Chapter 3
Introducing Clustering with a Focus in Marketing and Consumer Analysis

Natalie Jane de Vries, Łukasz P. Olech, and Pablo Moscato

Abstract Clustering has become an extremely popular methodology for consumer analysis with many business applications. Mainly, when a consumer market needs to be segmented, clustering methodologies are some of the most common ways of doing so nowadays. Clustering, however, is a hugely heterogeneous field in itself with advances and explanations coming from many different disciplines. Clustering has been discussed in debates almost as heated as those about politics or religions, yet still, researchers and professionals agree on the methodology's usefulness in data analytics. This chapter provides the novice data scientist with a brief introduction and review of the field with links to previous large surveys and reviews for recommended further reading. The clustering contributions in this book focus largely on partitional clustering; hence, this is the type of clustering that will feature more prominently in this chapter. Besides sparking the interest of business and marketing researchers and professionals into this ever evolving methodological field, we aim at inspiring dedicated computer scientists and data analysts to continue to explore the wide application domains coming from business and consumer analytics in which clustering and grouping are making great strides.

Keywords Cluster analysis · Internal measures · External measures · k-Means · KNN

N. J. de Vries (✉) · P. Moscato
School of Electrical Engineering and Computing, The University of Newcastle, Callaghan, NSW, Australia
e-mail: natalie.devries@newcastle.edu.au; Pablo.Moscato@newcastle.edu.au

Ł. P. Olech
Department of Computational Intelligence, Faculty of Computer Science and Management, Wrocław University of Science and Technology, Wrocław, Poland
e-mail: Lukasz.Olech@pwr.edu.pl

P. Moscato, N. J. de Vries (eds.), *Business and Consumer Analytics: New Ideas*,
https://doi.org/10.1007/978-3-030-06222-4_3

3.1 Introduction

A good colleague of us once opened a seminar to computer scientists with a quote that reverberates in our heads: *"Clustering is a Religion, so I prefer not to talk about it"*. However, his talk was about a method to group data according to some similarities. His approach was different to others in that he used methods based on graph theory and combinatorial optimization. The message, nevertheless, was clear. There are many different techniques to group data and researchers seem to be particularly drawn to some mathematical models and they possess deep beliefs about them. We thus keep that in mind, we try to convey a message to newcomers in the field. There are many different ways to order and group data and practitioners need to be aware of the large variety of techniques that exist before passionately embracing only a small subset of them.

Clustering is a large methodological field, with many different approaches, algorithms and applications. Many good reviews exist and it is very difficult to select "the best one" as the criteria would depend on the reader and the application. There is indeed an extremely large body of literature in clustering, including many reviews of the area. For instance, the review by Jain et al. [57] titled "Data Clustering: A Review" is one of the most cited and comprehensive introductory reviews to the area of clustering. Jain added to this extensive review in 2010 with a more recent review of clustering in "Data clustering: 50 years beyond k-means" [56]. Some more reviews and introductions to data clustering have been done by Kaufman and Rousseeuw [61], Xu and Wunsch II [116] and Gan et al. [42] among many others. A recent survey on clustering methods based on combinatorial perspective was published by Levin [70] (and just this subset of the literature exceeds 200 selected references). Here we will add to this rich body of research with an emphasis on clustering in more recent business and marketing applications.

Even though clustering is such a popular area and methodology, and it is the source of many hard computational problems (i.e. those that we discussed in the first chapter as being NP-complete) which is why it continues to be further investigated today and why it is a prominent topic in this volume. We mentioned before that when a problem is NP-complete it is unlikely that efficient algorithms can be found for them. However, the mathematical model may be very useful for a practitioner (see, for instance [78, 94]). As a consequence, heuristics and metaheuristics are applied to find feasible solutions for these problems when we face large datasets.

This chapter will provide the reader with a useful introduction to the area of clustering, an idea of how and why clustering is an important (almost crucial) area to business and consumer analytics and provide specific examples of existing clustering methodologies used in these areas. A solid understanding of clustering methodologies, their inputs and their outputs, will go a long way to providing the data analytics novice with a solid base for further data-science exploration. For data science experts, it is useful to reflect on those methods most commonly used in marketing and in business and consumer analytics applications. First, we provide the novice data scientist with an introduction into clustering, how it works, why we

do it, the most common and popular methods, their pros and cons and a final focus on those methodologies particularly developed by business and consumer analytics researchers.

