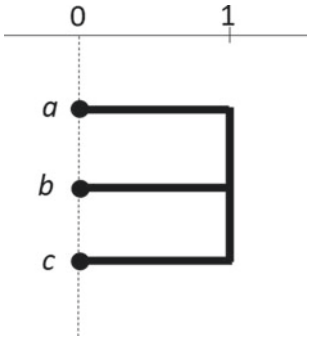

Fig. 2.10 Dendrogram of three points $\{a, b, c\}$ in Fig. 2.9 using the single linkage

points a, b, c, d on the plane. Points a, b, c are on the horizontal axis with the equal interval of $[0, 1]$ and $[1, 2]$. Points a, b, d form an equilateral triangle.

Example 7 We first consider the problem of more than one pairs of the minimum dissimilarity, i.e., a tie breaking rule. Let us consider the set of three points $\{a, b, c\}$ with the Euclidean distance in Fig. 2.9. As the linkage methods, we consider the single linkage (SL), complete linkage (CL), and average linkage (AL). Applying **AHC** algorithm, we find two pairs $\{a, b\}$ and $\{b, c\}$ as the minimizing elements, since

$$D(a, b) = D(b, c) = 1 \text{ and } D(a, c) = 2.$$

According to the lexicographic order rule, we select $\{a, b\}$ as the pair for the first merging.

Let us denote the distance between the merged group $\{a, b\}$ and $\{c\}$ using the three linkage methods by $D_{SL}(\{a, b\}, \{c\})$, $D_{CL}(\{a, b\}, \{c\})$, and $D_{AL}(\{a, b\}, \{c\})$, respectively. We then have

$$D_{SL}(\{a, b\}, \{c\}) = 1, \quad D_{CL}(\{a, b\}, \{c\}) = 2, \quad D_{AL}(\{a, b\}, \{c\}) = 1.5.$$

We thus have the three different dendrograms shown in Figs. 2.10, 2.11, 2.12.

As these figures show, the results from the complete and average linkages seem strange, since if we change the rule of selection from $\{a, b\}$ to $\{b, c\}$, then the results

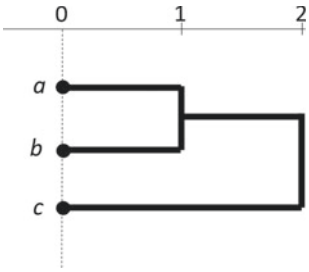

Fig. 2.11 Dendrogram of three points $\{a, b, c\}$ in Fig. 2.9 using the complete linkage

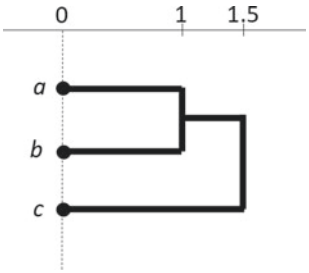

Fig. 2.12 Dendrogram of three points $\{a, b, c\}$ in Fig. 2.9 using the average linkage

change. In contrast to these two, the result from the single linkage does not change even if we select another pair. The same situations also occur for the centroid method and Ward method. Thus the results of the linkage methods are order-dependent except for the single linkage.

Such a problem cannot be avoided, since the method of agglomerative hierarchical clustering is defined by the algorithm. Nevertheless, we will prove that the single linkage method is order-independent: even when we change the labeling order, the results remain the same.

This observation (together with theoretical results in the next chapter) means that the single linkage method is theoretically better than other linkage methods, contrary to the general understanding that other methods are preferred than the single linkage in applications.

Example 8 We next consider another set of three points $\{a, b, d\}$ which forms an equilateral triangle:

$$D(a, b) = D(b, c) = D(a, c) = 1.$$

The dendrograms of the three linkage methods (SL, CL, and AL) are the same and shown in Fig. 2.13.

Example 9 Consider the same set of points $\{a, b, d\}$ and this time we use the centroid method. After a and b are merged at the level $D(a, b) = 1$, we have $D(\{a, b\}, \{c\}) = 3/4 < D(a, b)$. We thus have a reversed dendrogram shown in Fig. 2.14. Such a reversal is known already in old literature (see, e.g., [2]).

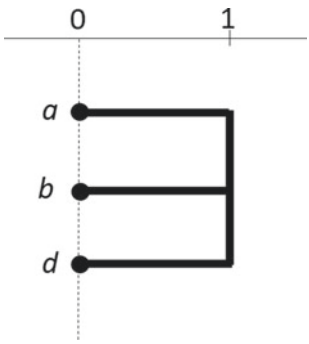

Fig. 2.13 Dendrogram of three points $\{a, b, d\}$ in Fig. 2.9 using the single linkage, complete linkage, or average linkage

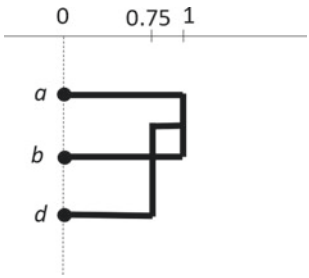

Fig. 2.14 Dendrogram of three points $\{a, b, d\}$ in Fig. 2.9 using the centroid method

2.5.1 Reversals: Monotone Property of Merging Levels

The last example can theoretically be discussed. Indeed, we prove the following property in this section. Readers who are not interested in the details of the proof can skip the descriptions until the next section.

Theorem 4 *The single linkage, complete linkage, average linkage, and Ward method do not have any reversal (reversed dendrogram) for any given set of objects and dissimilarity/similarity values.*

The rest of this section is devoted to definitions to precisely describe reversals and the proofs of the above property.

We define the set of the values of a dissimilarity measure $\mathcal{D}$ given X and the initial dissimilarity $D(x, y)$.

Definition 3 The set of the values of a dissimilarity measure given the initial dissimilarity $D(x, y)$ and the object set X using the linkage method Link when the number of clusters is K is denoted by $\mathcal{D}(X, D, K, \text{Link})$. Moreover the minimum element of $\mathcal{D}(X, D, K, \text{Link})$ is denoted by

$$\min \mathcal{D}(X, D, K, \text{Link}). \tag{2.31}$$

Here Link is one of SL, CL, AL, CNT, and WRD which means the five linkage methods.

Note 5 We emphasize that $\mathcal{D}(X, D, K, \text{Link})$ is the set of values of *inter-cluster* dissimilarity.

We have the next theorem:

Theorem 5 *If* Link *is one of* SL, CL, AL, *and* WRD, *we have*

$$\min \mathcal{D}(X, D, K-1, \text{Link}) \geq \min \mathcal{D}(X, D, K, \text{Link}), \quad \text{for } 2 \leq K \leq N, \tag{2.32}$$

for any given set of objects and dissimilarity measures. Note that the dissimilarity measure when Ward method is used is limited to $D(G, G') = \Delta E(G, G')$ *derived from the squared Euclidean distance.*

The similarity version of the above definition and the theorem are as follows.

Definition 4 The set of the values of a similarity measure given the initial similarity $S(x, y)$ and the object set X using the linkage method Link when the number of clusters is K is denoted by $\mathcal{S}(X, S, K, \text{Link})$. Moreover the maximum element of $\mathcal{S}(X, S, K, \text{Link})$ is denoted by

$$\max \mathcal{S}(X, S, K, \text{Link}). \tag{2.33}$$

Here Link is one of SL, CL and AL, which means the three linkage methods.

Theorem 6 *If* Link *is one of* SL, CL, *and* AL, *we have*

$$\max \mathcal{S}(X, S, K-1, \text{Link}) \leq \max \mathcal{S}(X, S, K, \text{Link}), \quad \text{for } 2 \leq K \leq N, \tag{2.34}$$

for any given set of objects and similarity measures.

Let us note that (2.32) means that the levels of merging in **AHC** algorithm is monotone increasing:

$$L_N \leq L_{N-1} \leq \cdots \leq L_2 \leq L_1. \tag{2.35}$$

In the case of similarity:

$$L_N \geq L_{N-1} \geq \cdots \geq L_2 \geq L_1. \tag{2.36}$$

The last two relations clearly imply that the dendrograms have no reversals. In summary, the monotone property (2.32) and (2.34) is equivalent to the non existence of a reversed dendrogram.

We proceed to prove the above theorem. Assume that a dissimilarity measure is used. It is easier to see that SL, CL, and AL satisfy (2.32), since it is easily seen that updated dissimilarity satisfy

$$D(G_r, G_j) \geq \min \mathcal{D}(X, D, K, \text{Link}), \tag{2.37}$$

from (2.7), (2.11), and (2.15). The last relation immediately leads to (2.32). The proof for a similarity measure with SL, CL, and AL is the same as above and is omitted.

We next prove the theorem for WRD using (2.26). Let

$$M = \min \mathcal{D}(X, D, K, \text{Link})$$

for simplicity. Since $D(D_p, D_j) \geq M$, $D(D_q, D_j) \geq M$, and $D(D_p, D_q) = M$ hold in (2.26), we have

$$\begin{aligned} D(G_r, G_j) &= \frac{1}{|G_r| + |G_j|}\{(|G_p| + |G_j|)D(G_p, G_j) + (|G_q| + |G_j|)D(G_q, G_j) \\ &\quad - |G_j|D(G_p, G_q)\} \\ &\geq \frac{1}{|G_r| + |G_j|}\{(|G_p| + |G_j|)M + (|G_q| + |G_j|)M - |G_j|M\} = M, \end{aligned}$$

where $|G_r| = |G_p| + |G_q|$. The last equation clearly implies (2.32). Thus the theorem is proved.

Note again that the previous example of a reversed dendrogram shows that the centroid method does not satisfy the monotone property in general.

2.5.2 Similar Dendrograms

Let us consider *similarity* between two dendrograms. We mean abstract dendrogram that can be described by **AHC** algorithm.

Definition 5 A pair of dendrograms for two hierarchical clusters $\mathcal{G}$ and $\mathcal{G}'$ is called **similar** if their hierarchical clusters are equivalent:

$$\mathcal{G}(k) = \mathcal{G}'(k), \quad \text{for } k = N, N-1, \ldots, 2, 1, \tag{2.38}$$

Note that the same tie breaking rule is applied in the two clusters $\mathcal{G}$ and $\mathcal{G}'$. Also, we use $\mathcal{G}$ and $\mathcal{G}'$ to denote the whole clusters: $\mathcal{G} = (\mathcal{G}(N), \mathcal{G}(N-1), \ldots, \mathcal{G}(1))$, $\mathcal{G}' = (\mathcal{G}'(N), \mathcal{G}'(N-1), \ldots, \mathcal{G}'(1))$, by abuse of terminology for simplicity's sake.

In this sense, all dendrograms of the simple example of five objects in the first chapter are similar.

Consideration of similarity in dendrograms is useful in a later chapter. In this section, however, we discuss a rather trivial example of the similarity.

Monotone Functions

A strictly monotone function f on the non negative real numbers $[0, +\infty)$ is defined as follows.

Definition 6 Assume that f is defined on $[0, +\infty)$ with its values in $[0, +\infty)$: $(f : [0, +\infty) \to [0, +\infty))$. It is called strictly monotone increasing if

$$f(x) < f(y), \quad \forall\, x, y \text{ such that } x < y. \tag{2.39}$$

For simplicity, assume that D is a matrix of dissimilarity: $D = (D(x_i, x_j))$. Let us moreover define $f(D)$ by

$$f(D) = (f(D(x_i, x_j))), \tag{2.40}$$

i.e., $f(D)$ is the matrix whose components are $f(D(x_i, x_j))$.

We now have immediate conclusions:

Theorem 7 *We apply the single linkage to the same set of objects X but with different dissimilarity measures of D and $f(D)$. Let $\mathcal{G}$ and $\mathcal{G}'$ be the two produced hierarchical clusters. Then the two dendrograms are similar. Moreover when we use the complete linkage method instead of the single linkage, we have the same conclusion: the two dendrograms are similar.*

Since a dendrogram is abstractly defined, we can express the similarity as

$$\mathcal{G} \sim \mathcal{G}'. \tag{2.41}$$

using $\mathcal{G}$ and $\mathcal{G}'$ as the whole structure of hierarchical clusters.

The above theorem can easily be extended to a monotone decreasing function transforming a dissimilarity measure into a similarity measure and *vice versa*.

Let be g be a strictly monotone decreasing function defined on $[0, +\infty)$ with its values in $[0, +\infty)$:

$$g(x) > g(y), \quad \forall\, x, y \text{ such that } x < y. \tag{2.42}$$

Assume SL or CL is used. If two clusters produced from D and $g(D)$ on the same set of objects with different dissimilarity D and similarity $g(D)$ are denoted by $\mathcal{G}$ and $\mathcal{G}'$, we have $\mathcal{G} \sim \mathcal{G}'$, i.e., two dendrograms are similar.

The proofs of these properties are easy and omitted. Note that only minima and maxima are used in the calculations of SL and CL.

References

1. B. Everitt, *Cluster Analysis*, 2nd edn. (Heinemann, London, 1974)
2. M.R. Anderberg, *Cluster Analysis for Applications* (Academic Press, New York, 1973)
3. J.H. Ward Jr., Hierarchical grouping to optimize an objective function. Am. Stat. Assoc. J. **58**, 236–244 (1963)

Chapter 3
Theory of the Single Linkage Method

A well-known but less popular in applications than some other method (like Ward method) is the single linkage method (abbreviated SL). As stated in the previous section, this method has sound theoretical properties when compared with other linkage methods. This section describes the theoretical properties of this method. In particular, we will prove a fundamental result that states the following equivalence of clusters derived by four methods [1]:

- the single linkage,
- the minimum spanning tree algorithm,
- the connected components of a fuzzy graph, and
- the transitive closure of a fuzzy relation.

Readers may notice that 'fuzzy' concepts are seen in the above. Indeed, many studies related to the single linkage have been done in the field of fuzzy sets [2]. The details of fuzzy sets and systems are unnecessary and the minimum requirements that are directly related to the above result will be described below.

Before describing the details of the theory in this chapter, we note that the distinction between dissimilarity and similarity is not essential in the theory of the single linkage, since two dissimilarity and similarity matrices corresponding to a strictly monotone decreasing transformation produce similar dendrograms, as shown in the last section of the previous chapter. In other words, a theoretical result proved by assuming a similarity measure is also true for a dissimilarity measure, and vice versa. Specifically, we assume a similarity measure when we discuss fuzzy graphs and fuzzy relations [3].

S. Miyamoto, *Theory of Agglomerative Hierarchical Clustering*,
Behaviormetrics: Quantitative Approaches to Human Behavior 15,
https://doi.org/10.1007/978-981-19-0420-2_3

3.1 Network Algorithm and the Single Linkage

There is a well-known algorithm to find a minimum spanning tree (MST) of a given network. A network algorithm in this chapter is just for MST, although we mainly consider a maximum spanning tree (also abbreviated MST) instead of minimum. The equivalence between the single linkage (SL) and MST has already been described in the early book by Anderberg [4].

Brief Review of Graph Concept

Let us briefly review the concept of *minimum/maximum spanning tree*. A graph is a pair (V, E) where $V = \{v_1, \ldots, v_N\}$ is a nonempty finite set of *nodes*. *Edges* E is a subset of $V \times V$ such that if $e = (v_i, v_j) \in E$ then $(v_i, v_j) \in E$. An edge can also be represented by an unordered pair $e = \{v_i, v_j\}$ in view of the last property. A graph can be visualized using points on the plane expressing nodes and lines or curves connecting nodes expressing edges. A *complete graph* is a graph (V, E) where $E = V \times V$. When a complete graph is expressed as a figure, every pair of nodes is connected by a line. We sometimes use a single symbol $\boldsymbol{G}$ for a graph: In this case $V(\boldsymbol{G})$ and $E(\boldsymbol{G})$ means the node set and the edge set of $\boldsymbol{G}$, respectively.

Note that an arrow is not used, but a line or curve is used for visualizing an edge, since an edge is essentially an unordered pair.

A graph (V, E) is said to have a *loop* at v if there is an edge $(v, v) \in E$. This property will be used later in this chapter.

A graph (V', E') is called a *subgraph* of (V, E) if $V' \subseteq V$ and $E' \subseteq E$. A subgraph is called a *proper subgraph* if $(V', E') \neq (V, E)$. Moreover (V, E) is called a *supergraph* of (V', E') if (V', E') is a *subgraph* of (V, E). A *proper supergraph* is defined in the same way. A subgraph (V', E') of (V, E) is called a *spanning subgraph* if $V' = V$. The relations of a subgraph and supergraph can be expressed by $\boldsymbol{G}' \subseteq \boldsymbol{G}$; a proper subgraph/supergraph is expressed by $\boldsymbol{G}' \subset \boldsymbol{G}$.

A graph (V, E) has a *cycle* if there is a node $v_{i_1} \in E$ and there are edges $(v_{i_k}, v_{i_{k+1}}) \in E$ for $k = 1, 2, \ldots, \ell$ and $v_{i_\ell} = v_{i_1}$. When E is visualized, a cycle means that we find a number of lines (or curves) starting from a point, connecting subsequent points, and finally arriving at the starting point.

For two nodes $v, w \in V$, there is a *walk* connecting v and w if there is a sequence of edges $(v_{i_k}, v_{i_{k+1}}) \in E$ for $k = 1, 2, \ldots, L-1$ such that $v = v_{i_1}$ and $w = v_{i_L}$. The number L is called the length of the walk. A graph is said to be *connected*, if for any two different nodes of that graph, there is a walk connecting the two nodes.

A subgraph $\boldsymbol{G}'$ of $\boldsymbol{G}$ is called a *connected component* if $\boldsymbol{G}'$ is connected and there is no connected proper supergraph $\boldsymbol{G}''$ ($\boldsymbol{G}' \subset \boldsymbol{G}''$) that is a subgraph of $\boldsymbol{G}$ ($\boldsymbol{G}'' \subseteq \boldsymbol{G}$).

A *tree* is a graph without any cycle. A *spanning tree* of a graph $\boldsymbol{G}$ is a spanning subgraph of $\boldsymbol{G}$ without a cycle.

A *weighted graph* or *network* $\boldsymbol{N} = (V, E, W)$ generally consists of a node set V, an edge set E, and also a set of weight W defined on E with real values: $W: E \to (-\infty, +\infty)$. In this chapter, the object set X with dissimilarity/similarity: (X, D) or (X, S) is considered. They are regarded as networks by assuming $V = X$, $E = X \times X$ (complete graph), and $W = D$ or $W = S$. For $W = D$, the range is $[0, \infty)$, and for $W = S$, the range is $[0, 1]$.

For a weighted graph $\boldsymbol{N}$, a minimum spanning tree $\boldsymbol{T}_N$ is the spanning tree in which the sum of weights of $\boldsymbol{T}_N$ is the minimum of all spanning trees of $\boldsymbol{N}$ as the graph (V, E). Let $\mathcal{T}(\boldsymbol{N})$ be the set of all spanning trees of $\boldsymbol{N}$. Then,

$$\boldsymbol{T}_N = \arg \min_{\boldsymbol{T} \in \mathcal{T}(\boldsymbol{N})} \sum_{e \in E(\boldsymbol{T})} W(e). \tag{3.1}$$

We use the minimum spanning tree when we use dissimilarity. In contrast, we use the maximum spanning tree when we use similarity. Hence, a maximum spanning tree $\boldsymbol{T}'_N$ is the spanning tree in which the sum of weights of $\boldsymbol{T}_N$ is maximum of all spanning trees of $\boldsymbol{N}$ as the graph. Let $\mathcal{T}(\boldsymbol{N})$ be the set of all spanning trees of $\boldsymbol{N}$. Then,

$$\boldsymbol{T}'_N = \arg \max_{\boldsymbol{T} \in \mathcal{T}(\boldsymbol{N})} \sum_{e \in E(\boldsymbol{T})} W(e). \tag{3.2}$$

Fuzzy Graph

A *fuzzy graph* $\boldsymbol{FG} = (V, E, \mu)$ is a network in which $\mu: E \to [0, 1]$. The reason why a network is called a fuzzy graph is that a fuzzy graph is a collection of its α-cuts. An α-cut has a parameter $\alpha \in [0, 1]$ and a family of graphs $[\boldsymbol{FG}]_\alpha = ([V]_\alpha, [E]_\alpha)$, where $V = [V]_\alpha$ for all $0 \leq \alpha \leq 1$ (later we consider an exceptional case when $V \neq [V]_\alpha$), while $[E]_\alpha$ is defined as follows:

$$(v, w) \in [E]_\alpha \iff \mu((v, w)) \geq \alpha. \tag{3.3}$$

Accordingly, we define α-connected components as the family of the connected components of $[\boldsymbol{FG}]_\alpha$, $0 \leq \alpha \leq 1$.

3.1.0.1 Kruskal's Algorithm

Kruskal's algorithm for MST [5] is useful for showing the equivalence property in this chapter. We describe a version of this algorithm, which includes clusters of nodes of a network [6].

An MST Algorithm [6].
MST1: Let $VS = \{\{v_1\} \ldots, \{v_N\}\}$ and $\underline{\hat{E} = \emptyset}$.