3.2 The Methodology of Clustering

In Clustering the objective is to assign labels to objects (or observations, or data points). A set of objects that have the same label (or labels) is said to be a "group" or a "cluster". The aim of clustering algorithms and heuristics is to achieve the best possible grouping. The outcome of the algorithm applied will thus depend on the choice of the similarity between the objects. It will also depend on the nature of the dataset. In this way, we can define a cluster as a collection of data instances (or objects) which are "similar" to each other and "dissimilar" to instances in other clusters [72]. To give a proper introduction to clustering we cover the questions of *"What is clustering?"* and *"Why do we cluster?"* We also provide a brief introduction to the main different types of clustering approaches including those most frequently used by business and marketing researchers and practitioners. Furthermore, as with any data analysis method, it is important to be able to evaluate or compare our results and we include a brief explanation of some approaches to do this.

3.2.1 What Is Clustering and Why Do We Do It?

There are many definitions for clustering, but in essence, it is a methodology with the purpose of organizing objects into groups (clusters) that are similar to each other (and dissimilar to other groups) based on a set of measurements/characteristics that are known about those objects. These objects could be consumers, patients, companies, products, images, genes and proteins in a biological network or any other dataset that could contain multi-attributed objects. Examples of the measurements or characteristics could be online consumer behaviour patterns, a set of answers to survey questions by clients, customer shopping patterns, financial investment patterns, gene expression patterns and so forth. Ideally, the objective is to group them, based only on the information provided in the dataset, without biasing the clustering method/algorithm how to group them, how many objects to group in each cluster, and ideally, without predicting the number of resulting groups and with the least amount of a priori parameters (more on this later in this chapter).

A "natural" clustering method would generally convey the meaning that the basic method is an unsupervised learning approach. Unsupervised learning, unlike with supervised learning techniques, has no a priori classes and no identification labels or partitions are given [72]. The objective of these techniques is to uncover the *natural groupings* of objects in a dataset.

It is thus very important to distinguish between supervised and unsupervised learning. Supervised *clustering* would not be clustering, it would in fact be *classification* [57]. Classification can be called supervised learning because, in a way, the method operates under *supervision* by being provided with the actual outcome of each of the training examples (it's class) [113] by the person operating it. Classification is used when there are existing classes to which the data objects belong to, such as classifying patients into disease subtypes in the area of bioinformatics or classifying products into their correct categories for an online retail webpage.

In many clustering applications in a business and marketing context, classes are usually not known a priori and analysts often aim to find and explore "the unknown" in their dataset without previous assumptions. Three main purposes of clustering in general are presented in Jain [56] (we quote):

- Underlying structure: to gain insight into data, generate hypotheses, detect anomalies, and identify salient features.
- Natural classification: to identify the degree of similarity among forms or organisms (phylogenetic relationship)
- Compression: as a method for organizing the data and summarizing it through cluster prototypes.

These purposes convey the drive to find out the "unknown" about a group of people, a set of topics or documents or any other dataset with underlying natural groupings.

Clustering has not been "championed" by one dominant discipline in particular, but rather it has received many contributions from many disciplines [92]. As a consequence, there are many different approaches, varying vocabularies and sometimes even multiple names for the same approach.

To make this area even more complex, there are endless amounts of applications for the ever-increasing number of clustering methodologies. A popular field in which clustering has been used extensively is in the medical, health and biological research domains. Considering clustering is an unsupervised learning technique, it is a useful tool for exploratory analysis of large datasets that we do not know a lot about. This is why it has been very useful for medical and health researchers in recent decades who may deal with millions of data points when analysing datasets of thousands of people with thousands of samples and variables. Business, finance, marketing, psychology and many other areas have rapidly caught up with the size of datasets that they produce. Consumer analytics is now a big contender for large data instances that can be generated in a very short period of time. This means that highly scalable and high performance clustering algorithms are now necessary for successful business and consumer analytics.

As Chap. 2 has already explained, in marketing, a common objective is to segment the market into similar segments of consumers. Market segmentation has therefore been the most common application for clustering methodologies in marketing and consumer analytics to date. However, with the ever-increasing size of data instances, data types, sources and applications, clustering exercises now have many more uses in business and consumer analytics. Other applications include exploratory research of a large dataset, inputs for other methods such

as recommender systems, visualization of a set of information, product analysis, logistics research and operational applications, financial analysis and prediction among many others.

In this volume some of these applications will be presented. In this chapter, however, we will focus on those methods of which the purpose is to uncover a *natural grouping*. That is, *clustering* (or partitioning) is the main purpose of the experiment. We need to look at the different clustering methodologies that already exist, particularly the most standard and well-known clustering algorithms and look at those methods most commonly used by consumer behaviour and business analytics researchers.

3.2.2 Different Types of Clustering

There are *many* different types of clustering approaches and many different names for highly similar clustering methodologies. The most well-known types of clustering include: partitional, density-based and hierarchical methodologies [100]. Most clustering approaches can be said to fall in one of these three categories, especially any method you will come across in this book. Some approaches such as *k*-means or nearest-neighbour clustering are described in further detail in Sects. 3.5.1 and 3.5.2, respectively. However, other clustering approaches besides the three "main" categories do exist. For instance, distribution-based clustering, which is somewhat similar to density-based clustering but rather than separating clusters by "low density" areas, it investigates the *distribution*, or spread, of the data points around the initial centre of the clusters. In this section we will discuss partitional, density-based and hierarchical clustering approaches as well as briefly introduce various other methodologies.

3.2.2.1 Partitional Clustering

The most commonly used clustering methods (at least in analysing consumer behaviour and for market segmentation) are partitional clustering methods. Partitional clustering separates a dataset into a set of disjoint clusters. With partitioning clustering, the objective is to maximize some function that measures the similarity of objects within the clusters, while at the same time, minimizing the similarities between objects assigned to different clusters. Partitional clustering procedures generate a single partition (as opposed to a nested sequence) of the data in an attempt to recover the natural grouping present in the data. There are many different partitional clustering methods and approaches, and in this chapter we will cover the most common ones and those commonly used by business and marketing researchers.

Partitional clustering can again be split into two sub-categories: *hard* and *fuzzy* clustering. With hard clustering, each data point is assigned to one and only one of

the clusters which means that there are well-defined separations between clusters. However, with real-world data, there are often no real and well-defined boundaries between groups of objects that are being investigated. Particularly when it comes to analysing humans (rather than physical properties) whose behaviours could not lead to characteristics which can be discretized. For this reason, fuzzy clustering has increased in use and popularity. With fuzzy clustering, each node (or object) has a variable degree of membership in each of the output clusters [57]. More details on fuzzy clustering approaches are provided in Sect. 3.6 of this chapter.

3.2.2.2 Hierarchical Clustering

Hierarchical clustering produces a dendrogram, a structure that is a nested sequence of clusters which look like a *tree* as depicted in Fig. 3.1. The y-coordinate of the horizontal line is the similarity of the two clusters that were merged, where the objects being clustered are viewed as singleton clusters. Similarity will be further discussed in Sect. 3.2.3. Hierarchical clustering uses either a top-down or a bottom-up algorithm which has implications on the way in which the data is separated by the algorithm [20]. They can also be referred to as either divisive (top-down) or agglomerative (bottom-up). Depending on the approach selected, a different outcome can be obtained because at the top of the dendrogram, there is one root cluster which covers all data points, whereas at the bottom of the hierarchy, there are singleton clusters (representing individual data points) [72].

With divisive hierarchical clustering, all the observations are assigned to a single cluster and the first step splits them up into two least similar clusters. These are then each split up again and so forth. This process is continued iteratively until there is one cluster for each of the observations at the bottom of the hierarchy. Oppositely, with agglomerative clustering, each observation is assigned its own cluster and then, based on similarity, they are combined into clusters. This is repeated until there is only one cluster left at the top. Agglomerative (bottom-up) clustering is more frequently used than top-down clustering.