MST2: Sort the set of edges E into the decreasing order of the weight W (we consider the maximum spanning tree; in the case of the minimum spanning tree, sort into the increasing order) and set the result in the sequence Q_E. (Q_E is a queue: the first element is the edge with the maximum weight. When the first element is deleted, the edge with the second maximum weight becomes first, and so on.)

MST3: Take the first element $e = (v, w) \in Q_E$ of the current maximum weight.

MST4: If v and w are in different subsets of VS, i.e., $v \in V'$ and $w \in V''$ ($V', V'' \in VS$, $V' \cap V'' = \emptyset$), remove V' and V'' from VS and add $V' \cup V''$ to VS (i.e., $VS = VS \cup \{V' \cup V''\} - \{V', V''\}$). Let $\hat{E} = \hat{E} \cup e$. Else do nothing.

MST5: Delete e from Q_E and push the queue so that the second maximum element comes first.

MST6: If the number of elements in VS is 1 ($|VS| = 1$), stop. Else go to **MST3**.

End MST.

This algorithm outputs $\hat{E}$ as the set of edges of MST. However, we note that this algorithm actually generates clusters in VS.

Insightful readers already noted a close relation between the MST clusters and the single linkage clusters.

We proceed to prove the equivalence property. Let us delete the underlined parts concerning $\hat{E}$, and note that the algorithm is now purely for clustering. Let us moreover assume that **MST3** is replaced by

MST3': Take the first element $e = (v, w) \in Q_E$ of the current maximum weight. We assume that an optional parameter $\alpha \in [0, 1]$ is set, and if $W(e) < \alpha$, stop.

and call the modified algorithm **MST**(α). In case when the minimum spanning tree is considered, the condition "if $W(e) < \alpha$" is replaced by "if $W(e) > \alpha$".

Then the following property holds. Note that a similarity (and not a dissimilarity) is used for weight W.

Proposition 2 *Clusters by* **MST**(α)*, i.e., the subsets of the nodes in VS is the nodes of the connected components of $\boldsymbol{FG}_\alpha$, where $\boldsymbol{FG} = (V, E, W)$, i.e., the network is regarded as a fuzzy graph.*

Proof We show that the subsets in VS is the α-connected components. Assume Q_E consists of $(e_1, e_2, \ldots, e_m)$ in this order without loss of generality. Let $\alpha_1 = W(e_1) = \cdots = W(e_k) > W(e_{k+1})$. Then any set V of VS at that time is connected by a walk taken from $e_1, \ldots, e_k$ and two different sets in VS are not connected by these edges. Thus the elements in VS are α_1 connected. Hence we put $VS_{\alpha_1} = VS$. We set $\alpha_2 = W(e_{k+1}) = \cdots = W(e_l) > W(e_{l+1})$, and then the same argument yields that the elements in VS are α_2-connected. We put $VS_{\alpha_2} = VS$. In this way, VS_{α_i} ($i = 1, 2, \ldots$) are formed until VS has just one element. For any $\alpha \in (\alpha_i, \alpha_{i+1})$, the connected components remain unchanged. If we put $VS_\alpha = VS_{\alpha_i}$ for $\alpha \in (\alpha_i, \alpha_{i+1})$, VS_α is defined for all α and for any element V' of VS_α, any two nodes in V' are connected but two elements $V', V'' \in VS_\alpha$ are not connected. Thus VS_α has just α-connected components of $\boldsymbol{FG}$. □

We next consider the single linkage clusters and $\boldsymbol{FG}$. For this purpose the agglomerative clustering procedure **AHC** is modified: after

$$(G_p, G_q) = \arg\max_{i,j} S(G_i, G_j) \tag{3.4}$$

we insert a statement

if $S(G_p, G_q) < \alpha$, output the current clusters as $\mathcal{G}(\alpha)$ and stop.

Note that we are using similarity S.

Note 6 When we use a dissimilarity, (3.4) is replaced by

$$(G_p, G_q) = \arg\min_{i,j} D(G_i, G_j) \tag{3.5}$$

and the statement is replaced by

if $D(G_p, G_q) > \alpha$, output the current clusters as $\mathcal{G}(\alpha)$ and stop.

We prove the next result.

Proposition 3 *The clusters $\mathcal{G}(\alpha)$ derived from the modified* **AHC** *is the same as the collection of node sets of the α-connected components of $\boldsymbol{FG}$; in other words, the connected components of $[\boldsymbol{FG}]_\alpha$, the α-cut of $\boldsymbol{FG}$.*

Proof Let us note that

$$S(G_p, G_q) = \max_{v \in G_p, w \in G_q} S(x, y),$$

which means that there is an edge $e_{pq} = (v', w')$ such that $v' \in G_p$, $w' \in G_q$ and $W(e_{pq}) = S(v', w') = S(G_p, G_q) \geq \alpha$ connects nodes in G_p and G_q by a walk and the weights of the edges of the walk are all greater than or equal to α. On the other hand, different clusters in $\mathcal{G}(\alpha)$ are not connected by such a walk. Thus, the two clusters: $\mathcal{G}(\alpha)$ and the nodes in the α-connected components of $\boldsymbol{FG}$ are the same. □

We thus have seen the three clusters of

- those generated by the single linkage,
- those produced by the MST algorithm, and
- the node sets of α-connected components of the fuzzy graph $\boldsymbol{FG}$.

are the same.

There is still another result of equivalence: the *transitive closure* of a fuzzy relation, which needs another algebraic consideration.

3.2 Max–Min Composition and Transitive Closure

Let $X = \{x_1, \ldots, x_N\}$ be an object set and $S = [S(x_i, x_j)]$ $(0 \leq S(x_i, x_j) \leq 1)$ be a similarity measure as usual. We also set $S_{ij} = S(x_i, x_j)$ and regard $S = [S_{ij}]$ as an $N \times N$ matrix. We moreover assume $S(x_i, x_i) = S_{ii} = 1$ for $1 \leq i \leq N$. We regard S as a fuzzy relation [3, 7] on $X \times X$ and consider clustering using operations of fuzzy relations. The main purpose of this section is an algebraic formulation of clustering that is equivalent to the single linkage having possibility to study further extensions.

As noted in Chap. 1, S is reflexive and symmetric. If S is transitive (see (1.29)) in addition, S directly expresses hierarchical clustering. In real applications, however, S is not transitive in general. Hence what we should consider is to derive a transitive relation from a given reflexive and symmetric (nontransitive) relation. For this purpose we need a preliminary consideration, i.e., operations of fuzzy relations.

Composition of Fuzzy Relations

Let X, Y, and Z be three finite sets. Assume that T and U are fuzzy relations on $X \times Y$ and $Y \times Z$, i.e., $T \subseteq X \times Y$ and $U \subseteq Y \times Z$, but we use prefix notation of $T(x, y) = 1$ or 0 and $U(y, z) = 1$ or 0. T and U need not be reflexive nor symmetric.

A *max–min composition* (or simply composition) $T \circ U$ is a fuzzy relation on $X \times Z$ defined as follows:

$$(T \circ U)(x, z) = \max_{y \in Y} \min\{T(x, y), U(y, z)\}. \tag{3.6}$$

Assume $X = Y = Z$ hereafter. We write

$$T^2 = T \circ T, \quad T^k = T^{k-1} \circ T, \quad k = 2, 3, \ldots \tag{3.7}$$

Moreover, we define $T \vee U$:

$$(T \vee U)(x, y) = \max\{T(x, y), U(x, y)\}. \tag{3.8}$$

We now define the transitive closure of a fuzzy relation.

Definition 7 Let T be a fuzzy relation on X. The *transitive closure* of T is defined by

$$T^* = T \vee T^2 \vee \cdots \vee T^k \vee \cdots \tag{3.9}$$

We have the next proposition.

Proposition 4 *There exists a natural number h such that*

$$T^* = T \vee T^2 \vee \cdots \vee T^h. \tag{3.10}$$

Proof Let

$$T[k] = T \vee T^2 \vee \cdots \vee T^k, \quad k = 1, 2, \ldots \tag{3.11}$$

Let $\mathcal{T} = \{T(x, y) : \forall x, y \in X\}$. Since X is finite, the set $\mathcal{T}$ is also finite.

For arbitrary $x, y \in X$, consider $T[k](x, y)$. Then $T[k](x, y)$ is monotone non-decreasing as k increases. Since $T[k](x, y) \in \mathcal{T}$ and $\mathcal{T}$ is finite, there is a number $l(x, y)$ such that when $k = l(x, y)$, $T[k](x, y)$ will not increase any more: $T[k'](x, y) = T[k](x, y)$, for $k' = k, k + 1, \ldots$. Let

$$h = \max_{x,y \in X} l(x, y). \tag{3.12}$$

Then $T[k'] = T[h]$ for $k' = h + 1, \ldots$. Thus we have $T^* = T[h]$. □

We now proceed to consider S that is reflexive and symmetric. We have the next theorem.

Theorem 8 *Assume that S is a fuzzy relation on X that is reflexive and symmetric. Then,*

$$S^{k-1} \subseteq S^k, \quad k = 2, 3, \ldots \tag{3.13}$$

and there exists h such that

$$S^* = S^h. \tag{3.14}$$

The transitive closure S^* is transitive:

$$S^* = S^* \circ S^*. \tag{3.15}$$

Moreover, S^* is reflexive and symmetric, and hence S^* is a fuzzy equivalence relation.

Proof

$$\begin{aligned} S^k(x, y) &= \max_{z \in X} \min\{S^{k-1}(x, z), S(z, y)\} \\ &\geq \min\{S^{k-1}(x, y), S(y, y)\} = S^{k-1}(x, y). \end{aligned}$$

Hence $S^{k-1} \subseteq S^k$. This implies that $S[k] = S \vee \cdots \vee S^k = S^k$. Accordingly, the previous proposition implies that

$$S^* = S[h] = S^h.$$

Note that $S(x, x) = 1$ is the maximum value, and if we assume $S^{k-1}(x, x) = 1$, $S^{k-1} \subseteq S^k$ implies $S^k(x, x) = 1$. Hence $S^* = S^h$ is reflexive.

Assume $S^{k-1}(x, y)$ is symmetric.

$$\begin{aligned} S^k(x, y) &= \max_{z \in X} \min\{S^{k-1}(x, z), S(z, y)\} \\ &= \max_{z \in X} \min\{S^{k-1}(z, x), S(y, z)\} \\ &= (S \circ S^{k-1})(y, x) = S^k(y, x). \end{aligned}$$

We thus have the symmetry of S^h and hence S^* is symmetric.
The transitivity of S^* is also easily seen. Since $S^* = S^h$

$$S^* = S^h \subseteq S^h * S^h = S^* \circ S^*.$$

On the other hand, what we have seen above is $S^h = S^{h+1} = \cdots = S^h * S^h$. Hence we have

$$S^* = S^* \circ S^*,$$

which means the transitivity. □

Observing the equivalence between a fuzzy equivalence relation and hierarchical clusters discussed in Chap. 1, we see that for a given similarity measure S the transitive closure S^* provides us a family of hierarchical clusters $\mathcal{G}(\alpha)$ depending on a parameter $\alpha \in [0, 1]$.

Composition for Dissimilarity Measures

As seen in Chap. 1, we can discuss another version of the above composition in terms of dissimilarity measure. W thus consider a *min–max composition*:

Definition 8 For two dissimilarity measures D and D' defined on X, the min–max composition is defined by

$$(D \bullet D')(x, y) = \min_{z \in X} \max\{D(x, z), D(z, y)\}. \tag{3.16}$$

We write

$$D^k = D^{k-1} \bullet D, \quad k = 2, 3, \ldots \tag{3.17}$$

We also define

$$(D \wedge D')(x, y) = \min\{D(x, y), D'(x, y)\}. \tag{3.18}$$

We moreover define the closure of a dissimilarity measure.

Definition 9 For a given dissimilarity measure D on X, the closure of D is defined by

$$D^\star = D \wedge D^2 \wedge \cdots D^k \wedge \cdots \tag{3.19}$$

The following result can easily be seen by using a monotone function F and a similarity measure S and putting $D = F(S)$, as seen in Chapter 1. We hence omit the proofs.

Theorem 9 *Assume that D is a dissimilarity measure on X, i.e., $D(x, x) = 0$ and $D(x, y) = D(y, x)$ for all $x, y \in X$. Then, there is a natural number ℓ such that*

$$D^\star = D^\ell. \tag{3.20}$$

Moreover, $D^\star$ is an ultrametric and satisfies

$$D^\star = D^\star \bullet D^\star. \tag{3.21}$$

As seen in Chapter 1, the closure $D^\star$ derived from D provides a hierarchical cluster $\mathcal{G}(\lambda)$ for $\lambda \in [0, +\infty)$.

3.3 Transitive Closure and Fuzzy Graph

We proceed to prove the equivalence between clusters derived from S^* and those of fuzzy graph defined from (X, S). The main result of this section is the following.

Theorem 10 *Assume that S is a similarity measure on X. Let VS_α be the collection of node sets of connected components of $\boldsymbol{FG}_\alpha$, where $\boldsymbol{FG}$ is fuzzy graph defined from (X, S) (i.e., $\boldsymbol{FG}$ is the weighted complete graph of which the set of nodes is X and the weight is S) and $\boldsymbol{FG}_\alpha$ is its α-cut. Then VS_α is the same as the partition derived from $[S^*]_\alpha$:*

$$\mathcal{G}(\alpha) = \{[x]_{[S^*]_\alpha} : x \in X\}, \tag{3.22}$$

i.e., $VS_\alpha = \mathcal{G}(\alpha)$. (Note that the symbol VS_α is used in the proof of Proposition 2.)

In order to prove this theorem, we need to prove the next proposition.

Proposition 5 *For a given similarity S, $S^k(x, y)$ means that there is a walk of length k between x and y in $\boldsymbol{FG}_\alpha$ for $\alpha \leq S^k(x, y)$. At the same time, there is no walk of length k between x and y in and there is no walk $\boldsymbol{FG}_\alpha$ for $\alpha > S^k(x, y)$.*

Proof The value $S(x, y)$ is the maximum α such that x and y is connected by an edge in $\boldsymbol{FG}_\alpha$, and $S^2(x, y)$ is the maximum α such that x and y is connected by a walk of length 2 in $\boldsymbol{FG}_\alpha$, since

$$S^2(x, y) = \max_{z \in X} \min\{S(x, z), S(z, y)\}.$$

Accordingly, it is immediately seen that $S^k(x, y)$ is the maximum α such that x and y are connected by a walk of length k in $\boldsymbol{FG}_\alpha$ from

$$S^k(x, y) = \max_{z \in X} \min\{S^{k-1}(x, z), S(z, y)\}$$

using the mathematical induction for k. □

We hence prove the following.

Proposition 6 *For a given similarity S, $S^*(x, y)$ means the maximum value of α such that x and y are connected in $\boldsymbol{FG}_\alpha$. In other words, if $\alpha \leq S^*(x, y)$, x and y are connected in $\boldsymbol{FG}_\alpha$. If $\alpha > S^*(x, y)$, x and y are not connected in $\boldsymbol{FG}_\alpha$.*

Proof The conclusion is immediate by observing that x and y are connected in $\boldsymbol{FG}_\alpha$ if and only if there is a walk between x and y in $\boldsymbol{FG}_\alpha$ and $S^* = S \vee \cdots \vee S^k \vee \cdots$ □

It is now immediate to see that Theorem 10 holds from the last proposition.

We have thus proved the equivalence of the clusters obtained by the four methods stated in the beginning of this chapter: the single linkage, the MST algorithm, the connected components of the fuzzy graph, and the transitive closure of the fuzzy relation. Note that the transitive closure can be derived from dissimilarity D instead of S using the min–max composition and $D^\star$.

3.4 An Illustrative Example

We consider a simple example to see the equivalence of the above methods.

Let us see the left illustration of a weighted graph in Fig. 3.1. The graph shows an object set $X = \{A, B, C, D\}$ and a similarity measure:

$$S = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{pmatrix} 1 & 0.5 & 0 & 0.9 \\ 0.5 & 1 & 0.3 & 0.4 \\ 0 & 0.3 & 1 & 0.7 \\ 0.9 & 0.4 & 0.7 & 1 \end{pmatrix} \end{array}. \tag{3.23}$$

Although the graph has a loop on each node, we omit these loops and write the figure '1' at each node instead, to simplify the illustration shown as the right graph in Fig. 3.1. Such a simplification is nonstandard in graph theory literature, but useful in writing a *fuzzy graph* and used throughout this book.

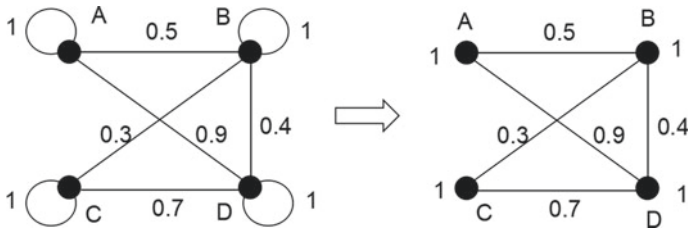

Fig. 3.1 A weighted graph with loops and the corresponding fuzzy graph: a loop on each node is omitted and only a figure '1' is attached on each node. An edge with weight zero is omitted in both graphs

If we apply the single linkage to the above, the following clusters are generated:

Level	Clusters
0.9	$\{A, D\}$, $\{B\}$, $\{C\}$
0.7	$\{A, C, D\}$, $\{B\}$
0.5	$\{A, B, C, D\}$

The next table shows the result of Kruskal's algorithm generating clusters in VS and MST in $\hat{E}$. The last three rows are unnecessary from the viewpoint of clustering. The results in VS clearly coincides with the above clusters by the single linkage. Note also that the MST $\hat{E}$ is unnecessary for clustering.

Edge	Weight	VS	MST: $\hat{E}$
(A, D)	0.9	$\{A, D\}, \{B\}, \{C\}$	$\{(A, D)\}$
(C, D)	0.7	$\{A, C, D\}, \{B\}$	$\{(A, D), (C, D)\}$
(A, B)	0.5	$\{A, B, C, D\}$	$\{(A, D), (C, D), (A, B)\}$
(B, D)	0.4	$\{A, B, C, D\}$	$\{(A, D), (C, D), (A, B)\}$
(B, C)	0.3	$\{A, B, C, D\}$	$\{(A, D), (C, D), (A, B)\}$
(A, C)	0	$\{A, B, C, D\}$	$\{(A, D), (C, D), (A, B)\}$

The graph in Fig. 3.1 is regarded as a fuzzy graph, and accordingly its α-cuts are shown in Fig. 3.2. The connected components are as follows:

α-cut	Node sets for connected components
$\alpha = 0.85$	$\{A, D\}, \{B\}, \{C\}$
$\alpha = 0.7$	$\{A, C, D\}, \{B\}$
$\alpha = 0.5$	$\{A, B, C, D\}$

We note again that the clusters are the same as the single linkage clusters. The α in the first row is 0.85, but we can change it to $\alpha = 0.9$ without changing the connected components.

What we see finally is the transitive closure by the max–min algebra on S. We have

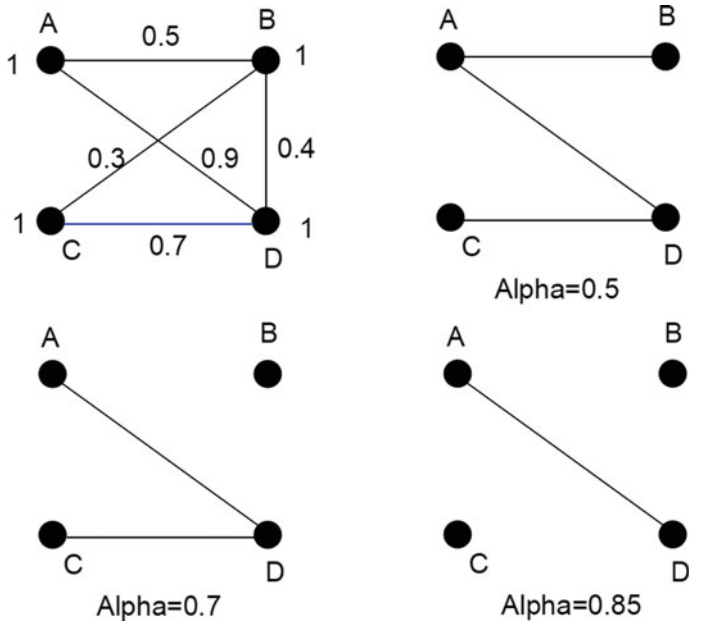

Fig. 3.2 A fuzzy graph and its α-cuts

$$S^2 = S \circ S = \begin{pmatrix} 1 & 0.5 & 0 & 0.9 \\ 0.5 & 1 & 0.3 & 0.4 \\ 0 & 0.3 & 1 & 0.7 \\ 0.9 & 0.4 & 0.7 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 & 0.5 & 0 & 0.9 \\ 0.5 & 1 & 0.3 & 0.4 \\ 0 & 0.3 & 1 & 0.7 \\ 0.9 & 0.4 & 0.7 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0.5 & 0.7 & 0.9 \\ 0.5 & 1 & 0.4 & 0.4 \\ 0.7 & 0.4 & 1 & 0.7 \\ 0.9 & 0.4 & 0.7 & 1 \end{pmatrix}.$$

Moreover, we have

$$S^* = S^3 = S^2 \circ S = \begin{pmatrix} 1 & 0.5 & 0.7 & 0.9 \\ 0.5 & 1 & 0.4 & 0.4 \\ 0.7 & 0.4 & 1 & 0.7 \\ 0.9 & 0.4 & 0.7 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 & 0.5 & 0 & 0.9 \\ 0.5 & 1 & 0.3 & 0.4 \\ 0 & 0.3 & 1 & 0.7 \\ 0.9 & 0.4 & 0.7 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0.5 & 0.7 & 0.9 \\ 0.5 & 1 & 0.5 & 0.5 \\ 0.7 & 0.5 & 1 & 0.7 \\ 0.9 & 0.5 & 0.7 & 1 \end{pmatrix}.$$

When S^* is cut at $\alpha = 0.9$, the (i, j) component of $[S^*]_{0.9}$ is 1 if it is greater than or equal to $\alpha = 0.9$; otherwise the component is 0. Thus we have

$$[S^*]_{0.9} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix},$$

which represents clusters $\{A, D\}, \{B\}, \{C\}$, since A and D are connected by an edge, while other edges are not connected.

Moreover,

$$[S^*]_{0.7} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix},$$

$$[S^*]_{0.5} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix},$$

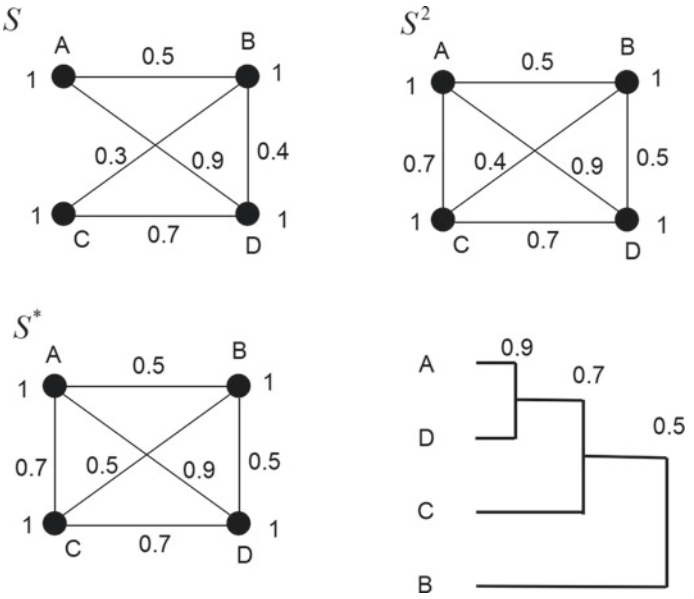

Fig. 3.3 S, S^2, and S^*. Note also the dendrogram derived from S^*

which respectively correspond to clusters $\{A, C, D\}$, $\{B\}$, and $\{A, B, C, D\}$. Note that each $[S^*]_\alpha$ represents equivalence relations.

Figure 3.3 shows S^2 and S^* as weighted graphs. It also shows a dendrogram derived from S^*, which is just the result of the single linkage.

3.5 A Refinement Theory

The definition of refinement in clustering was given in Chapter 1. In this section we generalize the definition and consider refinement relations between different clustering methods. Throughout this section, a dissimilarity measure D is used, but the same results also hold for similarity measures.

Let us remind that for given two clusters $\mathcal{G}$ and $\mathcal{G}'$, $\mathcal{G}$ is called a *refinement* of $\mathcal{G}'$ if for every $G_i \in \mathcal{G}$, there exists $G'_j \in \mathcal{G}'$ such that $G_i \subseteq G'_j$ and we write $\mathcal{G} \prec \mathcal{G}'$ if $\mathcal{G}$ is a refinement of $\mathcal{G}'$.

We use subscript symbols like $\mathcal{G}_A(\alpha)$ when a linkage method A of hierarchical clustering is used. An example is $\mathcal{G}_{SL}(\alpha)$ when the single linkage is used. Note that when we write $\mathcal{G}_A(\alpha) \prec \mathcal{G}_B(\alpha)$ for linkage methods A and B, the refinement property holds for an arbitrary dissimilarity or similarity measure.

Let us assume that initial clusters consist of one element ($G_i = \{x_i\}, i = 1, 2, \ldots, N$). We prove the following proposition.

Proposition 7 *If a linkage method A does not have any reversal (reversed dendrogram) and satisfies*

$$D_A(G, G') \geq \min_{x \in G, y \in G'} D(x, y) \tag{3.24}$$

for any two clusters G, G', then

$$\mathcal{G}_A(\alpha) \prec \mathcal{G}_{SL}(\alpha), \quad 0 \leq \alpha < +\infty, \tag{3.25}$$

i.e., clusters generated by linkage method A are the refinements of those by the single linkage.

Proof Let us suppose at some α_0, $\mathcal{G}_A(\alpha_0) \prec \mathcal{G}_{SL}(\alpha_0)$ holds. Note that this is true at the initial stage $\alpha = 0$ when all clusters consist of one object: $G_i = \{x_i\}, i = 1, 2, \ldots, N$. Suppose also that next merging occurs at $\alpha \geq \alpha_0$ for linkage method A: there is a merged pair G_p and G_q for method A, and there are two clusters G' and G'' for SL at α_0 such that $G_p \subseteq G'$ and $G_q \subseteq G''$. Let us suppose G' and G'' are not merged at α by the single linkage and the refinement property is broken.

When we observe (3.24), however, G' and G'' are also merged at α by the single linkage. Thus the proposition is proved. □

Note 7 If the initial clusters may have more than one elements: $\mathcal{G}(N_0)$ $(N_0 < N)$, we assume

$$D_A(G, G') \geq D_{SL}(G, G') = \min_{x \in G, y \in G'} D(x, y), \quad \forall G, G' \in \mathcal{G}(N_0). \tag{3.26}$$

Then the conclusion of Proposition 7 also holds. The proof is a slight modification of the above.

Proposition 8 *If a linkage method A does not have any reversal (reversed dendrogram) and satisfies*

$$D_A(G, G') \geq D_{SL}(G, G') \tag{3.27}$$

for any two clusters G, G', then

$$\mathcal{G}_A(\alpha) \prec \mathcal{G}_{SL}(\alpha), \quad 0 \leq \alpha < +\infty, \tag{3.28}$$

i.e., clusters generated by linkage method A are the refinements of those by the single linkage.

The proof is easy by observing the definition $D_{SL}(G, G') = \min_{x \in G, y \in G'} D(x, y)$, and hence omitted.

Next proposition concerns updating formulas.

Proposition 9 *If a linkage method A has an updating formula that satisfies*

$$D_A(G_r, G_j) \geq \min\{D_A(G_p, G_j), D_A(G_q, G_j)\}, \tag{3.29}$$

where $G_r = G_p \cup G_q$ and the initial clusters satisfy (3.26), then

$$\mathcal{G}_A(\alpha) \prec \mathcal{G}_{SL}(\alpha), \quad 0 \leq \alpha < +\infty, \tag{3.30}$$

i.e., clusters generated by linkage method A are the refinements of those by the single linkage.

Proof It is easy to prove that the linkage method A does not have any reversal since

$$D_A(G_r, G_j) \geq \min\{D_A(G_p, G_j), D_A(G_q, G_j)\} \geq D_A(G_p, G_q),$$

which means the monotonicity of the merging levels of the linkage method A.

Note next that from $D_{SL}(G_r, G_j) = \min\{D_{SL}(G_p, G_j), D_{SL}(G_q, G_j)\}$ and (3.26), Eq. (3.29) implies

$$D_A(G_p, G_j) \geq \min_{x \in G_p, y \in G_j} D(x, y).$$

The rest of the proof is essentially the same as that of the previous proposition and the details are omitted. □

We hence have the next theorem.

Theorem 11 *The complete linkage, the average linkage, and Ward method satisfy the refinement property with respect to the single linkage method:*

$$\mathcal{G}_{CL}(\alpha) \prec \mathcal{G}_{SL}(\alpha), \tag{3.31}$$
$$\mathcal{G}_{AL}(\alpha) \prec \mathcal{G}_{SL}(\alpha), \tag{3.32}$$
$$\mathcal{G}_{WRD}(\alpha) \prec \mathcal{G}_{SL}(\alpha), \tag{3.33}$$

for $0 \leq \alpha < +\infty$. *Note that* $D(G, G') = \Delta E(G, G')$ *and* $D(x, y) = \frac{1}{2}\|x - y\|^2$ *for Ward method.*

Proof For the complete linkage *CL* and the average linkage *AL*, it is clear that they satisfy the property (3.24).

For Ward method, let $\beta = D_{WRD}(G_p, G_q)$ and

$$\gamma = \min\{D_{WRD}(G_p, G_j), D_{WRD}(G_q, G_j)\}.$$

Then we have $\gamma \geq \beta$. From (2.26) and $|G_r| = |G_p| + |G_q|$, we have

$$\begin{aligned} D(G_r, G_j) &= \frac{1}{|G_r| + |G_j|}\{(|G_p| + |G_j|)D(G_p, G_j) + (|G_q| + |G_j|)D(G_q, G_j) \\ &\quad - |G_j|D(G_p, G_q)\} \\ &\geq \frac{1}{|G_r| + |G_j|}\{(|G_p| + |G_j|)\gamma + (|G_q| + |G_j|)\gamma - |G_j|\beta\} \\ &\geq \frac{1}{|G_r| + |G_j|}\{(|G_p| + |G_j|)\gamma + (|G_q| + |G_j|)\gamma - |G_j|\gamma\} = \gamma. \end{aligned}$$

Hence Proposition 9 is applied and (3.33) is proved. □

If we define

$$\mathcal{G}_A \prec \mathcal{G}_{SL} \iff \mathcal{G}_A(\alpha) \prec \mathcal{G}_{SL}(\alpha), \quad \text{for all } 0 \leq \alpha < +\infty, \tag{3.34}$$

for linkage method *A* and the single linkage *SL*, then we have

$$\mathcal{G}_{CL} \prec \mathcal{G}_{SL}, \tag{3.35}$$
$$\mathcal{G}_{AL} \prec \mathcal{G}_{SL}, \tag{3.36}$$
$$\mathcal{G}_{WRD} \prec \mathcal{G}_{SL}, \tag{3.37}$$

from the above theorem.

3.6 A Variation of the Single Linkage Method

Let us consider a variation of the single linkage in which each object has a 'degree of qualification'. For this purpose we consider a fuzzy graph induced by (X, S, A): X and S are respectively an object set and a similarity measure. In addition, $A = (a_1, \ldots, a_N)$ is a fuzzy set on $X = \{x_1, \ldots, x_N\}$.

Note 8 Readers do not need to have knowledge of *fuzzy sets* but simply can regard $A = (a_1, \ldots, a_N)$ as a vector of which components are in the unit interval: $0 \leq a_i \leq 1$ $(i = 1, \ldots, N)$.

The interpretation of this variation is simple: we consider a modified similarity $S' = (s'_{ij})$ where

$$s'_{ij} = \min\{a_i, a_j, s_{ij}\} \tag{3.38}$$

and apply the single linkage for (X, S').

The reason why we consider S and A is that there have been algorithms [8, 9] related to the consideration of A. The idea is to 'qualify' objects for clustering. The objects x_i and x_j are not qualified when the level L to generate clusters is more than $\min\{a_i, a_j\}$. When $L \leq \min\{a_i, a_j\}$, the both objects are qualified.

We observe a simple example.

Example 10 Let us consider again the graph in Fig. 3.1. We define A to be the second largest value of the edges from each node. For example, value at B should be selected from {0.5, 0.4, 0.3} and the second largest value is 0.4. Thus A is shown at each node of the left graph in Fig. 3.4. Thus $A = (0.5, 0.4, 0.3, 0.7)$ and the modified graph (X, S') is shown as the right graph in the same figure:

$$S' = \begin{pmatrix} 1 & 0.4 & 0 & 0.5 \\ 0.4 & 1 & 0.3 & 0.4 \\ 0 & 0.3 & 1 & 0.3 \\ 0.5 & 0.4 & 0.3 & 1 \end{pmatrix}. \tag{3.39}$$

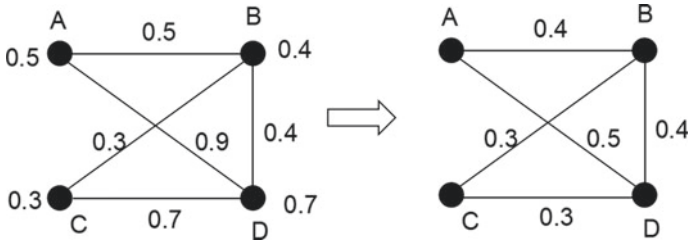

Fig. 3.4 A fuzzy graph induced by (X, S, A) and (X, S')

The hierarchical cluster by the single linkage is as follows.

Level	Clusters
0.5	$\{A, D\}, \{B\}, \{C\}$
0.4	$\{A, B, D\}, \{C\}$
0.3	$\{A, B, C, D\}$

The corresponding transitive closure is given as follows:

$$(S')^* = \begin{pmatrix} 1 & 0.4 & 0.3 & 0.5 \\ 0.4 & 1 & 0.3 & 0.4 \\ 0.3 & 0.3 & 1 & 0.3 \\ 0.5 & 0.4 & 0.3 & 1 \end{pmatrix}. \tag{3.40}$$

Note that the results of the orderings of the merging are different between (X, S) and (X, S').

We will see that two methods [8, 9] are closely related to this idea in a later chapter.

References

1. S. Miyamoto, *Fuzzy Sets in Information Retrieval and Cluster Analysis* (Springer, Heidelberg, 1990)
2. L.A. Zadeh, Fuzzy sets. Inf. Control **8**, 338–353 (1965)
3. D. Dubois, H. Prade, *Fuzzy Sets and Systems* (Academic Press, New York, 1988)
4. M.R. Anderberg, *Cluster Analysis for Applications* (Academic Press, New York, 1973)
5. J.B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem. Proc. Am. Math. Soc. **7**, 48–50 (1956)
6. A.V. Aho, J.E. Hopcroft, J.D. Ullman, *The Design and Analysis of Computer Algorithms* (Adddison-Wesley, Reading, Massachusetts, 1974)
7. L.A. Zadeh, Similarity relations and fuzzy orderings. Inf. Sci. **3**(2), 177–200 (1971)
8. M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in *KDD-96 Proceedings* (1996), pp. 226–231
9. D. Wishart, Mode analysis: a generalization of nearest neighbor which reduces chaining effects, in ed. by A.J. Cole *Numerical Taxonomy, Proceedings of the Colloquium, in Numerical Taxonomy* (University of St Andrews, 1968), pp. 283–311

Chapter 4
Positive-Definite Kernels in Agglomerative Hierarchical Clustering

Positive-definite kernels are popular in the method of support vector machines (e.g., [1, 2]) which is one of the best known technique in supervised classification. We consider their application to agglomerative hierarchical clustering. We should note that positive-definite kernels themselves are independent topic from support vector machines, and indeed no reference to support vector machines is needed herein.

We also note that positive-definite kernels are already discussed in K-means [3], but not popular in agglomerative hierarchical clustering. It is obvious that the single linkage, complete linkage, and average linkage do not need the consideration of positive-definite kernels, since any symmetric matrix of similarity or dissimilarity can be used for them. In contrast, the centroid method and Ward method are based on the squared Euclidean distance and hence the present consideration is needed. In other words, the discussion in this chapter is solely for the latter two linkage methods.

Hence the problem to be addressed is whether the centroid method and/or Ward method can be applied to a case when a similarity or dissimilarity is available, instead of a set of points in an Euclidean space. The answer is that if the similarity matrix is positive-definite, they can be applied.

Preliminary considerations are needed before discussing the main subject of this chapter.

4.1 Positive-Definite Kernels

The application of kernels to data classification assumes a mapping from data space to a *high-dimensional feature space*. Here the data space includes the set of objects $X = \{x_1, \ldots, x_N\}$ which is not a vector space in general, and a high-dimensional space H is assumed. Generally, H is an abstract vector space but we will give a specific form to it without loss of generality.

S. Miyamoto, *Theory of Agglomerative Hierarchical Clustering*,
Behaviormetrics: Quantitative Approaches to Human Behavior 15,
https://doi.org/10.1007/978-981-19-0420-2_4

In order to discuss specific positive-definite kernels herein, we need notations and definitions for this chapter.

An n-dimensional real vector space $\boldsymbol{R}^n$ is a collection of vectors $\boldsymbol{z} = (z_1, z_2, \ldots, z_n)^\top$ of which components are real values. For all vectors $\boldsymbol{z}$ and $\boldsymbol{z}'$, addition $\boldsymbol{z} + \boldsymbol{z}'$ and scalar multiplication $c\boldsymbol{z}$ (c is an arbitrary real number) are defined, but details are omitted here.

The zero vector is denoted by $\mathbf{0} = (0, 0, \ldots, 0)^\top$, and elementary vectors are denoted by $\boldsymbol{e}_1 = (1, 0, \ldots, 0)^\top$, $\boldsymbol{e}_2 = (0, 1, 0, \ldots, 0)^\top$, $\ldots$, $\boldsymbol{e}_n = (0, 0, \ldots, 1)^\top$. We also introduce $\mathbf{1} = (1, 1, \ldots, 1)^\top$. For the most part $n = N$, the number of objects in X. Hence we can calculate $S\boldsymbol{z}$ or $\boldsymbol{z}^\top S$, where S is a $N \times N$ similarity matrix.

We consider so far an M-dimensional Euclidean space to be a vector space with the standard Euclidean norm and inner product:

$$\|\boldsymbol{z}\| = \sqrt{z_1^2 + z_2^2 + \cdots + z_n^2}, \tag{4.1}$$

$$(\boldsymbol{z}, \boldsymbol{z}') = \boldsymbol{z}^\top \boldsymbol{z}' = \boldsymbol{z}'^\top \boldsymbol{z} = z_1 z_1' + \cdots + z_n z_n'. \tag{4.2}$$

However, Euclidean spaces in a generalized sense are used in this chapter, i.e., a Euclidean space H is a vector space in which an inner product is defined.

An *inner product* denoted by $\langle \cdot, \cdot \rangle$ here is a real-valued function of two elements in H ($\langle \cdot, \cdot \rangle : H \to \boldsymbol{R}$) which satisfies the following:

(i) $\langle \boldsymbol{z}, \boldsymbol{z} \rangle \geq 0$ and $\langle \boldsymbol{z}, \boldsymbol{z} \rangle = 0 \iff \boldsymbol{z} = \mathbf{0}$,
(ii) $\langle \boldsymbol{z}, \boldsymbol{z}' \rangle = \langle \boldsymbol{z}', \boldsymbol{z} \rangle$,
(iii) $\langle c_1 \boldsymbol{z}_1 + c_2 \boldsymbol{z}_2, \boldsymbol{z}' \rangle = c_1 \langle \boldsymbol{z}_1, \boldsymbol{z}' \rangle + c_2 \langle \boldsymbol{z}_2, \boldsymbol{z}' \rangle$.

Generally, H need not be finite-dimensional but we assume hereafter all spaces are finite-dimensional since we don't need an infinite-dimensional space for the present discussion.

Note that a symmetric matrix is called *positive-definite* if and only if all of its eigenvalues are positive. Moreover, a symmetric matrix is called *positive-semidefinite* or *nonnegative-definite* if and only if all of its eigenvalues are nonnegative.

Let us consider a condition that $\boldsymbol{z}^\top S \boldsymbol{z}'$ is an inner product.

Proposition 10 *Let S be an $n \times n$ real symmetric matrix. For arbitrary $\boldsymbol{z}, \boldsymbol{z}' \in \boldsymbol{R}^n$, $\langle \boldsymbol{z}, \boldsymbol{z}' \rangle = \boldsymbol{z}^\top S \boldsymbol{z}'$ is an inner product if and only if S is positive-definite.*

Proof Standard textbooks show the result that a positive-definite matrix S satisfies $\boldsymbol{z}^\top S \boldsymbol{z} \geq 0$ for every $\boldsymbol{z} \in \boldsymbol{R}^n$ and $\boldsymbol{z}^\top S \boldsymbol{z} = 0$ if and only if $\boldsymbol{z} = \mathbf{0}$. The above (ii) and (iii) are trivial and omitted. □

Note 9 Similarity matrix S rather than dissimilarity D is convenient for the present discussion. If we are given a dissimilarity measure, we can transform it into similarity, as we discussed in Chap. 1.

4.2 Linkage Methods Using Kernels

The above discussion is sufficient for our purpose and we proceed to consider clustering. Assume that we are given an object set $X = \{x_1, \ldots, x_N\}$ with similarity matrix S. X is not a vector space in general, but our motivation is to apply the centroid method and Ward method which are based on the squared Euclidean distance. The main result in this chapter is the following.