A well-known and popular consumer-related example of hierarchical clustering that can be used to easily explain this method is that of an online retail store. Or even better, an online retail aggregation website which combines the products on offer from several other sites into one place. The problem at hand is to organize each product (for example, items of clothing), into subcategories according to a clothing category hierarchy. Each online retail store could have thousands of products in different colours and styles which could lead to tens of thousands of products combinations (data points) to be analysed for a web retailer aggregation website. The question is how should these products be organized into categories? One way is for a human team to manually organize each product into their correct category at the correct level in the hierarchy. However, the manual organization is extremely time-consuming and therefore very costly. A better way to do this would be through a hierarchical clustering algorithm that sorts through the products based on their features and meta-information. Or even visually analyses the image and uses

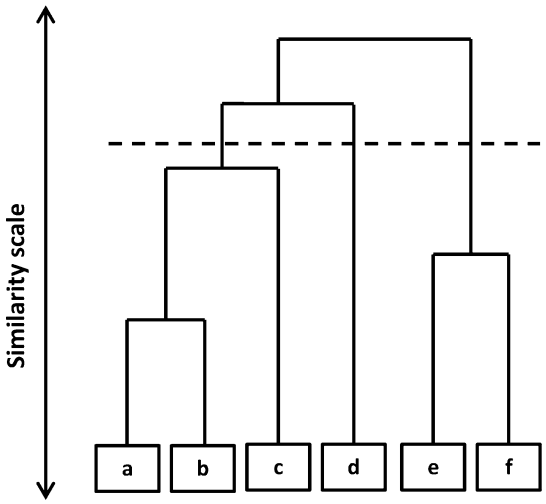

Fig. 3.1 A basic representation of a dendrogram showing hierarchical clustering. Each merge is represented by a horizontal line. The y-coordinate of the horizontal line is the similarity of the two clusters that were merged, where the objects being clustered are viewed as singleton clusters. The dashed line shows a section of the hierarchy that can be selected by the user for inspection. At this point we have three clusters: one cluster containing points *a*, *b* and *c*; one cluster containing only point *d* and one cluster containing points *e* and *f*

machine learning techniques to categorize the image (and give some extra help, for instance, by suggesting proper sets of tags according to the hierarchy level, etc.). This is one of the many examples of the use of hierarchical clustering. In essence, hierarchical clustering can be used for any set of information where there is some form of ranked order to be uncovered.

3.2.2.3 Density-Based Clustering

Density-based clustering uses a model to group objects according to specific density objective functions. Density is generally defined as the number of objects in a particular neighbourhood of a data objects. It is for this reason that density-based clustering is common in spatial applications of clustering. Density-based clusters are separated from each other by continuous regions of low density of objects. The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is an algorithm introduced by Ester et al. in 1998 which provides a density-based clustering suitable for managing spatial data [97]. This algorithm is probably the main density-based approach known to researchers. Figure 3.2 shows how density based clustering works. The two dense regions are clearly separated by less dense regions which is why they are in two separate clusters.

In the image, we can see data point alpha. With certain methods, such as *k*-means clustering (which we will explain in Sect. 3.5.1), the partitional clustering process starts by selecting centroids in the graph which will see its neighbouring nodes becoming part of its cluster. Therefore, if point alpha would be selected as an initial centroid, and the next nearest randomly selected centroid is point beta, then the node connected to alpha with a dashed line would likely be attributed to the smaller blue (lighter colour) cluster on the left rather than the red bigger cluster on the right. In instances like these, it is clear to see why and when density-based clustering would be the more appropriate method as in this case the density-based approach is more successful at uncovering the natural grouping (even through the peculiar shapes of the clusters). One limitation of density-based clustering methods is that they have trouble handling high-dimensional data [56]. This same aspect is the reason why other methods, such as *k*-means, have been more frequently and are so popular as they handle large datasets better.