If matrix S is positive-definite, we can apply the centroid method and Ward method.

We show how the above statement is realized. Apart from the subject of clustering, the theory of kernel classification uses a *high-dimensional mapping* $\phi(x)\colon X \to H$ (see, e.g., [1]) and $\phi(x_i) = \boldsymbol{y}_i$. We put $H = \boldsymbol{R}^N$, and in particular we take

$$\phi(x_i) = \boldsymbol{y}_i = \boldsymbol{e}_i, \quad i = 1, 2, \ldots, N. \tag{4.3}$$

Hence

$$\langle \phi(x_i), \phi(x_j) \rangle = \boldsymbol{e}_i^\top S \boldsymbol{e}_j = s_{ij}. \tag{4.4}$$

We also introduce the norm of H:

$$\|z\|_S^2 = \langle z, z \rangle = z^\top S z. \tag{4.5}$$

The point of application of the positive-definite kernel to the two linkage method is simple by using the following definitions.

Kernel centroid method:
We take

$$M_H(G_i) = \frac{1}{|G_i|} \sum_{x_l \in G_i} \phi(x_l), \tag{4.6}$$

where $|G_i|$ is the number of elements in G_i, and define

$$D_H(G_i, G_j) = \|M_H(G_i) - M_H(G_j)\|_S^2. \tag{4.7}$$

Kernel Ward method:
We take

$$E_H(G_i) = \sum_{x_l \in G_i} \|\phi(x_l) - M_H(G_i)\|_S^2, \tag{4.8}$$

and define

$$D_H(G_i, G_j) = \Delta E_H(G_i, G_j) = E_H(G_i \cup G_j) - E_H(G_i) - E_H(G_j). \tag{4.9}$$

Note that we can compute everything in the above equations of definitions, which is very different from the discussion of the support vector machines where we cannot

compute $\phi(x)$ directly. Nonetheless, the points of computations are essentially the same in both the consideration of classification and clustering.

To summarize, the *initial values* and updating formulas when kernel S is used are as follows.

Proposition 11 *The initial value for the kernel centroid method is given by*

$$D_H(x_i, x_j) = s_{ii} + s_{jj} - 2s_{ij}; \tag{4.10}$$

the initial value for the kernel Ward method is given by

$$D_H(x_i, x_j) = \frac{1}{2}(s_{ii} + s_{jj} - 2s_{ij}). \tag{4.11}$$

Proof The proof is immediate, since

$$\begin{aligned} D_H(x_i, x_j) &= \|\phi(x_i) - \phi(x_j)\|_S^2 \\ &= \boldsymbol{e}_i S\boldsymbol{e}_i + \boldsymbol{e}_j S\boldsymbol{e}_j - 2\boldsymbol{e}_i S\boldsymbol{e}_j = s_{ii} + s_{jj} - 2s_{ij} \end{aligned}$$

for the centroid method, and

$$D_H(x_i, x_j) = \frac{1}{2}\|\phi(x_i) - \phi(x_j)\|_S^2$$

for Ward method. □

Proposition 12 *The updating formulas for the kernel centroid method and the kernel Ward method are respectively given by (2.20) and (2.26). That is, they are the same as the standard formulas without a kernel.*

Specifically, they are as follows.

Kernel centroid method:

$$D_H(G_r, G_j) = \frac{|G_p|}{|G_r|} D_H(G_p, G_j) + \frac{|G_q|}{|G_r|} D_H(G_q, G_j) - \frac{|G_p||G_q|}{|G_r|^2} D_H(G_p, G_q), \tag{4.12}$$

Kernel Ward method:

$$\begin{aligned} D_H(G_r, G_j) = \frac{1}{|G_r| + |G_j|} \{ &(|G_p| + |G_j|) D_H(G_p, G_j) \\ &+ (|G_q| + |G_j|) D_H(G_q, G_j) - |G_j| D_H(G_p, G_q) \}, \end{aligned} \tag{4.13}$$

where $|G_r| = |G_p| + |G_q|$.

The proof of the formulas (4.12) and (4.13) are essentially the same as those for (2.20) and (2.26) and omitted.

To summarize, the centroid method and Ward method can be used for positive-definite similarity matrix with the same updating formulas as the standard ones, except that the initial values should be set to (4.10) and (4.11), respectively.

A question is whether or not a similarity matrix with a negative eigenvalue is acceptable in this case. The support vector machines sometimes use nonpositive-definite (alias indefinite) kernels, although indefinite kernels are theoretically unacceptable in general, and hence the case of an indefinite matrix for the centroid method and Ward method should be an *ad hoc* technique.

4.3 Indefinite Similarity Matrix for Ward Method

As noted above, the centroid method and Ward method for *indefinite similarity matrices* should be regarded as *ad hoc* methods: they are not based on the sound mathematical model of the Euclidean space. However, a further result is available for Ward method, which we discuss in this section. For discussing this, Ward method applied to indefinite matrix is called *ad hoc* Ward method and that applied to positive-definite matrix is called *real* Ward method.

Note also that for a given indefinite similarity matrix S, there is a positive real value β such that $S + \beta I$ is positive-definite, since the following proposition holds.

Proposition 13 *For an arbitrary symmetric $N \times N$ matrix S, the eigenvalues of S denoted by $\lambda_1, \ldots, \lambda_N$ are real values. Moreover, the eigenvalues of $S + \beta I$ are $\lambda_1 + \beta, \ldots, \lambda_N + \beta$.*

Proof The proof of the former half is written in standard textbooks of matrix theory. For the proof of the latter half, suppose $S = U^\top \Lambda U$ where Λ is the diagonal matrix: $\Lambda = \mathrm{diag}[\lambda_1, \ldots, \lambda_N]$, and U is an orthogonal matrix. Note also that standard textbooks of matrix theory state that such a decomposition is possible for every symmetric matrix.

Then it is easy to see that we have

$$S + \beta I = U^\top (\Lambda + \beta I) U.$$

which implies that the eigenvalues of $S + \beta I$ are $\lambda_1 + \beta, \ldots, \lambda_N + \beta$. □

We now state the following result.

The dendrogram produced by ad hoc Ward method applied to S and the dendrogram produced by real Ward method applied to $S + \beta I$ are similar.

Note that the ward *similar* has already been defined in the last section of Chap. 2. Moreover, the similarity here is stronger than the general definition. Let us therefore describe the result in more detail.

First, what we assume is a set of objects $X = \{x_1, \ldots, x_N\}$ as usual and a similarity matrix S. S generally may have a negative eigenvalue alias indefinite. We assume moreover that diagonal values are unity: $s_{ii} = 1$ for $i = 1, \ldots, N$ and $s_{ij} \leq 1$ for $1 \leq$

$i, j \leq N$. This assumption holds for most similarity matrices used in applications. Note that we do not assume a dissimilarity matrix; when we are given a dissimilarity matrix, we can transform it into a similarity matrix, as we discussed in Chap. 1.

The *ad hoc* Ward method thus use S regardless whether it is positive-definite or not. The dissimilarity for the *ad hoc* method is denoted by $D'(G, G')$ here. By applying the above assumption $s_{ii} = 1$ to (4.11), the initial value is given by

$$D'(x_i, x_j) = 1 - s_{ij}, \quad 1 \leq i, j \leq N, \tag{4.14}$$

and the updating equation having the same form as (4.13) is used:

$$D'(G_r, G_j) = \frac{1}{|G_r| + |G_j|}\{(|G_p| + |G_j|)D'(G_p, G_j) + (|G_q| + |G_j|)D'(G_q, G_j) - |G_j|D'(G_p, G_q)\}, \tag{4.15}$$

$(|G_r| = |G_p| + |G_q|)$.

We now *regularize* the matrix S by adding β to the diagonal elements: We thus use $S_\beta = S + \beta I$ instead of S. The parameter β is chosen so that all the eigenvalues are positive. In this case we are using the *real* Ward method having the squared Euclidean distance as mentioned above. In the latter case we use the symbol $D_H(G, G')$ defined above. The initial values for the latter method are

$$D_H(x_i, x_j) = 1 + \beta - s_{ij}, \quad 1 \leq i, j \leq N, \tag{4.16}$$

as the diagonal elements are $1 + \beta$ and off-diagonal elements are s_{ij}. The updating equation is given by (4.13).

We have the next theorem that states both methods give equivalent results except the merging levels. Note that we use the same tie breaking rule for both methods.

Theorem 12 *Let $\mathcal{G}'(k)$ $(k = N, N-1, \ldots, 1)$ be clusters formed at level L'_k by the ad hoc Ward method with similarity S, and $\mathcal{G}^\beta(k)$ $(k = N, N-1, \ldots, 1)$ be clusters formed at level L^β_k by the real Ward method with similarity $S + \beta I$. Then*

$$\mathcal{G}^\beta(k) = \mathcal{G}'(k), \quad k = N, N-1, \ldots, 1, \tag{4.17}$$

and

$$L^\beta_k = L'_k + \beta, \quad k = N, N-1, \ldots, 1, \tag{4.18}$$

hold.

Proof Note first the following holds:

$$D_H(x_i, x_j) = D'(x_i, x_j) + \beta, \quad 1 \leq i, j \leq N, \ i \neq j. \tag{4.19}$$

Next, suppose that $\mathcal{G}^\beta(k) = \mathcal{G}'(k)$ holds at a particular k. We suppose moreover that

$$D_H(G, G') = D'(G, G') + \beta \tag{4.20}$$

holds for all clusters G, G'. Note that these two equations are valid for $k = N$. Then the same pair of G_p and G_q attains the minimum of the distances for all cluster pairs and they are merged: $G_r = G_p \cup G_q$ in **AHC** algorithm. The same updating formula of (4.15) and (4.13) are applied to the *ad hoc* Ward method and the real Ward method with $S + \beta I$. Putting (4.20) into these two updating equations, we have

$$D_H(G_r, G_j) = D'(G_r, G_j) + \beta, \tag{4.21}$$

which means that (4.20) holds for $k - 1$. Thus $\mathcal{G}^{\beta}(k) = \mathcal{G}'(k)$ holds with $L_k^{\beta} = L'_k + \beta$ for $k = N, N - 1, \ldots, 2, 1$ and hence the theorem is proved. □

This theorem thus shows that the *ad hoc* Ward method produces the same dendrogram as that of the real Ward method with the *regularized matrix* $S + \beta I$ except the merging levels are augmented by $\beta > 0$.

Note that such a property does not hold for the centroid method. Hence if we use the centroid method for indefinite similarity matrix, we cannot refer to the original meaning of the centroids in an Euclidean space.

An Illustrative Example

Let us see a simple example that illustrates the above theory. The similarity matrix $S = (s_{ij})$ was generated by random numbers in the unit interval: $s_{ij} \in [0, 1]$ $(1 \leq$

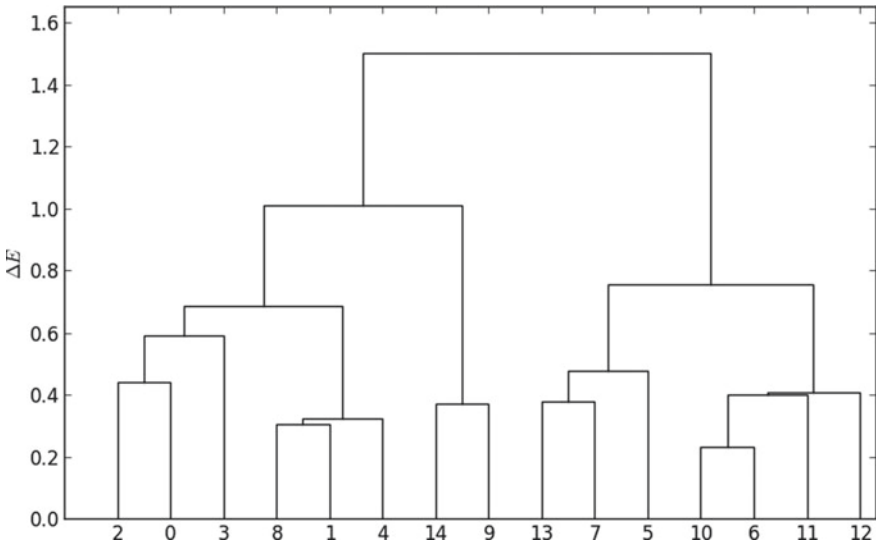

Fig. 4.1 Dendrogram using the *ad hoc* Ward method for a randomly generated similarity matrix S with the minimum eigenvalue -0.26

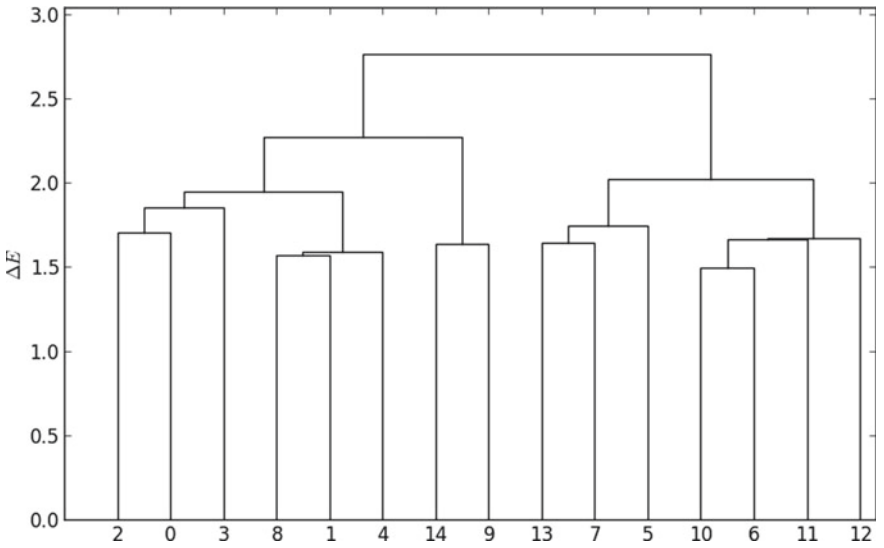

Fig. 4.2 Dendrogram using the real Ward method for $S + \beta I$ where S is the last similarity matrix and $\beta = 1.26$ is chosen so that all eigenvalues of S are positive

$i, j \leq N, i \neq j$) except the diagonal elements $s_{ii} = 1$ ($1 \leq i \leq N$). Actually, we had negative eigenvalues of S: the minimum eigenvalue is -0.26. Figure 4.1 shows the resulting dendrogram using the *ad hoc* Ward method. On the other hand, Fig. 4.2 is the dendrogram using $S + \beta I$, where $\beta = 1.26$ so that all eigenvalues are positive.

As readers see, both dendrograms are the same except the length from the bottom to the first merging level, showing the validity of the above theory of Ward method for indefinite similarity matrices.

References

1. V.N. Vapnik, *Statistical Learning Theory* (Wiley, New York, 1998)
2. B. Schölkopf, A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* (MIT Press, Cambridge, Massachusetts, 2001)
3. M. Girolami, Mercer kernel based clustering in feature space. IEEE Trans. Neural Netw. **13**(3), 780–784 (2002)

Chapter 5
Some Other Topics in Agglomerative Hierarchical Clustering

Several other topics in agglomerative hierarchical clustering studied by the author and his colleagues are described in this chapter. They are as follows.

1. Two algorithms related to the single linkage are described. They use the idea of qualified objects.
2. A relation between clustering and supervised classification is focused upon, which leads to the idea of two-stage agglomerative hierarchical clustering.
3. Agglomerative algorithms with constraints are studied.
4. The idea of model-based clustering is used in agglomerative hierarchical clustering.
5. An algorithm of agglomerative hierarchical clustering using an asymmetric similarity measure is proposed.

5.1 Single Linkage, DBSCAN, and Mode Analysis

DBSCAN [1] is a well-known method, while *mode analysis* by Wishart [2] is now unknown to most researchers and users of clustering. However, both the methods have a close relation with the single linkage. The relation has already been discussed at the last part of Chap. 3 concerning the theory of the single linkage.

In this section we observe in what way these two methods are related to the single linkage. For this purpose we first review DBSCAN algorithm.

S. Miyamoto, *Theory of Agglomerative Hierarchical Clustering*,
Behaviormetrics: Quantitative Approaches to Human Behavior 15,
https://doi.org/10.1007/978-981-19-0420-2_5

5.1.1 DBSCAN and the Single Linkage

This algorithm first defines an *Eps*-neighborhood $N_{Eps}(x)$ for each $x \in X$ using a positive real number $Eps > 0$:

$$N_{Eps}(x) = \{\, z \in X : \; D(z, x) \leq Eps \,\}. \tag{5.1}$$

Thus $N_{Eps}(x)$ is a subset of X.

Another parameter is MinPts, a natural number qualifying a core point. A point $x \in X$ is defined to be a *core point* if it satisfies

$$|N_{Eps}(x)| \geq \text{MinPts}. \tag{5.2}$$

A point $y \in X$ is called *directly density-reachable* from x if x is a core point and $y \in N_{Eps}(x)$.

Next, a point $z \in X$ is called *density-reachable* from x if there is a chain of points $y_1, \ldots, y_\ell$ with $y_1 = x$ and $y_\ell = z$ such that y_{i+1} is directly density-reachable from y_i $(i = 1, \ldots, \ell - 1)$. Note that the last point z of the chain may or may not be a core point, while other points in the chain are core points.

Two points x, y are called *density-connected* if there exists $z \in X$ such that x is density-reachable from z and also y is density-reachable from z. Note that x and y may or may not be core points.

A cluster in DBSCAN is defined as follows.

Definition 10 A cluster C in DBSCAN is a subset of X satisfying the following two conditions.

- If $x \in C$ and y is density-reachable from x, then $y \in C$.
- For $\forall x, y \in C$, x and y are density-connected.

The algorithm of DBSCAN starts from a core point x and searches all density-reachable points and make a cluster $C(x)$. It includes core points and noncore points as end points of a chain in the definition of density-reachable points. Then another core point x' that is not in clusters found so far is selected and the same search is performed until no such cluster is found.

Algorithm for DBSCAN.
Step 0. Let the collection of clusters be $\mathcal{C}$. Put $\mathcal{C} = \emptyset$. Assume core(X) be the set of core points of X.
Step 1. If core$(X) = \emptyset$, go to **Step 4**, else take $x_i \in$ core(X) and core$(X) =$ core$(X) - \{x_i\}$.
Step 2. Find all points that is density-reachable from x_i and let C_i be the set of the density-reachable points and also x_i itself. $X = X - C_i$.
Step 3. Update $\mathcal{C} = \mathcal{C} \cup \{C_i\}$. Go to **Step 1**.
Step 4. Output $\mathcal{C}$ and stop.
End DBSCAN.

Note 10 Schubert et al. [3] describe two algorithms of DBSCAN called **Abstract DBSCAN Algorithm** and **Original Sequential DBSCAN Algorithm**. The above **Algorithm for DBSCAN** is similar to the former, the abstract algorithm. The latter sequential algorithm is more complicated, including an advanced tree data structure to reduce the algorithm complexity.

Let clusters found by this DBSCAN algorithm be $C_1, \ldots, C_K$. Obviously, we have

$$\bigcup_{j=1}^{K} C_j \subseteq X, \qquad C_j \cap C_\ell = \emptyset \ \ (j \neq \ell), \tag{5.3}$$

and generally, $\bigcup_{j=1}^{K} C_j \neq X$. Thus the definition of clusters by DBSCAN is different from the definition by (1.1) in Chap. 1.

Moreover, note $\bigcup_{j=1}^{K} C_j \supseteq \mathrm{core}(X)$ and generally $\bigcup_{j=1}^{K} C_j \neq \mathrm{core}(X)$.

In Chap. 3, we considered $\mathbf{MST}(\alpha)$ as clusters obtained from the cut of the maximum spanning tree at α, which coincide with clusters found by the single linkage. Let us moreover assume that $\mathbf{MST}(d)$ is the set of clusters from the cut of the *minimum spanning tree* at the distance d.

Assume also that the *minimum* spanning tree algorithm $\mathbf{MST}(d)$ is applied to $\mathrm{core}(X)$, the set of core points. The obtained clusters are supposed to be $G_i, \ldots, G_{K'}$. Obviously,

$$\bigcup_{j=1}^{K'} G_j = \mathrm{core}(X), \qquad G_j \cap G_\ell = \emptyset \ \ (j \neq \ell). \tag{5.4}$$

This method is hereafter called DBSCAN-CORE [4]: it is closely related to DBSCAN, but the difference between these two is that DBSCAN-CORE includes only core points, while DBSCAN includes both noncore points and core points in clusters. More precisely, we have the following.

Proposition 14 *Clusters obtained from* $\mathbf{MST}(d)$ *is the refinement of those from DBSCAN, i.e.,* $K = K'$ *and for any* G_i*, there exists* C_{j_i} *such that* $G_i \subseteq C_{j_i}$ $(i = 1, \ldots, K)$*. Moreover,* $C_{j_i} - G_i$ *includes only noncore points* $(i = 1, \ldots, K)$*.*

The proof of this proposition is not difficult but omitted here (see [4]). This proposition shows that DBSCAN is in a sense a variation of the single linkage.

5.1.2 Mode Analysis and the Single Linkage

Unlike DBSCAN, mode analysis algorithm by Wishart [2] is hierarchical, and Wishart states that this algorithm is a generalization of the nearest neighbor clustering (alias the single linkage). He proposes both nonhierarchical and hierarchical versions of the algorithm and we review the hierarchical algorithm that is suited for our pur-

pose. His algorithm uses a distance but we use a similarity $S(x, y)$ to emphasize that his algorithm can be used both for similarity and dissimilarity measures.

Before stating Wishart's algorithm, the concept of a *dense point* is introduced. For each point $x_j \in X$ and a positive number k, $y(x_j; k)$ means the kth nearest point to x_j: in terms of a similarity measure, $y(x_j; k)$ means the point having kth largest similarity value to x_j. Assume that sorting $S(x_j, y(x_j; k))$ into the decreasing order, we have

$$S(x_{j_1}, y(x_{j_1}; k)) \geq S(x_{j_2}, y(x_{j_2}; k)) \geq \cdots \geq S(x_{j_N}, y(x_{j_N}; k)). \tag{5.5}$$

Thus the first dense point is x_{j_1}, second is x_{j_2}, and so on. The next algorithm is due to Wishart: the description is simplified from the original one in [2].

Mode Analysis (Wishart [2]).

1. For $l = 1, \ldots, N$ repeat steps 2–3.
2. Set *threshold* $= S(x_{j_l}, y(x_{j_l}; k))$. Make x_{j_l} a new dense point.
3. Find every dense point z with $S(z, x_{j_l}) \geq$ *threshold*, connect x_{j_l} with the corresponding cluster to which z belongs to. If not, make x_{j_l} a new cluster of $\{x_{j_l}\}$.

End of Mode Analysis.

It is clear that all clusters up to *threshold* consist of dense points and they are connected by the nearest neighbor rule using the neighborhood of

$$\{z \ : \ S(z, x_{j_l}) \geq S(x_{j_l}, y(x_{j_l}; k)) \ \}. \tag{5.6}$$

Note also that Wishart's method uses *k-nearest neighbor*, while DBSCAN uses a fixed radius neighbor. Thus each point x_j has a different size of neighborhood in Wishart's method. It is now easy to see that Wishart's mode analysis is the single linkage method with fuzzy graph (X, S, A) defined in Sect. 3.6.

Let us consider again the example in Sect. 3.6.

Example 11 Consider again the graph in Fig. 3.1, where A is defined by the second largest value of the edges from each node. In terms of Wishart's method, this means $k = 2$. Thus,

- for node A, we have three edges with weights $\{0.9, \underline{0.5}, 0\}$;
- for node B, we have edges with $\{0.5, \underline{0.4}, 0.3\}$;
- for node C, we have edges with $\{0.7, \underline{0.3}, 0\}$;
- for node D, we have edges with $\{0.9, \underline{0.7}, 0.4\}$.

Each node becomes dense at the underlined value of similarity.

By the above algorithm for mode analysis, the dense points are introduced by the following order:

$$\text{D at } 0.7; \quad \text{A at } 0.5; \quad \text{B at } 0.4; \quad \text{C at } 0.3.$$

Table 5.1 Example showing application of Wishart's algorithm to the graph in Fig. 3.1 with $k = 2$

Level	Dense points	Clusters
0.7	D	
0.5	D, A	$\{A, D\}$
0.4	D, A, B	$\{A, B, D\}$
0.3	D, A, B, C	$\{A, B, C, D\}$

Accordingly, clusters are formed as in Table 5.1. This result clearly coincides with that of the example in Sect. 3.6, except that the cluster of the single nondense point is not recognized as a cluster in this algorithm.

Note 11 Wishart's method can be algebraically represented using the concept of the transitive closure. Define $\boldsymbol{a} = (a_1, \ldots, a_N)^\top$ as the vector for fuzzy set A in (X, S, A). Let us note that

$$(\boldsymbol{a} \circ \boldsymbol{a}^\top)(x, x') = \min\{A(x), A(x')\}, \tag{5.7}$$

where $A(x)$ is the membership value at node x. Then clusters by Wishart's method is given by

$$(S \wedge (\boldsymbol{a} \circ \boldsymbol{a}^\top))^*, \tag{5.8}$$

which is symmetric and transitive, but not reflexive. Since $S \wedge (\boldsymbol{a} \circ \boldsymbol{a}^\top)$ is not reflexive, nondense points are not recognized as clusters, but we do not have any real problem in clustering. Its relation to S' is

$$S' = S \wedge (\boldsymbol{a} \circ \boldsymbol{a}^\top) \vee I, \tag{5.9}$$

where I is the identity matrix ($I = \mathrm{diag}(1, 1, \ldots, 1)$). For the above example, $\boldsymbol{a} = (0.5, 0.4, 0.3, 0.7)^\top$, and

$$\boldsymbol{a} \circ \boldsymbol{a}^\top = \begin{pmatrix} 0.5 & 0.4 & 0.3 & 0.5 \\ 0.4 & 0.4 & 0.3 & 0.4 \\ 0.3 & 0.3 & 0.3 & 0.3 \\ 0.5 & 0.4 & 0.3 & 0.7 \end{pmatrix}, \tag{5.10}$$

$$S \wedge (\boldsymbol{a} \circ \boldsymbol{a}^\top) = \begin{pmatrix} 0.5 & 0.4 & 0 & 0.5 \\ 0.4 & 0.4 & 0.3 & 0.4 \\ 0 & 0.3 & 0.3 & 0.3 \\ 0.5 & 0.4 & 0.3 & 0.7 \end{pmatrix}, \tag{5.11}$$

and we have

$$(S \wedge (\boldsymbol{a} \circ \boldsymbol{a}^{\top}))^{*} = \begin{pmatrix} 0.5 & 0.4 & 0.3 & 0.5 \\ 0.4 & 0.4 & 0.3 & 0.4 \\ 0.3 & 0.3 & 0.3 & 0.3 \\ 0.5 & 0.4 & 0.3 & 0.7 \end{pmatrix}. \tag{5.12}$$

Note 12 The reason why we emphasized the relation of these two algorithms with the single linkage is that the single linkage with some *qualified points* is a productive idea. This idea was already used in at least two well-known algorithms and should further be studied. A study in the author's research group [4] proposes a variation of DBSCAN in which objects are limited to core points and noncore points are allocated to clusters by the nearest neighbor allocation rule.

5.2 Clustering and Classification

Data clustering, in general including both hierarchical and nonhierarchical algorithms, is representative in unsupervised classification, while the other topic of supervised classification is also well-known and many methods for the latter have been developed. The section title of 'clustering and classification' implies the fields of clustering and supervised classification; it moreover implies that both are related in some way.

Two typical problems where clustering and classification are related are as follows:

(i) A part of objects is already classified and the problem is to classify or cluster all other objects.
(ii) Existing objects have been made into clusters using a certain method of clustering, and a new object or a new set of objects arrives after that and should be clustered.

It appears that both the problems are similar, and some classification technique can be used for the both. However, we note that the problems are essentially different.

In many cases the first problem can be solved by using a technique of classification; a typical classification method is the *support vector machine* (SVM) [5]. Sometimes a small number of objects are already classified, and an ordinary SVM cannot be used. In such a case methods of *semi-supervised learning* [6, 7] are applicable.

If we want to use agglomerative hierarchical clustering to the first problem, a natural method is to assume initial clusters to be given classes: $\mathcal{G}(N_0) = \{G_1, \ldots, G_{N_0}\}$ in **AHC** algorithm in Chap. 2. Another technique of *constrained clustering* is described in the next section. Overall, however, the application of agglomerative hierarchical clustering to the first problem is exceptional, since many methods of supervised classification are available, which are incompatible with clustering techniques.

The second problem is that of clustering. This problem is not widely studied, at least in agglomerative hierarchical clustering. In contrast, incremental clustering algorithms have been developed for K-means (see, e.g., [8, 9]), where a new point is added and cluster centers are accordingly modified by a simple algorithm.

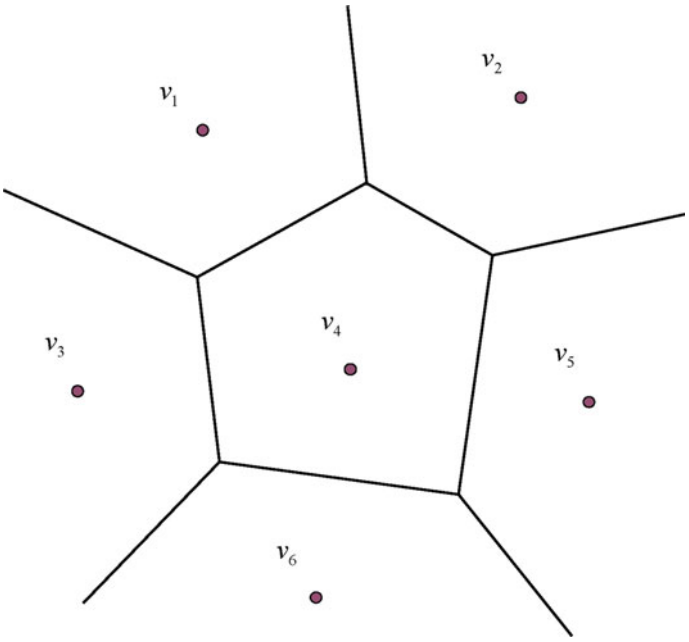

Fig. 5.1 An example of Voronoi regions on a plane with centers $v_1, \ldots, v_6$

Let us consider for the moment the simple K-means algorithm **KM** in Chap. 1. Suppose that clusters $G_1, \ldots, G_K$ with cluster centers $v_1, \ldots, v_K$ are already formed and a new object x arrives. Which cluster x should be allocated is clear, as the rule in **KM2** shows: if

$$i = \arg \min_{1 \le j \le K} \|x - v_i\|^2, \tag{5.13}$$

then cluster G_i should be selected. In other words, if cluster prototype v_i is nearest to x, the corresponding cluster is selected: this rule is called the *nearest prototype* rule.

This rule is simple and generic in the sense that it is applied to all $x \in \boldsymbol{R}^p$. Thus, the space $\boldsymbol{R}^p$ can be divided into partitions

$$P(v_i, V) = \{x \in \boldsymbol{R}^p : \|x - v_i\| < \|x - v_j\|,\ 1 \le j \le K,\ j \ne i\} \tag{5.14}$$

for $V = (v_1, \ldots, v_K)$. The regions $P(v_1, V), \ldots, P(v_K, V)$ are called *Voronoi regions* with centers $v_1, \ldots, v_K$ [9]. For plane $\boldsymbol{R}^2$, these regions are simply illustrated as Fig. 5.1, where equidistant line segments divide the plane. Thus the method of K-means has the intrinsic classification rule of the nearest prototype.

Let us consider another simple method of classification, i.e., the nearest neighbor method. Given a predefined set of clusters $G_1, \ldots, G_K$ such that

$$G_i \cap G_j = \emptyset \quad (i \neq j), \tag{5.15}$$

and a generic $x \notin \bigcup_{j=1}^{K} G_j$, the nearest neighbor rule allocates x to G_i:

$$x \to G_i \iff i = \arg \min_{1 \leq j \leq K} D(x, G_j), \tag{5.16}$$

where

$$D(x, G_j) = \min_{y \in G_j} D(x, y). \tag{5.17}$$

Although the nearest neighbor rule can be applied to general dissimilarity (and also similarity), let us suppose for the moment that the space $\boldsymbol{R}^p$ is assumed: $G_1, \ldots, G_K$ are set of points in this space and an arbitrary $x \in \boldsymbol{R}^p$ is considered. The nearest neighbor method is then related to Voronoi regions again. Let $G' = \bigcup_{j=1}^{K} G_j$ and consider $P(x_i, G')$ for $x_i \in G'$.

Then the nearest neighbor classification rule is rewritten using $P(x_i, G')$:

$$x \to G_j \iff x \in \bigcup_{x_i \in G_j} P(x_i, G'). \tag{5.18}$$

The nearest prototype rule and the nearest neighbor rule thus have different shapes of the regions of allocation: the former Voronoi region is a convex polygon or hyper-polygon, while the latter is a union of smaller Voronoi regions having a more general shape, sometimes nonconvex and/or prolonged.

We have observed two simple classification rules, but are they really related to methods of agglomerative hierarchical clustering? A trivial answer may be negative: a classification rule and a linkage method has no relation. We, however, try to answer this question in an affirmative way.

For this purpose we take two methods: nonhierarchical method of the K-means and nonhierarchical single linkage, i.e., **MST**(α) of the minimum spanning tree algorithm. As noted earlier, the K-means has the nearest prototype classification rule, and the minimum spanning tree has the nearest neighbor rule. Hence clusters formed by these two have different shapes: the former is a convex polygon, while the latter may be prolong and/or nonconvex.

These two nonhierarchical methods are related to the well-known agglomerative hierarchical algorithms: the K-means to Ward method and the minimum spanning tree to the single linkage. The relation between the minimum spanning tree and the single linkage has already been shown, while the function to be minimized in the K-means is common with that of Ward method.

In relation to the above problem (ii), after Ward method or the single linkage method is used for clustering, a new object can be allocated to the cluster of the nearest prototype or the nearest neighbor, respectively.

For the single linkage, we can develop a promising algorithm using this idea. First we use DBSCAN-CORE in the previous section and then unclassified noise points are allocated using NN (the nearest neighbor method of classification). Then all points are allocated to a certain cluster. This clustering algorithm, which is not hierarchical, is called DBSCAN-CORE-NN [4].

5.2.1 Two-Stage Clustering

Two-stage clustering herein means that a nonhierarchical clustering is performed first and those clusters are put as initial clusters in agglomerative clustering algorithm: A part of $\mathcal{G}(N_0) = \{G_1, \ldots, G_{N_0}\}$ in **AHC** is initial clusters and others are points unclassified in the first stage.

Suppose that we have too many objects for agglomerative hierarchical clustering, e.g., thousands of them, and yet we need a dendrogram output. In such a case two-stage clustering may be useful.

Although the choices of the first-stage algorithm and the linkage method in the second stage seem rather arbitrary, we consider two types *consistent* choices: the first type is a variation of the single linkage in the first stage and the single linkage itself in the second stage; the second type is K-means and Ward method.

Apparently the discussion in the previous section works, but we add some more remarks. For the single linkage, a typical example is to take DBSCAN-CORE as the first stage algorithm. Then the rest of the unclassified points is allocated using the single linkage in the second stage.

For the combination of K-means and Ward method, a medium number, e.g., $N_0 = 100$ of initial clusters $\mathcal{G}(N_0)$ should be first generated. In K-means algorithm, initial cluster centers should be selected, and N_0 objects are randomly selected for the centers. Frequently N_0 objects are not well-scattered (i.e., concentrated somewhere in the object space) and resulting clusters are inadequate as $\mathcal{G}(N_0)$. An advanced algorithm of K-means++ [10] is available for solving this problem in which initial cluster centers are systematically scattered by calculating dissimilarity values among the centers. Tamura and Miyamoto [11] shows a variation of *K-means++* in which the number of iteration limited to two is effective in the two-stage algorithm of K-means and Ward method.

We moreover considered a *kernel Ward method* in the previous chapter. The methods of *kernel K-means* method and also *kernel K-means++* can be developed without difficulty by considering $\phi(x_i)$ instead of x_i and putting

$$\|\phi(x_i) - \phi(x_j)\|^2 = S(x_i, x_i) + S(x_j, x_j) - 2\,S(x_i, x_j). \tag{5.19}$$

Note that matrix $S(x_i, x_j)$ should be positive-definite.

5.3 Constrained Agglomerative Hierarchical Clustering

The topic of semi-supervised learning has been popular in the field of machine learning [6, 7]. In data clustering, *constrained clustering* (see, e.g., [12]) has been studied as a particular type of semi-supervised learning. Most literature in constrained clustering is concentrated on nonhierarchical clustering, but there are some studies in *constrained agglomerative hierarchical clustering* (e.g., [13]). We briefly review the idea of constrained agglomerative clustering in this section.

Semi-supervised learning in general implies that a relatively small number of objects have predefined class labels, while other large number of objects are unclassified and each of them should be allocated to one of the classes.

On the other hand, constrained clustering means that a *must-link* set ML and a *cannot-link* set CL are defined in the object set X. They are graphical concept: ML and CL are subsets of $X \times X$ such that links are symmetric: if $(x_i, x_j) \in ML$, then $(x_j, x_i) \in ML$. There are no reflexive relation: $(x_i, x_i) \notin ML$ for $\forall x_i \in X$. We moreover assume that ML is *transitive*: if $(x_i, x_j) \in ML$ and $(x_j, x_l) \in ML$, then $(x_i, x_l) \in ML$. (Later we consider how to handle a non-transitive must-link.)

The symmetry also holds for CL: if $(x_k, x_l) \in CL$, then $(x_l, x_k) \in CL$, $(x_k, x_k) \notin CL$ for $\forall x_k \in X$. The meaning of them are clear from their names: $(x_i, x_j) \in ML$ means that x_i and x_j have to be linked in a cluster, while $(x_k, x_l) \in CL$ means that x_k and x_l have to be in different clusters. This concept represents semi-supervised learning in the sense that if $C = \{x, y, \ldots, z\}$ are in a class, it can be represented by a must-link of complete subgraph $ML = C \times C$. Hence the existence of CL is an essential generalization of predefined partial class labels.

We assume a dissimilarity measure which takes $0 \leq D(x, y) < +\infty$, while it is more difficult to use a similarity measure, since $S(x, y)$ is limited to the unit interval: $S(x, y) \in [0, 1]$. We moreover consider only three linkage methods of the single linkage, complete linkage, and average linkage. Generally to use the centroid method and Ward method is problematic, since the squared Euclidean distance cannot be distorted using the above constraints.

The main result in this section is as follows: *to use the complete linkage and average linkage has no problem with these constraints, while to use the single linkage needs careful consideration as shown below.*

We moreover note that we have two ways to handle the constraints: *hard constraints* and *soft constraints*: hard constraints mean that a pair of objects in ML are never permitted to be in different clusters and a pair in CL are never permitted to be in the same cluster. Soft constraints mean that those constraints are soft and some *penalties* are put if a pair breaks a constraint but the penalty is not very hard as those in hard constraints. We consider soft constraints as penalties in this section and note hard constraints can be handled as a special case of soft constraints. We use a positive penalty term $P(x_k, x_l)$ when (x_k, x_l) is in CL, while we do not use a penalty term for ML: instead we simply put

$$D(x_i, x_j) = 0, \quad for\ (x_i, x_j) \in ML. \tag{5.20}$$

We assume that $P(x, y)$ is defined for all $x, y \in X$ by putting

$$P(x, y) = \begin{cases} P(x, y) = Const. > 0, & (x, y) \in CL, \\ P(x, y) = 0 & (x, y) \notin CL. \end{cases} \tag{5.21}$$

If we set the constant to be sufficiently large, it means the hard constraint, e.g.,

$$Const. > \max_{x,y \in X} D(x, y). \tag{5.22}$$

The algorithm of agglomerative hierarchical clustering with the constraint uses a modified dissimilarity $D'(x, y)$:

$$D'(x, y) = \max\{D(x, y), P(x, y)\}. \tag{5.23}$$

Another choice for $D'(x, y)$ is

$$D'(x, y) = D(x, y) + P(x, y). \tag{5.24}$$

Accordingly, the modified algorithm is in Fig. 5.2, where updating $D(G_r, G_j)$ uses the rule of the single linkage, complete linkage, or average linkage, while updating $P(G_r, G_j)$ is

$$P(G_r, G_j) = \max\{P(G_p, G_j), P(G_q, G_j)\}, \tag{5.25}$$

i.e., it uses the rule of the complete linkage.

Let us moreover consider another method: define $D'(x, y)$ by (5.23) or (5.24) and use the updating formulas of the three linkage methods using $D'(G, G')$ instead of $D(G, G')$. This method also works for the complete linkage and the average linkage except that we should be careful for the choice of the constant of the penalty when we use the average linkage. On the other hand, this method does not work for the single linkage, as the single linkage may ignore the penalty, since it will use

$$D'(G, G') = \min_{x \in G, y \in G'} D'(x, y) = \min_{x \in G, y \in G'} \max\{D(x, y), P(x, y)\}. \tag{5.26}$$

To summarize, when we use the single linkage, we should use **CAHC**, while we can use both **CAHC** and **AHC** in Chap. 2 with $D'(x, y)$ instead of $D(x, y)$ when we apply the complete linkage or the average linkage.

Note 13 Constrained clustering for the centroid method and Ward method are omitted but the method of **CAHC** can also be used for these two. Updating formulas of the centroid method and Ward method should be used for $D(G, G')$ while $P(G, G')$ should be updated using the maximum (5.25).