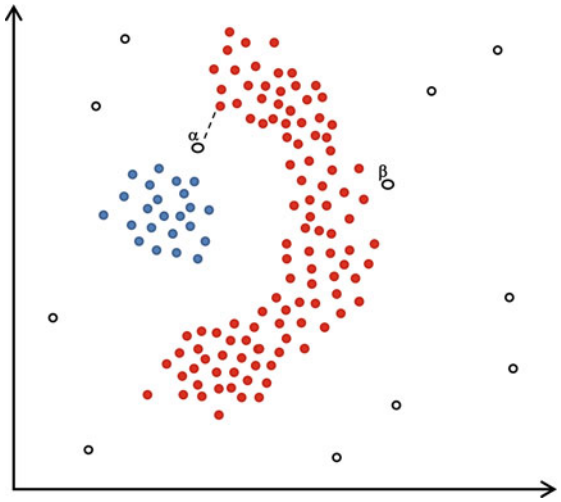

Fig. 3.2 This figure shows two clusters that are rather different in size and shape and are separated by an area of lower density. These two clusters would be identified much better using a density-based clustering technique than with, for instance, a *k*-means approach. If point alpha is selected as one of *k*-means initial centroids, it would likely attract the point connected via a dashed line to its cluster. This would in turn mean that the separation created by the lower density area would not be recognized by the method

3.2.2.4 Model-Based Clustering

Model-Based (MB) approaches can provide alternatives for heuristic approaches and have become more prevalent in the marketing literature since the early 2000s [105]. MB clustering approaches have the goal to optimize the fit between the given data and some mathematical model [48].

A mixture model corresponds to the mixture distribution that represents the probability distribution of observations in an overall population (dataset). Gaussian mixture models are some of the most commonly used model-based approaches. They investigate the number of Gaussian distributions evident in the data. A Gaussian distribution is simply a name for a "normal distribution" (or the "bell-shaped" distribution). The Gaussian distribution is a continuous function which approximates the exact binomial distribution of events. The Gaussian distribution is normalized so that the sum over all values of x gives a probability of 1. A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. Variations of Gaussian mixture models can also be used for hierarchical clustering applications [88].

Finite Mixture Models (FFMs) are a type of MB approach and FMMs have become more prominently used in marketing literature as they are able to simultaneously derive segments and segment-specific weights that relate to dependent variables (e.g. ratings) to a set of independent variables (e.g. product quality) as well as derive a unique regression model for each segment [105]. Those who are great advocates of FMM say that it is/should be a preferred approach because it is a formal statistical model (i.e. does not have a priori parameters such as *k*-means). A methodologically detailed review of Finite Mixture Modelling for the keen readers can be found in Melnykov and Maitra [81].

3.2.2.5 Hybrid Approaches

Of course, there are several other methods that combine some of the approaches above. One example is the hierarchical clustering approach based on "Arithmetic-Harmonic Cuts" of Rizzi et al. [94]. In this approach the method works by finding a partition of a set of objects that are linked by weighted edges of a graph. At each stage, an NP-hard optimization problem needs to be solved. The solution of this problem is a partition of the graph vertices in a way that minimizes an objective function (the weight of the arithmetic-harmonic cut). Then the two sets of vertices of the partition are recursively partitioned. The input is now the set of edges that is not part of the previous cut, thus again giving rise to another NP-hard optimization problem of the same type. This methodology combines partitioning within a "top-down" hierarchical clustering approach and it has been tested in different scenarios in [78, 94].

Another hybrid approach, again using hierarchical approaches is that of Fernández and Gomez where they include multidendrograms into an agglomerative hierarchical clustering approach [35]. They propose a variable-group algorithm groups more than two clusters at the same time when ties occur to deal with the problem of non-uniqueness when two or more distances between different clusters coincide during the amalgamation process.

The types of clustering and grouping approaches presented in this section are not an exhaustive list of ways to segment consumers. Many more grouping, regression,

clustering and statistical approaches exist and are being invented as this area of research continues to grow (including more combinations of hybrid approaches). However, the ones presented in this section provide a good basic understanding of the most commonly used clustering, segmentation and grouping methodologies.

3.2.3 Distances and Similarities in the Context of Clustering

When reading about clustering, grouping and classifying, you will hear and see the words "similarity" and "distance" come by many times. In many cases, the input for a clustering algorithm is not a similarity matrix. Instead, each object will have a set of features (also known as variables, attributes or characteristics) which may be extremely unique to that object, or very similar to other objects of interest. It is this information that is needed for many clustering algorithms. In the case of segmenting consumers, customers, users or followers, products, these are your objects and the variables relating to them (e.g. purchase patterns) are the features. In some cases, distance and similarity metrics will usually be (or will be normalized to) a value between [0,1]. With similarity metrics, a value closer to 1 means *more similar* and with distance metrics a value closer to 1 means *more distant* from each other.