Note 14 When we handle a nontransitive must-link, to put $D(x_i, x_j) = 0$ by (5.20) is useless, since there may be triplet (x_i, x_j, x_l) such that $D(x_i, x_j) = 0$, $D(x_j, x_l) = 0$

Input: initial clusters $\mathcal{G}(N_0) = \{G_1, \ldots, G_{N_0}\}$
Output: Dendrogram produced in **FORM_DENDROGRAM**
begin CAHC
$K = N_0$
Find a pair (G_p, G_q) of minimum dissimilarity (or maximum similarity):
$D'(G_p, G_q) = \min_{i,j(i \neq j)} D'(G_i, G_j)$
% If there are more than one pairs than attain the minimum/maximum,
% we need a *tie breaking rule.*
Put $L_K = D(G_p, G_q)$
Merge clusters:
$G_r = G_p \cup G_q$
Update $\mathcal{G}(K)$:
$\mathcal{G}(K-1) = \mathcal{G}(K) \cup \{G_r\} - \{G_p, G_q\}$
$K = K - 1$
If $K = 1$ call **FORM_DENDROGRAM** and stop
call **KEEP_DATA** of $(L_K, \mathcal{G}(K))$
Update dissimilarity $D(G_r, G_j)$ and $P(G_r, G_j)$ for all other $G_j \in \mathcal{G}(K)$
and let $D'(G_r, G_j) = \max\{D(G_r, G_j), P(G_r, G_j)\}$
(or let $D'(G_r, G_j) = D(G_r, G_j) + P(G_r, G_j)$)
go to **Find a pair**
end CAHC

Fig. 5.2 **CAHC**: Modified procedure of constrained agglomerative hierarchical clustering

and $D(x_i, x_l) > 0$, which may lead to a case when $\{x_i, x_j, x_l\}$ are not in a cluster if the complete linkage or average linkage is used. (Suppose $D(x_i, x_l) >> 0$.) In this case of nontransitivity, must-link should be handled as initial clusters $\mathcal{G}(N_0) = \{G_1, \ldots, G_{N_0}\}$ instead of $D(x_i, x_j) = 0$ by (5.20). To generate $\mathcal{G}(N_0)$, we should use **MST**(d) with $d = 0$; in other words, the single linkage should be first used for all pairs in ML and the resulting clusters should be used as initial $\mathcal{G}(N_0)$.

Note 15 Another interpretation of nontransitive must-link is to handle it as a soft constraint, since nontransitive must-link has ambiguity. In this interpretation the penalty of must-link P_{ML} can be minus: $P_{ML}(x_i, x_j) < 0$ for $(x_i, x_j) \in ML$.

5.3.1 An Illustrative Example

Let $X = \{a, b, c, d, e, f\}$, $D(a, b) = 1$, $D(b, c) = 3$, $D(a, c) = 2$, $D(e, f) = 3$, $D(d, f) = 2$, $D(c, f) = 4$, and $ML = \{(d, e)\}$, $CL = \{(a, d)\}$. Dissimilarities for all other pairs $x, y \in X$ are assumed to be $D(x, y) = 10$. The points and dissimilarity values with the constraints are shown in Fig. 5.3.

The dissimilarity in ML is interpreted as $D(d, e) = 0$, while $D(a, d)$ is set to be a large value: we set $D(a, d) = 20$ ($D(a, d) > \max_{x,y \in X} D(x, y)$ except $(x, y) \in CL$).

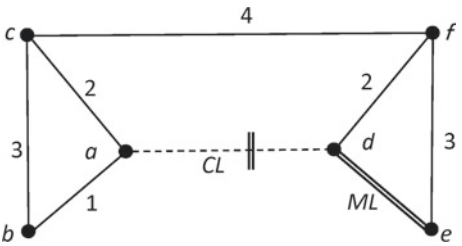

Fig. 5.3 Points $X = \{a, b, c, d, e, f\}$ with dissimilarity values $D(a, b) = 1$, $D(b, c) = 3$, $D(a, c) = 2$, $D(e, f) = 3$, $D(d, f) = 2$, $D(c, f) = 4$ and constraints $ML = \{(d, e)\}$, $CL = \{(a, d)\}$ shown by different types of edges. Dissimilarities for all other pairs $x, y \in X$ are assumed to be $D(x, y) = 10$ but not shown by an edge

Table 5.2 Result by the single linkage using **CAHC**

Level	Clusters
0	$\{d, e\}, \{a\}, \{b\}, \{c\}, \{f\}$
1	$\{d, e\}, \{a, b\}, \{c\}, \{f\}$
2	$\{d, e, f\}, \{a, b, c\}$
10	$\{d, e, f\}, \{a, b, c\}$

Table 5.3 Result by the complete linkage using **CAHC**

Level	Clusters
0	$\{d, e\}, \{a\}, \{b\}, \{c\}, \{f\}$
1	$\{d, e\}, \{a, b\}, \{c\}, \{f\}$
3	$\{d, e, f\}, \{a, b, c\}$
10	$\{d, e, f\}, \{a, b, c\}$

Apparently we will have two clusters $\{a, b, c\}, \{d, e, f\}$ and these two will not be connected because of $CL = \{(a, d)\}$. Indeed, **CAHC** produce hierarchical clusters in Tables 5.2, 5.3, and 5.4.

On the other hand, if we use **AHC** (not **CAHC**) with the modified dissimilarity $D'(x, y)$, the single linkage produces the result in Table 5.5, while the complete linkage and the average linkage result in the same clusters respectively in Tables 5.3 and 5.4. Thus the single linkage ignores the cannot-link, while the results by the complete linkage and the average linkage are unchanged.

Table 5.4 Result by the average linkage using **CAHC**

Level	Clusters
0	$\{d, e\}, \{a\}, \{b\}, \{c\}, \{f\}$
1	$\{d, e\}, \{a, b\}, \{c\}, \{f\}$
2.5	$\{d, e, f\}, \{a, b, c\}$
10	$\{d, e, f\}, \{a, b, c\}$

Table 5.5 Result by the single linkage using **AHC** with dissimilarity $D'(x, y)$

Level	Clusters
0	$\{d, e\}, \{a\}, \{b\}, \{c\}, \{f\}$
1	$\{d, e\}, \{a, b\}, \{c\}, \{f\}$
2	$\{d, e, f\}, \{a, b, c\}$
4	$\{a, b, c, d, e, f\}$

5.4 Model-Based Clustering and Agglomerative Hierarchical Algorithms

Model-based clustering generally refers to the mixture of distributions [14], typically *Gaussian mixture models* [15]. Using *EM algorithm* (see, e.g., [16]), parameters of mixtures are estimated and clusters are generated. The word 'model' above implies a statistical model.

However, this word in a more general usage means a mathematical model that describes objects and their relations in terms of a mathematical framework. In the latter sense, all clustering techniques are model-based. The single linkage, complete linkage, and average linkage are based on network (weighted graph) models, while the centroid and Ward methods are based on the Euclidean space model.

We note, at the same time, that some models are stronger, while others are weaker, especially in the sense of theoretical background and 'prediction or estimation capability'. Let us take a typical example of a statistical model. Given a set of data points, a statistical distribution model is assumed whereby the probability of occurrence of an event is estimated: in terms of a classification problem, a classification probability is estimated by the model of the distribution.

Such a property holds also for some methods of clustering. Unlike the probabilistic classification rule of a statistical model, the method of K-means has the deterministic rule of the nearest prototype classification as stated above. The single linkage has the nearest neighbor classification rule. Thus these two methods have the respective estimation rules for the classification problem, which means that the respective mathematical models have estimation capability.

Such consideration does not apply to the complete linkage and the average linkage, since the classification rules, if any, are not strong enough when compared with the

former two linkage methods, and geometrical properties like the Voronoi regions are not found in the latter two methods.

Another feature in agglomerative hierarchical clustering in relation to mathematical models is that some method uses a *single mathematical model*, while others use more than one mathematical models. Let us consider the latter case. Let us take the single linkage that is based on the weighted graph (alias network) model, while the definition of a similarity/dissimilarity measure is based on another mathematical model. Note that weights on the graph themselves should be calculated beforehand and cannot be defined in the network model. The example concerning Voronoi regions assumes the Euclidean space model to define the dissimilarity, and the minimum spanning tree (i.e., the single linkage) itself is not related to the Euclidean space. In this way, the single linkage, complete linkage, and average linkage have two types of mathematical models, one for defining a similarity or dissimilarity measure, and another for the linkage that is based on networks. (Note that the complete linkage is related to cliques, while how the average linkage is related to a network is ambiguous.)

In contrast, the centroid method and Ward method are based on the single model of the squared Euclidean distance throughout the whole procedures. Even when a positive definite kernel is used, the model is a high-dimensional Euclidean space.

We now proceed to consider a question which is not interested in by most users of agglomerative hierarchical clustering. 'Is it possible to develop a new and meaningful linkage method?' The answer is of course affirmative. Indeed, a number of other linkage methods than the standard ones described in Chap. 2 have been proposed. For example, 'the method of average linkage within the merged group' has been proposed [17]. This method is different from the average linkage method described in Chap. 2 and rarely used now, because a problem occurs that a pair of objects having no relation at all may be merged into a cluster (see, e.g., [18]).

ℓ_1-space based linkage method

Another possibility to develop a new linkage algorithm is to use the idea of the single model which is not based on the squared Euclidean distance. Let us consider, for example, a set of points $x_1 \ldots, x_N$ in $\boldsymbol{R}^M$: $x_i = (x_i^1, \ldots, x_i^M)$ $(i = 1, \ldots, N)$. However, $\boldsymbol{R}^M$ is not the Euclidean space but the ℓ_1-*space*. The dissimilarity is the norm of ℓ_1-space given by the following:

$$D(x, y) = \|x - y\|_1 = \sum_{j=1}^{M} |x^j - y^j|, \tag{5.27}$$

where $x = (x^1, \ldots, x^M)$ and $y = (y^1, \ldots, y^M)$. The ℓ_1-norm is also called the *Manhattan distance* or the *city block distance*.

We consider the ℓ_1-space version of the centroid method and Ward method. The *median* plays the role of the centroid in this space. Given real-valued data $z_1, \ldots, z_L$ such that $z_1 \leq z_2 \leq \cdots \leq z_L$, the median is defined by

$$\text{Median}((z_1, \ldots, z_L)) = \begin{cases} z_{\frac{L}{2}} & (L \text{ is even}) \\ \frac{1}{2}(z_{\frac{L-1}{2}} + z_{\frac{L+1}{2}}) & (L \text{ is odd}) \end{cases} \tag{5.28}$$

For a set of real numbers $\{w_1, \ldots, w_L\}$ that is not ordered, we arrange them into the increasing order $w_{p_1} \leq w_{p_2} \leq \cdots \leq w_{p_L}$, and define

$$\text{Median}(\{w_1, \ldots, w_L\}) = \text{Median}((w_{p_1}, \ldots, w_{p_L})). \tag{5.29}$$

Note that $\{w_1, \ldots, w_L\}$ is a set and unordered, while $(w_{p_1}, \ldots, w_{p_L})$ in the right hand side is a sequence and ordered.

For a set $X = \{x_1, \ldots, x_N\}$ of real vectors, we define the median of each component $\{x_1^j, \ldots, x_N^j\}$ and its median as above. Let the median vector for X be

$$\begin{aligned} \text{Median}(X) &= \text{Median}(\{x_1, \ldots, x_N\}) \\ &= (\text{Median}(\{x_1^1, \ldots, x_N^1\}), \ldots, \text{Median}(\{x_1^M, \ldots, x_N^M\})); \end{aligned} \tag{5.30}$$

in other words, the jth component of $\text{Median}(X)$ is the median of $\{x_1^j, \ldots, x_N^j\}$. In the same way, we define $\text{Median}(G)$ for a subset G of X.

Let us define

$$L_1(G, y) = \sum_{x_k \in G} \|x_k - y\|_1. \tag{5.31}$$

It is well known that the next equation holds.

$$L_1(G, \text{Median}(G)) = \min_{y \in \boldsymbol{R}^M} L_1(G, y). \tag{5.32}$$

This equation implies that $\text{Median}(G)$ is a good prototype of G in ℓ_1-space. We moreover define

$$E_1(G) = L_1(G, \text{Median}(G)) = \sum_{x_k \in G} \|x_k - \text{Median}(G)\|_1. \tag{5.33}$$

$E_1(G)$ can be compared with $E(G)$ for Ward method. Note that when G and G' that are disjoint ($G \cap G' = \emptyset$) are merged, we have

$$E_1(G \cup G') \geq E_1(G) + E_1(G'). \tag{5.34}$$

This inequality is clear from Eq. (5.32).

We now define, in parallel with the centroid and Ward methods, two methods for ℓ_1-space. A method[1] uses

[1] We do *not* call this method 'a median method', since it is different from 'the median method' proposed by Gower [19]. See Chap. 6.

$$D_{1M}(G, G') = \|\text{Median}(G) - \text{Median}(G')\|_1, \tag{5.35}$$

which is compared with the centroid method. The initial value is given by

$$D_{1M}(x, y) = D_{1M}(\{x\}, \{y\}) = \|x - y\|_1. \tag{5.36}$$

Another uses

$$D_{2M}(G, G') = E_1(G \cup G') - E_1(G) - E_1(G'), \tag{5.37}$$

which is compared with Ward method. The initial value is given by

$$\begin{aligned} D_{2M}(x, y) &= D_{2M}(\{x\}, \{y\}) = E(\{x, y\})) - E(\{x\}) - E(\{y\}) \\ &= \|x - \frac{x+y}{2}\|_1 + \|y - \frac{x+y}{2}\|_1 = \|x - y\|_1. \end{aligned} \tag{5.38}$$

Thus $D_{1M}(G, G')$ and $D_{2M}(G, G')$ are used as dissimilarity measures in **AHC** algorithm. We should note the following:

(i) Updating formulas like the ones in Chap. 2 is unavailable. In other words, we must calculate medians anew when two clusters are merged.
(ii) The method using $D_{1M}(G, G')$ does not satisfy the monotone property. The dendrogram may have reversals.
(iii) The method using $D_{2M}(G, G')$ does not satisfy the monotone property either. When the vertical level of the dendrogram is changed to $E_1(G \cup G')$ from $D_{2M}(G, G')$, then the dendrogram does not have a reversal. Note the inequality (5.34).

Example 12 Let us consider again the five points of the first example in Fig. 1.1, where the five points are $v = (0, 0), w = (1.5, 0), x = (4, 0), y = (2, 0), z = (4, 2)$. The plane is now assumed to be ℓ_1-space. Hence $D(v, z) = |4 - 0| + |2 - 0| = 6$, and so on.

Suppose the first method with D_{1M} is applied. First, the minimum ℓ_1-distance is between v and w: $D_{1M}(v, w) = 1.5$. Second merge occurs between x and z with $D_{1M}(x, z) = 2$. Note the medians are given by $\text{Median}(\{v, w\}) = (0, 0.75)$ and $\text{Median}(\{x, z\}) = (4, 1)$. Then $\{v, w\}$ and $\{y\}$ are merged with $D(\{v, w\}, \{y\}) = 2.75$. Finally $\{v, w, y\}$ and $\{x, z\}$ are merged:

$$\begin{aligned} D_{1M}(\{v, w, y\}, \{x, z\}) &= \|\text{Median}(\{v, w, y\}) - \text{Median}(\{x, z\})\|_1 \\ &= \|(0, 0) - (4, 1)\|_1 \\ &= 4 + 1 = 5. \end{aligned} \tag{5.39}$$

For the second method, first merge is between v and w:

$$D_{2M}(v, w) = E_1(\{v, w\}) = |(0, 0) - (0.75, 0)| + |(1.5, 0) - (0.75, 0)| = 1.5,$$

and the second merge is also between x and z with

$$D_{2M}(x, z) = E_1(\{x, z\}) = |(4, 0) - (4, 1)| + |(4, 2) - (4, 1)| = 2$$

Alternatively, we can merge $\{v, w\}$ and $\{y\}$ at the same level, since

$$\text{Median}(\{v, w, y\}) = v$$

and

$$\begin{aligned} D_{2M}(\{v, w\}, \{y\}) &= E_1(\{v, w, y\}) - E_1(\{v, w\}) \\ &= \|v - v\|_1 + \|w - v\|_1 + \|y - v\|_1 - E_1(\{v, w\}) \\ &= 3.5 - 1.5 = 2. \end{aligned} \tag{5.40}$$

Finally, $\{v, w, y\}$ and $\{x, z\}$ are merged:

$$\begin{aligned} D_{2M}(\{v, w, y\}, \{x, z\}) &= E_1(\{v, w, y, x, z\}) - E_1(\{v, w, y\}) - E_1(\{x, z\}) \\ &= \|v - w\|_1 + \|w - w\|_1 + \|x - w\|_1 + \|y - w\|_1 + \|z - w\|_1 \\ &\quad - E_1(\{v, w, y\}) - E_1(\{x, z\}) \\ &= 12 - 3.5 - 2 = 6.5. \end{aligned} \tag{5.41}$$

where $\text{Median}(\{v, w, y\}) = w$.

The dendrograms are omitted but the obtained clusters are listed in Table 5.6.

Table 5.6 List of clusters for the first example in Fig. 1.1, when ℓ_1-space-based methods using medians are applied

First method		
Clusters	Level of merging L_K	Median
$\{v\}, \{w\}, \{x\}, \{y\}, \{z\}$	–	–
$\{v, w\}, \{x\}, \{y\}, \{z\}$	1.5	$\text{Median}(\{v, w\}) = (0, 0.75)$
$\{v, w\}, \{y\}, \{x, z\}$	2	$\text{Median}(\{x, z\}) = (4, 1)$
$\{v, w, y\}, \{x, z\}$	2.75	$\text{Median}(\{v, w, y\}) = (0, 0)$
$\{v, w, x, y, z\}$	5	$\text{Median}(\{v, w, x, y, z\}) = (1.5, 0)$
Second method		
Clusters	Level of merging L_K	$E_1(G) = E_1(G_p \cup G_q)$
$\{v\}, \{w\}, \{x\}, \{y\}, \{z\}$	–	–
$\{v, w\}, \{x\}, \{y\}, \{z\}$	1.5	$E_1(\{v, w\}) = 1.5$
$\{v, w\}, \{y\}, \{x, z\}$	2	$E_1(\{x, z\}) = 2$
$\{v, w, y\}, \{x, z\}$	2	$E_1(\{v, w, y\}) = 3.5$
$\{v, w, x, y, z\}$	6.5	$E_1(\{v, w, x, y, z\}) = 12$

Note 16 We note, at the end of this section, a general idea of the method of a single mathematical model. The single model requires the definition of an inter-cluster similarity or dissimilarity measure. Let $D(G, G')$ is defined in some way for two clusters. Then, the dissimilarity $D(x, y)$ between two objects x, y is defined from $D(G, G')$:

$$D(x, y) = D(\{x\}, \{y\}). \tag{5.42}$$

In short, an inter-cluster similarity or dissimilarity is first defined and the corresponding inter-object similarity or dissimilarity is defined from the former. Note that the centroid and Ward methods are obeying this rule. On the other hand, two model methods first define $S(x, y)$ or $D(x, y)$ in some way and $S(G, G')$ or $D(G, G')$ is defined without reference to the model of the definition of $S(x, y)$ or $D(x, y)$.

5.5 Agglomerative Clustering Using Asymmetric Measures

There have been a number of studies in agglomerative hierarchical clustering using *asymmetric measures* (see, e.g., [20, 21]). A motivation behind these studies is that natural asymmetric measures are found in some applications of which we observe an example below. This motivation is related to a mathematical model of an asymmetric similarity/dissimilarity measure. Another motivation, somewhat technical one, is to obtain an *asymmetric dendrogram* (e.g., [20]).

Thus these studies consider the two problems of

1. a mathematical model of an asymmetric similarity/dissimilarity measure and
2. an asymmetric dendrogram.

There is still another problem of the *symmetrization* of a measure in an agglomerative hierarchical algorithm, which will be discussed later.

We begin with the definition of an asymmetric similarity/dissimilarity measure. Given a set of objects $X = \{x_1, \ldots, x_N\}$, an asymmetric similarity (or dissimilarity) measure is denoted by $S(x, y)$ (or $D(x, y)$) for all $x, y \in X$, $x \neq y$. $S(x, x)$ (or $D(x, x)$) may or may not be defined. Unlike symmetric cases, $S(x, y) = S(y, x)$ (or $D(x, y) = D(y, x)$) does not hold in general. Hence no general axioms like metrics or ordinary similarities/dissimilarities are assumed, but these asymmetric similarities/dissimilarities have implications of relatedness in specific applications.

A similarity/dissimilarity between two clusters $S(G, G')$ (or $D(G, G')$) for $G, G' \subset X$, $G \cap G' = \emptyset$ is then defined. There are many possible definitions for them, some of which are given in Saito and Yadohisa [21], but we omit the details, since we show a different way to define an asymmetric measure. Before stating a specific measure herein, we give an algorithm of asymmetric agglomerative clustering as **AHC-AS** in Fig. 5.4.

Readers will note that **AHC-AS** algorithm is essentially the same as **AHC** in Chap. 2. Indeed, nothing has been changed except the underlined part where

Input: initial clusters $\mathcal{G}(N_0) = \{G_1, \ldots, G_{N_0}\}$
Output: Dendrogram produced in **FORM_DENDROGRAM**
begin AHC-AS
  $K = N_0$
  Find a pair (G_p, G_q) of minimum dissimilarity (or maximum similarity):
    $S(G_p, G_q) = \max_{i,j(i \neq j)} S(G_i, G_j)$
    (or $D(G_p, G_q) = \min_{i,j(i \neq j)} D(G_i, G_j)$)
  % If there are more than one pairs than attain the minimum/maximum,
  % we need a *tie breaking rule*. See below.
    Put $L_K = S(G_p, G_q)$ (or $L_K = D(G_p, G_q)$)
  Merge clusters:
    $G_r = G_p \cup G_q$
  Update $\mathcal{G}(K)$:
    $\mathcal{G}(K-1) = \mathcal{G}(K) \cup \{G_r\} - \{G_p, G_q\}$
    $K = K - 1$
  If $K = 1$ call **FORM_DENDROGRAM** and stop
  call **KEEP_DATA** of $(L_K, \mathcal{G}(K))$
  Update $S(G_r, G_j)$ and $S(G_j, G_r)$ (or dissimilarity $D(G_r, G_j)$ and $D(G_j, G_r)$)
    for all other $G_j \in \mathcal{G}(K)$
  go to **Find a pair**
end AHC-AS

Fig. 5.4 **AHC-AS**: Procedure of agglomerative hierarchical clustering for an asymmetric measure

$S(G_r, G_j)$ and $S(G_j, G_r)$ are distinguished. If the measure is symmetric, **AHC-AS** is reduced to **AHC**. Hence the problem is focused upon the way to define a similarity/dissimilarity between clusters.

Note 17 **AHC-AS** algorithm is different from the algorithm given in Saito and Yadohisa [21], where many options are used instead of

$$S(G_p, G_q) = \max_{i,j(i \neq j)} S(G_i, G_j). \tag{5.43}$$

For example, the followings are proposed:

$$S(G_p, G_q) = \max_{i,j(i \neq j)} \frac{1}{2}\{S(G_i, G_j) + S(G_j, G_i)\}, \tag{5.44}$$

$$S(G_p, G_q) = \max_{i,j(i \neq j)} \min\{S(G_i, G_j), S(G_j, G_i)\}. \tag{5.45}$$

A Reference Probability Model

Unlike other studies in asymmetric measures, we consider a method of a single model, i.e., we define $S(G, G')$ based on a specific model and put $S(x, y) = S(\{x\}, \{y\})$ as stated above.

A specific mathematical model here is related to 'reference or citation studies' [18]. As noted above, the motivation to use a particular model for an asymmetric measure is essential and hence we consider a specific example instead of a general description.

Assume that X is the set of articles. An article x in X refers to other articles: some references are y in X, and others are not in X. In other words, X is not closed with respect to reference relations. For $x, y \in X$, the number of references from x to y is denoted by $n(x, y)$. Total number of references in x is denoted by $\bar{n}(x)$. Since x refers to articles that may or may not be in X in general, we assume

$$\bar{n}(x) \geq \sum_{y \in X} n(x, y), \tag{5.46}$$

and the equality does not hold in general.

We can define a *reference probability* (or more precisely an estimate of the reference probability) from x to y denoted by $P(x, y)$:

$$P(x, y) = \frac{n(x, y)}{\bar{n}(x)}. \tag{5.47}$$

For two disjoint subsets G, G' ($G \cap G' = \emptyset$) of X, the above measures are immediately extended to $n(G, G')$, $\bar{n}(G)$, and $P(G, G')$:

$$n(G, G') = \sum_{x \in G, y \in G'} n(x, y), \tag{5.48}$$

$$\bar{n}(G) = \sum_{x \in G} \bar{n}(x), \tag{5.49}$$

$$P(G, G') = \frac{n(G, G')}{\bar{n}(G)}. \tag{5.50}$$

We then define the averaged probability

$$S(G, G') = \frac{P(G, G')}{|G'|}, \tag{5.51}$$

where $|G'|$ is the number of objects in $|G'|$. Note that $P(x, y) = S(\{x\}, \{y\})$.

We are now ready to use $S(G, G')$ as asymmetric similarity in **AHC-AS**. Moreover updating formulas are developed. When $G_r = G_p \cup G_q$, we have

$$\bar{n}(G_r) = \bar{n}(G_p) + \bar{n}(G_q), \tag{5.52}$$

$$n(G_r, G_j) = n(G_p, G_j) + n(G_q, G_j), \tag{5.53}$$

$$n(G_j, G_r) = n(G_j, G_p) + n(G_j, G_q), \tag{5.54}$$

$$P(G_r, G_j) = \frac{n(G_r, G_j)}{\bar{n}(G_r)}, \tag{5.55}$$

$$P(G_j, G_r) = \frac{n(G_j, G_r)}{\bar{n}(G_j)}, \tag{5.56}$$

$$S(G_r, G_j) = \frac{P(G_r, G_j)}{|G_j|}, \tag{5.57}$$

$$S(G_j, G_r) = \frac{P(G_j, G_r)}{|G_r|}, \tag{5.58}$$

or more simply,

$$S(G_p \cup G_q, G_j) = \frac{n(G_p, G_j) + n(G_q, G_j)}{(\bar{n}(G_p) + \bar{n}(G_q))|G_j|}, \tag{5.59}$$

$$S(G_j, G_p \cup G_q) = \frac{n(G_j, G_p) + n(G_j, G_q)}{\bar{n}(G_j)(|G_p| + |G_q|)}. \tag{5.60}$$

We proceed to show the *monotone property* of the merging levels of this method. We have the next proposition.

Proposition 15 *For an arbitrarily given X with $n(x, y)$ and $\bar{n}(x)$ satisfying (5.46), assume that* **AHC-AS** *algorithm is applied to $S(x, y) = P(x, y)$. Assume also that the updating formulas (5.52), (5.53), (5.54), (5.59), and (5.60) are used. Then the merging levels are monotonic:*

$$L_N \geq L_{N-1} \geq \cdots \geq L_2. \tag{5.61}$$

Hence the dendrogram has no reversal.

Proof We first note a simple formula: given $p_1, p_2 \geq 0$ and $q_1, q_2 > 0$ such that $\frac{p_1}{q_1} \leq \frac{p_2}{q_2}$, we have

$$\frac{p_1}{q_1} \leq \frac{p_1 + p_2}{q_1 + q_2} \leq \frac{p_2}{q_2}. \tag{5.62}$$

Applying this inequality to (5.59) and (5.60), we have

$$S(G_r, G_j) \leq \max\{S(G_p, G_j), S(G_q, G_j)\}, \tag{5.63}$$

$$S(G_j, G_r) \leq \max\{S(G_j, G_p), S(G_j, G_q)\}. \tag{5.64}$$

Since

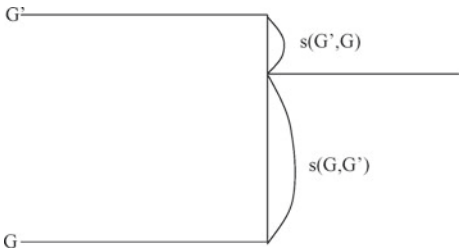

Fig. 5.5 A branch of an asymmetric dendrogram using the ratio of $S(G', G)$ and $S(G, G')$ where $S(G', G) \leq S(G, G')$

$$\max\{S(G_p, G_j), S(G_q, G_j), S(G_j, G_p), S(G_j, G_q)\} \leq \max_{i,k} S(G_i, G_k), \quad (5.65)$$

we have $L_K \geq L_{K-1}$ $(K = N, N-1, \ldots, 3)$. The proposition is thus proved. □

We moreover have the following. The proof is easy after seeing the above property and hence omitted.

Proposition 16 *For an arbitrarily given X with $n(x, y)$ and $\bar{n}(x)$ satisfying (5.46), assume that a modified version of* **AHC-AS** *algorithm where (5.43) is replaced by (5.44) or (5.45) is applied to $S(x, y) = P(x, y)$. Assume also that the updating formulas (5.52), (5.53), (5.54), (5.59), and (5.60) are used. Then the merging levels are monotonic:*

$$L_N \geq L_{N-1} \geq \cdots \geq L_2. \quad (5.66)$$

and the dendrogram has no reversal.

Asymmetric Dendrograms

Dendrograms reflecting the asymmetry of similarity measures have also been studied. We consider here a simple example of an asymmetric dendrogram using the idea by Okada and Iwamoto [20] which is shown in Fig. 5.5. The branch in this figure is asymmetric using the ratio of $S(G', G)$ and $S(G, G')$.

Example 13 Small real example of references among eight scientific journals in statistics is shown. The journals are listed in Table 5.7. The reference frequencies are shown in Stigler [22]. On the basis of this reference relations, the derived asymmetric dendrogram using the model in this section is shown in Fig. 5.6.

Asymmetric Dendrogram Using Hypothesis Testing

Another method for an asymmetric dendrogram is proposed in Takumi et al. [23]. This method uses a statistical hypothesis testing based on the probabilistic nature of $S(G, G')$ of the above model (5.51). We hence test the hypothesis:

$$\mathrm{H}: \quad S(G, G') = S(G, G'), \quad (5.67)$$

and asymmetry in the dendrogram is classified into three classes:

Table 5.7 Eight journals in statistics with their abbreviations: the abbreviations are used in the dendrogram in Figs. 5.6 and 5.8

AnnSt	Annals of statistics
Biocs	Biometrics
Bioka	Biometrika
Comst	Communications in statistics
JASA	Journal of the American statistical association
JRSSB	Journal of the royal statistical society: series B
JRSSC	Journal of the royal statistical society: series C
Tech	Technometrics

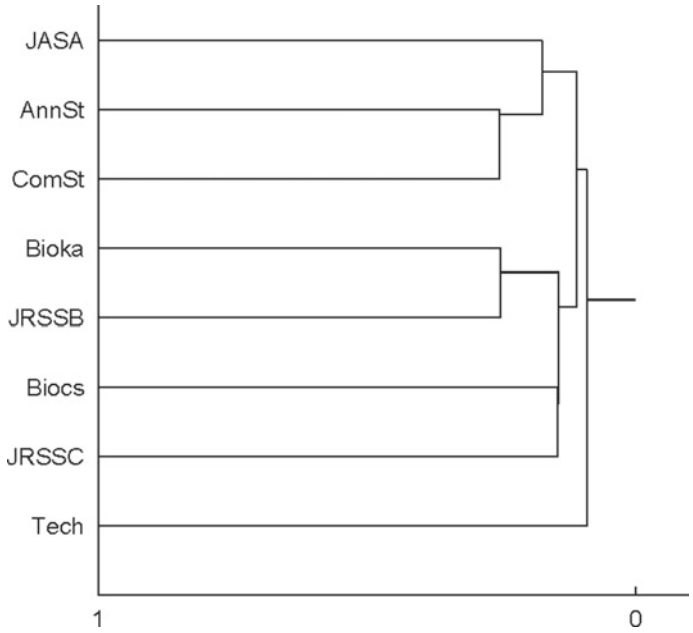

Fig. 5.6 The result of the asymmetric dendrogram of reference relations among eight journals in statistics. The data is from Stigler [22]

(i) Hypothesis H is rejected with the significance level $P = 1\%$,
(ii) Hypothesis H is rejected with the significance level $P = 5\%$ but not $P = 1\%$, or
(iii) Hypothesis H is not rejected.

Accordingly, three types of branches shown in Fig. 5.7 are defined.

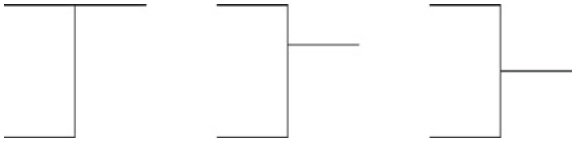

Fig. 5.7 Branches of asymmetric dendrogram using the hypothesis testing: the left figure means that the hypothesis H is rejected with 1% significance level, the center means that H is rejected with 5% significance level, and the right means that H is not rejected

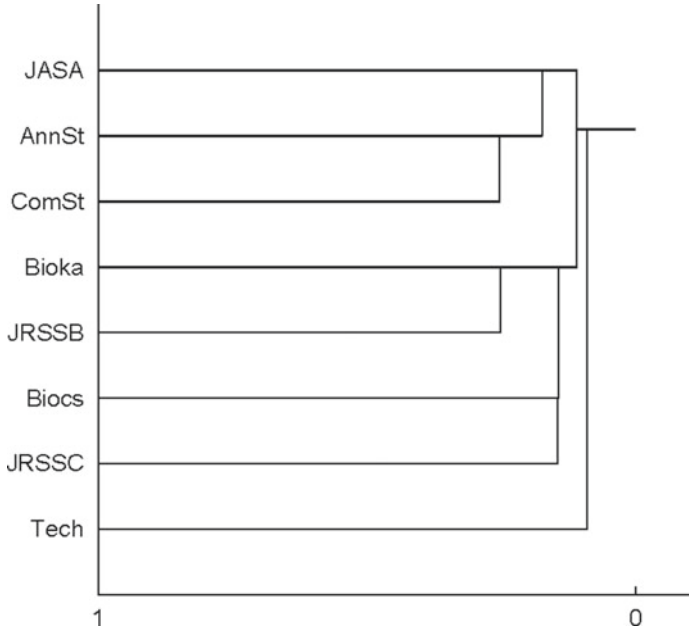

Fig. 5.8 The result of the asymmetric dendrogram using hypothesis testing of reference relations among eight journals in statistics. The data is from Stigler [22]

Example 14 Figure 5.8 shows the asymmetric dendrogram using the above method of hypothesis testing. The algorithm of merging and the obtained clusters is the same as the previous example but the dendrograms are different. Every hypothesis at merging has been rejected with $P = 1\%$ in this example.

References

1. M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in *KDD-96 Proceedings* (1996), pp. 226–231
2. D. Wishart, Mode analysis: a generalization of nearest neighbor which reduces chaining effects, in ed by A.J. Cole, *Numerical Taxonomy, Proceedings of the Colloquium, in Numerical Taxonomy* (University St Andrews, 1968), pp. 283–311
3. E. Schubert, J. Sander, M. Ester, H.-P. Kriegel, X. Xu, DBSCAN revisited: why and how you should (still) use DBSCAN. ACM Trans. Database Syst. **42**(3), 1–21 (2017)
4. S. Miyahara, S. Miyamoto, A family of algorithms using spectral clustering and DBSCAN, in *Proceedings of 2014 IEEE International Conference on Granular Computing (GrC2014), Noboribetsu, Hokkaido, Japan, 22–24 Oct 2014* (2014), pp. 196–200
5. V.N. Vapnik, *Statistical Learning Theory* (Wiley, Neww York, 1998)
6. O. Chapelle, A. Zien, B. Scholköpf (eds.), *Semi-Supervised Learning* (MIT Press, Cambridge, Massachusetts, USA, 2006)
7. X. Zhu, A.B. Goldberg, *Introduction to Semi-Supervised Learning* (Morgan and Claypool Publishers, San Rafael, CA, USA, 2009)
8. R.O. Duda, P.E. Hart, *Pattern Classification and Scene Analysis* (Wiley, New York, 1973)
9. S. Miyamoto, *Introduction to Cluster Analysis* (Morikita-Shuppan, Tokyo, 2000). ((in Japanese))
10. D. Arthur, S. Vassilvitskii, k-means++: the advantages of careful seeding, in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics Philadelphia, PA, USA* (2007), pp. 1027–1035
11. Y. Tamura, S. Miyamoto, A method of two stage clustering using agglomerative hierarchical algorithms with one-pass k-Means++ or k-Median++, in*Proceedings of 2014 IEEE International Conference on Granular Computing (GrC2014)* (Noboribetsu, Hokkaido, Japan, 22–24 Oct 2014), pp. 281–285
12. S. Basu, I. Davidson, K. Wagstaff (eds.), *Constrained Clustering: Advances in Algorithms, Theory, and Applications* (Chapman & Hall/CRC, Boca Raton, FL, USA, 2009)
13. I. Davidson, S.S. Ravi, Agglomerative hierarchical clustering with constraints: theoretical and empirical results. Knowl. Discov. Databases: PKDD **2005**(LNCS3721), 59–70 (2005)
14. P. Giordani, M.B. Ferraro, F. Martella, *An Introduction to Clustering with R* (Springer Nature, Singapore, 2020)
15. G.J. McLachlan, D. Peel, *Finite Mixture Models* (Wiley, New York, 2000)
16. G.J. McLachlan, T. Krishnan, *The EM Algorithm and Extensions*, 2nd edn. (Wiley, Hoboken, NJ, 2008)
17. M.R. Anderberg, *Cluster Analysis for Applications* (Academic Press, New York, 1973)
18. S. Miyamoto, *Fuzzy Sets in Information Retrieval and Cluster Analysis* (Springer, Heidelberg, 1990)
19. J.C. Gower, A comparison of some methods of cluster analysis. Biometrics **23**, 623–637 (1967)
20. A. Okada, T. Iwamoto, A comparison before and after the joint first stage achievement test by asymmetric cluster analysis. Behaviormetrika **23**(2), 169–185 (1996)
21. T. Saito, H. Yadohisa, *Data Analysis of Asymmetric Structures* (Marcel Dekker, New York, 2005)
22. S.M. Stigler, Citation patterns in the journals of statistics and probability. Stat. Sci. **9**, 94–108 (1994)
23. S. Takumi, S. Miyamoto, Top-down versus Bottom-up methods of linkage for asymmetric agglomerative hierarchical clustering, in *Proceedings of 2012 IEEE International Conference on Granular Computing* (11–12 Aug, Hangzhou, China, 2012), pp. 542–547

Chapter 6
Miscellanea

Some methods which have not discussed in the foregoing chapters are overviewed in this chapter. The main difference between this and the previous chapter is that the descriptions in the previous chapter is self-contained, while those in this chapter is generally not.

First, two more linkage methods are described, and monotonicity and refinement properties are considered. Second, the merging levels of Ward method are further studied. Weighted objects are then handled. Finally, discussions throughout this book including future study possibilities are given.

6.1 Two Other Linkage Methods

Two linkage methods that have not been described up to now are considered, which are *Lance–Williams formula* [1] and the *median method* [2].

6.1.1 Lance–Williams Formula

We first consider theoretical properties of the Lance–Williams formula [1] in this section.

Lance and Williams [1] proposed the following generalized form of the updating formula.

$$D(G_r, G_j) = \alpha_p D(G_p, G_j) + \alpha_q D(G_q, G_j) + \beta D(G_p, G_q) + \gamma |D(G_p, G_j) - D(G_q, G_j)|, \tag{6.1}$$

S. Miyamoto, *Theory of Agglomerative Hierarchical Clustering*,
Behaviormetrics: Quantitative Approaches to Human Behavior 15,
https://doi.org/10.1007/978-981-19-0420-2_6

where parameters α_p, α_q, β, and γ are either constants or functions of $|G_p|$, $|G_q|$, and $|G_j|$.

The five linkage methods in Chap. 5 and the median method are all represented as particular cases of (6.1) by adjusting parameters as follows. Note that a dissimilarity measure is used.

Single linkage:

$$\alpha_p = \alpha_q = \frac{1}{2}, \quad \beta = 0, \quad \gamma = -\frac{1}{2}. \tag{6.2}$$

Complete linkage:

$$\alpha_p = \alpha_q = \frac{1}{2}, \quad \beta = 0, \quad \gamma = \frac{1}{2}. \tag{6.3}$$

Average linkage:

$$\alpha_p = \frac{|G_p|}{|G_p| + |G_q|}, \quad \alpha_q = \frac{|G_q|}{|G_p| + |G_q|}, \quad \beta = 0, \quad \gamma = 0. \tag{6.4}$$

Centroid method:

$$\alpha_p = \frac{|G_p|}{|G_p| + |G_q|}, \quad \alpha_q = \frac{|G_q|}{|G_p| + |G_q|}, \quad \beta = -\frac{|G_p||G_q|}{(|G_p| + |G_q|)^2}, \quad \gamma = 0. \tag{6.5}$$

Ward method:

$$\alpha_p = \frac{|G_p| + |G_j|}{|G_p| + |G_q| + |G_j|}, \quad \alpha_q = \frac{|G_q| + |G_j|}{|G_p| + |G_q| + |G_j|},$$
$$\beta = -\frac{|G_j|}{|G_p| + |G_q| + |G_j|}, \quad \gamma = 0. \tag{6.6}$$

Median method:

$$\alpha_p = \alpha_q = \frac{1}{2}, \quad \beta = -\frac{1}{4}, \quad \gamma = 0. \tag{6.7}$$

Note 18 In case when a similarity measure is used, the single linkage uses $\alpha_p = \alpha_q = \frac{1}{2}, \beta = 0, \gamma = \frac{1}{2}$, and the complete linkage uses $\alpha_p = \alpha_q = \frac{1}{2}, \beta = 0, \gamma = -\frac{1}{2}$. The formula is unchanged if the average linkage is used.

Note 19 *The median method* [2] uses the following.

$$D(G_r, G_j) = \frac{1}{2}\{D(G_p, G_j) + D(G_q, G_j)\} - \frac{1}{4}D(G_p, G_q), \tag{6.8}$$

when G_p and G_q are merged: $G_r = G_p \cup G_q$.

Lance and Williams [1] proposed a family of linkage methods called *flexible strategy* by putting

$$\alpha_p + \alpha_q + \beta = 1, \quad \alpha_p = \alpha_q, \quad \beta < 1, \quad \gamma = 0 \tag{6.9}$$

and varying β.

We next consider theoretical properties of Lance–Williams formula. We assume a milder condition than (6.9):

$$\alpha_p + \alpha_q + \beta \geq 1, \quad \alpha_p \geq 0,\ \alpha_q \geq 0,\ \gamma \geq 0. \tag{6.10}$$

The next proposition is due to Lance and Williams [1].

Proposition 17 *By assuming (6.10), the family of linkage methods with the updating formula (6.1) has the monotone property of the merging levels, i.e., dendrograms from these methods have no reversals.*

Proof Let

$$\zeta = \min\{D(G_p, G_j), D(G_q, G_j)\}, \tag{6.11}$$
$$\eta = D(G_p, G_q), \tag{6.12}$$
$$\theta = |D(G_p, G_j) - D(G_q, G_j)| \tag{6.13}$$

for simplicity. Note $\zeta \geq \eta$. Then we have

$$\begin{aligned} D(G_r, G_j) &= \alpha_p D(G_p, G_j) + \alpha_q D(G_q, G_j) + \beta\eta + \gamma\theta \\ &\geq \alpha_p \zeta + \alpha_q \zeta + \beta\eta + \gamma\theta \\ &\geq \eta + \gamma\theta \geq \eta, \end{aligned}$$

which shows $D(G_r, G_j) \geq D(G_p, G_q)$ for all other clusters G_j. Thus the monotonicity of the merging levels is proved. □

Let $\mathcal{G}_{LW}(\alpha)$ ($\alpha \in [0, \infty)$) be the family of clusters generated by Lance–Williams formula. Moreover, we put $\mathcal{G}_{LW} = \{\mathcal{G}_{LW}(\alpha) : \alpha \in [0, \infty)\}$ as a simplified notation. We have the next property.

Proposition 18 *Assume (6.10) and in addition $\beta \leq 0$. Then the refinement property holds:*

$$\mathcal{G}_{LW}(\alpha) \prec \mathcal{G}_{SL}(\alpha), \quad \alpha \in [0, \infty). \tag{6.14}$$

Or in short,

$$\mathcal{G}_{LW} \prec \mathcal{G}_{SL}. \tag{6.15}$$

Proof It is sufficient to show that the method satisfies the condition (3.29) in Proposition 9.

We assume (6.11)–(6.13). Let $\alpha_p + \alpha_q + \beta = 1 + \epsilon$ with $\epsilon \geq 0$. Since $\beta \leq 0$, we have

$$\begin{aligned} D(G_r, G_j) &= \alpha_p D(G_p, G_j) + \alpha_q D(G_q, G_j) + \beta\eta + \gamma\theta \\ &\geq (1 + \epsilon - \beta)\zeta + \beta\eta + \gamma\theta \\ &\geq (1 + \epsilon)\zeta - \beta(\zeta - \eta) + \gamma\theta \geq \zeta. \end{aligned}$$

Thus the condition (3.29) is satisfied and hence $\mathcal{G}_{LW} \prec \mathcal{G}_{SL}$ from Proposition 9. □

Note 20 The median method with $\alpha_p = \alpha_q = \frac{1}{2}$, $\beta = -\frac{1}{4}$, $\gamma = 0$ does not satisfy the condition (6.10), and hence reversals may occur.

6.2 More on Ward Method

As noted before, the Ward method is the most popular in various applications. We discussed a number of methodological properties of this method: they are summarized as follows:

1. The Ward method is based on a stepwise optimization of the objective function $E(G)$ which is common with that of K-means.
2. The level of merging is the increment $\Delta E(G_p, G_q)$ of the objective function between clusters after merging and that before merging.
3. The Kernel method of clustering can be applied to the Ward method when the similarity matrix is positive-definite. Moreover, even when the matrix is not positive-definite, the result can be justified using the concept of regularization.

We consider more features of the Ward method in this section.

6.2.1 Merging Levels of Ward Method

Except the Ward method, other linkage methods have the common rule of merging levels, i.e., $L_K = D(G_p, G_q)$ (or $L_K = S(G_p, G_q)$) and $D(G_p, G_q)$ is a simple function of $D(x, y)$ in each method. In short, merging levels L_K is meaningful in each linkage method. In contrast, $L_K = D(G_p, G_q) = \Delta E(G_p, G_q)$ itself seems to be rather meaningless: a more meaningful quantity would be $E(G_r) = E(G_p \cup G_q)$ instead of $E(G_r) - E(G_p) - E(G_q)$ in Ward method. However, we have

$$E(G_r) = L_{N-1} + \cdots + L_K \tag{6.16}$$

and $E(G_r) \neq L_K$.

The exact reason that $\Delta E(G_p, G_q)$ is taken to be the level L_K is unknown and methodologically there is no good reason. The author imagines that dendrograms using $L_K = \Delta E(G_p, G_q)$ have nicer shapes than those with the levels of $E(G_r)$, and at the same time users in applications do not care about the meanings of the values of the merging levels. They often care 'shapes' but not the 'levels'. There is no fundamental problem for such attitude. There are still rooms of misunderstanding, however, since we are liable to consider a merging level to be a meaningful value of the error sum in this case. For example, it is meaningless to 'cut' a dendrogram between L_{K-1} and L_K, while it is still possible to cut it at L_K.

6.2.2 Divisive Clustering and X-means

Although *divisive clustering* [3] has not been studied up to now and is not a topic to be handled herein, let us consider a typical example of K-means. Suppose $K = 2$ and we divide X into two clusters of G_1 and G_2. Then G_1 and G_2 are further divided into two subclusters G_{11}, G_{12} and G_{21}, G_{22} using K-means with $K = 2$. This procedure continues until a sufficient number of clusters is obtained. Thus this procedure has the opposite direction of agglomerative clustering. Specifically, the above method of iterative division of clusters using K-means can be contrasted with Ward method, since the objective function is the same for the both.

A problem in divisive clustering is when to stop the iteration, since to determine an appropriate number of clusters is not an easy problem. The method of *X-means* [4] uses the above divisive procedure with a criterion of a statistical model selection technique to determine the number of clusters. Another problem thereby arises that such a method of determining the number of clusters in Ward method is possible or not. Abe et al. [5] proposed a method to determine the number of clusters in a modified Ward method which partly uses an idea in Newman's method [6] and X-means.

6.2.3 Some Similarity Measures and Ward Method

We have discussed several similarity measures in the first chapter which include those for binary data. There is a natural question whether or not they are applicable to Ward method (and the centroid method). Three measures of $S_{smc}(x, y)$, $S_{jc}(x, y)$, and $S_{cc}(x, y)$ have been introduced in Chap. 1. Although we can use all of them (and also other similarity measures in standard books) for *ad hoc* Ward method in Sect. 4.3, but not for the centroid method, we consider whether they can be used for the real Ward method and the centroid method. This also implies that the matrices of these measures are positive-definite or not.

As for the Jaccard coefficient $S_{jc}(x, y)$, we cannot prove the positive-definite property, and hence the real Ward method may not be used for this measure, while

the other two measures are applicable to the *real* Ward method and the centroid method.

Indeed, the dissimilarity measure $1 - S_{smc}(x, y)$ derived from the simple matching coefficient satisfies (1.12) which is equivalent to the squared Euclidean distance and hence Ward method is applicable. The measure $S_{cc}(x, y)$ is based on the cosine correlation (1.8). If we normalize data $x = (x^1, \ldots, x^N)^\top$ and $y = (y^1, \ldots, y^N)^\top$:

$$x \to x' = \frac{x}{\|x\|}, \quad y \to y' = \frac{y}{\|y\|}, \tag{6.17}$$

then

$$D_{cc}(x, y) = 1 - S_{cc}(x, y) = 1 - S_{cc}(x', y') = \frac{1}{2}\|x' - y'\|^2. \tag{6.18}$$

Hence the dissimilarity derived from $S_{cc}(x, y)$ is the squared Euclidean distance.

We thus observe that the simple matching coefficient and the cosine coefficient can be used for Ward method (and hence the centroid method also), but one point should be noted, i.e., Ward method and the centroid method use the centroid $m(G_i)$ for each cluster G_i. Apparently the components of $m(G_i)$ are not binary-valued, even when all data are binary. The simplest example is $G = \{(1, 0), (0, 1)\}$ and $m(G) = (\frac{1}{2}, \frac{1}{2})$. Thus the use of Ward method and the centroid method to binary data implies that the users are admitting those *non-binary* representative points for clusters, which may not be interpreted in the original framework of binary data.

6.3 Handling Weighted Objects

We sometimes need to handle weighted objects: x_j with weight $w_j > 0$ ($j = 1, 2, \ldots, N$) in clustering. A concrete example is a two-stage clustering in which first stage is K-means and the second stage is an agglomerative hierarchical clustering. The initial objects in the second stage are the centers of the K-means in the first stage. In this case a center is a representative of a cluster and different clusters have different numbers of objects. Thus the initial objects in the second stage represent different numbers of objects in the first stage. In such a case *weighted objects* should be handled.

We hence assume data are given with weights: $\{(x_j, w_j)\}_{1 \le j \le N}$ and apply **AHC** algorithm with the standard linkage methods. To handle the weights, the following definition of $|G_j|$ is useful:

$$|G_j| = \sum_{x_k \in G_j} w_k. \tag{6.19}$$

$|G_j|$ is the total weight in cluster G_j instead of the number of objects.

We describe below how each linkage method incorporates weight (w_j).

Single linkage and complete linkage:
For these two linkage methods, the weight have no influence, since the definitions

$$D_{SL}(G, G') = \min_{x \in G, y \in G'} D(x, y) \tag{6.20}$$

$$D_{CL}(G, G') = \max_{x \in G, y \in G'} D(x, y). \tag{6.21}$$

do not include (w_j). In other words, we cannot incorporate the weight for the single and complete linkages.

Average linkage:
The definition and the updating formula are the same except that $|G|$ is given by (6.19):

$$D_{AL}(G, G') = \frac{1}{|G||G'|} \sum_{x \in G, y \in G'} D(x, y) \tag{6.22}$$

$$D_{AL}(G_r, G_j) = \frac{1}{|G_r|}\{|G_p| D_{AL}(G_p, G_j) + |G_q| D_{AL}(G_q, G_j)\}. \tag{6.23}$$

Centroid method:
Note that the centroid is calculated by

$$m(G) = \frac{1}{|G|} \sum_{x_k \in G} w_k x_k, \tag{6.24}$$

where $|G|$ is calculated by (6.19). Then the definition and the updating formula are the same as before:

$$D_{CNT}(G, G') = \|m(G) - m(G')\|^2, \tag{6.25}$$

$$D_{CNT}(G_r, G_j) = \frac{|G_p|}{|G_r|} D_{CNT}(G_p, G_j) + \frac{|G_q|}{|G_r|} D_{CNT}(G_q, G_j) - \frac{|G_p||G_q|}{|G_r|^2} D_{CNT}(G_p, G_q). \tag{6.26}$$

Ward method:
Note that the definition of the error sum is now given by

$$E(G) = \sum_{x_k \in G} w_k \|x_k - m(G)\|^2. \tag{6.27}$$

The definition

$$D_{WRD}(G, G') = \Delta E(G, G') = E(G \cup G') - E(G) - E(G') \tag{6.28}$$

remains the same when $E(G)$ is calculated by (6.27). Updating formula is also the same:

$$D_{WRD}(G_r, G_j) = \frac{1}{|G_r| + |G_j|}\{(|G_p| + |G_j|)D_{WRD}(G_p, G_j) + (|G_q| + |G_j|)D_{WRD}(G_q, G_j) - |G_j|D_{WRD}(G_p, G_q)\}, \quad (6.29)$$

where $|G_r| = |G_p| + |G_q|$. The difference is in the initial dissimilarity value:

$$\begin{aligned} D_{WRD}(x_i, x_j) &= \Delta E(\{x_i\}, \{x_j\}) = E(\{x_i, x_j\}) - E(\{x_i\}) - E(\{x_j\}) \\ &= \|x_i - \frac{w_i x_i + w_j x_j}{w_i + w_j}\|^2 + \|x_j - \frac{w_i x_i + w_j x_j}{w_i + w_j}\|^2 \\ &= \frac{w_i^2 + w_j^2}{(w_i + w_j)^2}\|x_i - x_j\|^2. \end{aligned} \quad (6.30)$$

6.4 Discussion and Future Works

We have emphasized the importance of theoretical aspects throughout this book, although it is often overlooked by researchers and users of cluster analysis. Specifically, we have focused upon the following features.

- The whole technique of agglomerative hierarchical clustering is described by an algorithm called **AHC**, whereby an interested reader can develop a computer program by himself without difficulty.
- The theory of the single linkage is described and it is shown that the graph concept is essential.
- Some recent features such as the use of positive-definite kernels are described.
- It is shown that the concept of model-based clustering is applicable to agglomerative hierarchical clustering.

We comment these points below in more detail, and relations to some methods related to support vector machines.

Algorithm for Agglomerative Hierarchical Clustering

By the specific **AHC** algorithm, methodologically interested readers could develop his own computer program having possibly his own linkage method. For example, the author developed a multiset-based linkage method [7] with application to document index and clustering.

Another implication of this algorithm is that the tree data structure of the dendrogram can be regarded as a database, i.e., data related to objects are stored at nodes of a dendrogram, and statistics of clusters are accordingly calculated when necessary. For example, in addition to the record [id, label, level, left_child, right_child] in

subprocedure **KEEP_DATA** (see Chap. 3), we can add $m(G)$, the centroid of cluster G and $Cov(G)$, its covariance matrix.

Theory of Single Linkage

The single linkage method itself is relatively unpopular, but its theoretical properties are sound and worth to be noted. We hence showed that it is equivalent to weighted graph concepts, and there are variations of the single linkage such as the mode analysis.

It appears that the graph concept is not very useful in practice, but actually it is useful in developing an efficient algorithm and computer program. For example, the description of the mode analysis in Wishart's paper [8] seems a bit cumbersome. After we observe, however, that it uses nodes with weights, it is straightforward to develop an algorithm using a variation of a minimum spanning tree. A variation of DBSCAN algorithm [9] can be developed likewise. Such a variation of the single linkage may be a subject for future research, and there is possibility of developing a new algorithm.

Positive-Definite Kernel

The use of positive-definite kernel in clustering should not be regarded as an exceptional case, even for agglomerative hierarchical clustering. We showed if similarity matrix S is positive-definite, no problem arises in applying the centroid method and Ward method. On the other hand, if S is not positive-semidefinite, i.e., with negative eigenvalues, the centroid method is problematic and should be regarded as an *ad hoc* technique without a sound theory, while the Ward method can still be applied with the remark that the result is *similar* to that from a regularized matrix $S + \alpha I$. Such a use of Ward method is not exceptional.

Model-Based Hierarchical Clustering

Generally, a model-based clustering means a mixture of densities in nonhierarchical clustering. As noted before, however, each clustering algorithm is on the basis of some mathematical model. In this sense, what type of a mathematical model is used should be addressed.

As described earlier, the single linkage is based on a weighted graph model, while the definition of a similarity/dissimilarity is arbitrary: it uses two different models. On the other hand, the centroid method and Ward method are based on the squared Euclidean distance model, and thus these methods use the single model. We moreover showed ℓ_1-space-based linkage methods in the previous chapter. Such a single model method can be further studied. For example, a linkage method using the cosine correlation has been developed by the author [7].

In relation to statistical models, the method of X-means [4] uses a special model different from the usual mixture of densities, whereby a divisive hierarchical algorithm has been developed. In contrast, Abe et al. [5] proposed a linkage method of agglomerative hierarchical clustering based on a mixture of Gaussian distributions. These directions can be further studied to derive new algorithms related to both the mixture of densities and also another model such as that used in X-means.

Hierarchical and Nonhierarchical Clustering

We focused upon agglomerative hierarchical clustering in this book, but generally the concepts of hierarchical and nonhierarchical clustering are closely related. An example is the two-stage agglomerative hierarchical clustering described in this book. Another example is a nonhierarchical method of *network clustering* for community detection [10] which has been derived from an original agglomerative hierarchical algorithm [6, 11].

Relations with Support Vector Machines

Ben-Hur et al. [12] proposed a method named *support vector clustering* in which regions of dense points are first extracted using a variation of the support vector machine, and clusters are identified from the extracted regions in the second stage. However, the method in the second stage is very briefly described and there are rooms to use different algorithms in the latter stage. For example, methods of K-means type or a divisive hierarchical algorithm can be used for this purpose. It is interesting to consider if an agglomerative hierarchical algorithm or an algorithm similar to DBSCAN can be used, and investigate their usefulness in specific application fields.

Studies of support vector machines in relation to semi-supervised learning led us to an idea of *transductive support vector machines* [13, 14]: an ordinary support vector machine (called inductive machine) divides the whole space into classified regions, while a transductive support vector machine prepares a set of *new sample points* in addition to *training samples*. After training, classes for only new sample points are obtained, while classified regions are uninterested and not obtained. Such distinction can also be found in clustering. K-means divide the whole space into Voronoi regions, as we observed in Chap. 5, while agglomerative hierarchical algorithm can generate clusters only for a given set of points. Thus we may compare K-means to inductive machine and agglomerative hierarchical algorithm to transductive one. There are not many studies in this direction, but the author [15] discussed inductive and noninductive agglomerative algorithms, and future studies can be expected.[1]

Methodological Aspects of Agglomerative Hierarchical Clustering

The final remark of this book should be the emphasis on theoretical and methodological aspects of agglomerative hierarchical clustering again. Generally, a theory and a methodology should be distinguished, as a method is based on a theory as well as other features like heuristics and practical considerations. In this sense, this book is concerned with the central part of methodology of hierarchical clustering, and the theory will continue to play the central role of methodological development in future studies in this field.

[1] An 'agglomerative transduction' algorithm is given in Wikipedia (Dec. 08, 2021):https://en.wikipedia.org/wiki/Transduction_(machine_learning).

References

1. G.N. Lance, W.T. Williams, A general theory of classificatory sorting strategies 1: hierarchical systems. Comp. J. **9**, 373–380 (1967)
2. J.C. Gower, A comparison of some methods of cluster analysis. Biometrics **23**, 623–637 (1967)
3. B. Everitt, *Cluster Analysis*, 2nd edn. (Heinemann, London, 1974)
4. D. Pelleg, A. Moore, X-means: extending K-means with efficient estimation of the number of clusters, in *ICML2000*
5. R. Abe, S. Miyamoto, Y. Endo, Hierarchical clustering algorithms with automatic estimation of the number of clusters, in *Proceedings of IFSA-SCIS 2017* (Otsu, Shiga, Japan, 27–30 June 2017), pp. 1–5
6. M.E.J. Newman, Fast algorithm for detecting community structure in networks. Phys. Rev. E **69**(6), 066133 (2004)
7. S. Miyamoto, Information clustering based on fuzzy multisets. Inf. Process. Manage. **39**(2), 195–213 (2003)
8. D. Wishart, Mode analysis: a generalization of nearest neighbor which reduces chaining effects, in ed by A.J. Cole, *Numerical Taxonomy, Proceedings of the Colloquium, in Numerical Taxonomy* (University St Andrews, 1968), pp. 283–311
9. S. Miyahara, S. Miyamoto, A family of algorithms using spectral clustering and DBSCAN, in *Proceedings of 2014 IEEE International Conference on Granular Computing (GrC2014)* (Noboribetsu, Hokkaido, Japan, 22–24 Oct 2014, 2014), pp. 196–200
10. V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks. J. Stat. Mech. **2008**, 10008 (2008)
11. M. Girvan, M.E.J. Newman, Community structure in social and biological networks. Proc. Natl. Acad. Sci. U.S.A. **99**, 7821–7826 (2002)
12. A. Ben-Hur, D. Horn, H.T. Siegelmann, V. Vapnik, Support vecctor clustering. J. Mach. Learn. Res. **2**, 125–137 (2001)
13. V.N. Vapnik, *Statistical Learning Theory* (Wiley, New York, 1998)
14. O. Chapelle, A. Zien, B. Scholköpf (eds.), *Semi-Supervised Learning* (MIT Press, Cambridge, Massachusetts, USA, 2006)
15. S. Miyamoto, Inductive and non-inductive methods of clustering, in *Proceedings of 2012 IEEE International Conference on Granular Computing* (11–12 Aug, Hangzhou, China, 2012) pp. 12–17

Index

S. Miyamoto, *Theory of Agglomerative Hierarchical Clustering*,
Behaviormetrics: Quantitative Approaches to Human Behavior 15,
https://doi.org/10.1007/978-981-19-0420-2