The selection of a proper distance or similarity metric between objects then becomes an interesting issue in itself. There are many different metrics available for use and which one you choose, depends on your dataset, the context and the nature of your data among other aspects. In fact, a whole "Encyclopedia of Distances" has been published [28] and in no effect could we match this here. Instead, we provide a brief overview and introduction to the most commonly used distances and similarity metrics used for clustering. Further, we introduce some of the metrics that are used by subsequent chapters in this volume.

Distance matrices can be used to generate other graphs such as proximity graphs, relative neighbourhood graphs (RNGs) or any other distance-based graph such as those introduced in Chap. 4.

3.2.3.1 Distance and Similarity Metrics

Given a set of objects, a *metric* (or *distance function*) is a non-negative function that defines how far apart each pair of objects of a set are. Formally metric $d(a, b)$ is a function that for objects a and b:

1. returns 0 as the distance from point to itself, i.e. $d(a, b) = 0 \Leftrightarrow a = b$ (*identity of indiscernibles*),

2. distance between any two points is the same, regardless from which point we start to measure, i.e. $d(a, b) = d(b, a)$ (*symmetry*),
3. distance between any two points is lower or equal to the distance between the same points, but measuring through any third point c, i.e. $d(a, b) \leq d(a, c) + d(c, b)$ (*triangle inequality*).

For objects in an Euclidean space, a common and old metric to use is the *Euclidean distance*. The Euclidean distance is described as the "ordinary" (or straight-line, "like the crow flies") distance between two points in an Euclidean space [28]

$$d_{euc}(a, b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}, \tag{3.1}$$

where $d_{euc}(a, b)$ is the Euclidean distance, $a = [a_1, a_2, \ldots a_n]$ and $b = [b_1, b_2, \ldots b_n]$ are the n-dimensional objects between which we want to calculate distance and n is the number of features that correspond to the points. How "good" the Euclidean distance is at finding the most appropriate partitions of a dataset depends on the circumstances. As our example already showed in Fig. 3.2 however, sometimes a straight-line distance may not always be the best approach for finding the most natural underlying groupings in a dataset. Therefore, many other metrics have been suggested since to deal with other datasets and instanced for which the Euclidean distance may not be the most appropriate. A variation is the "squared Euclidean" which is commonly used for k-means clustering. K-means is implicitly based on pairwise Euclidean distances between points, because the sum of squared deviations from each centroid is equal to the sum of pairwise squared Euclidean distances divided by the number of points. Hence, the k-means approach needs to use the squared Euclidean distance rather than the standard Euclidean distance.

Another distance is called the *Manhattan distance* d_{cbox}:

$$d_{cbox}(a, b) = \sum_{i=1}^{n} |a_i - b_i|. \tag{3.2}$$

Figure 3.3 shows how the Euclidean distance and the Manhattan distance work and how they differ from each other. The Manhattan distance gets its name from "going around the city block"; something that city-dwellers in Manhattan probably know all too well and accordingly also gets referred to as "city-block". The Manhattan distance function finds the distance that would be travelled to get from one data point to the other if a grid-like path is followed (i.e. on a vector space it is the sum of the absolute value of the differences on each coordinate dimension).

Both Euclidean (Eq. (3.1)) and Manhattan (Eq. (3.2)) can be generalized by the *Minkowski distance* d_{min}

$$d_{min}(a, b) = \left(\sum_{i=1}^{n} |a_i - b_i|^p \right)^{\frac{1}{p}}, \tag{3.3}$$

which is also called as L_p. In case $p = 1$ (L_1) it is a Manhattan distance, when $p = 2$ (L_2) it gives Euclidean.

Another distance is the *Chebyshev distance* (or Tchebychev distance). This is a metric defined on a vector space where the distance between two vectors is the greatest of their differences along any coordinate dimension. It is also called as a *Chessboard Distance*, because it shows the minimum number of king figure from one square to another. This distance is equivalent to Minkowski distance (L_∞) with $p \to \infty$ (Eq. (3.3)).

A *Similarity Measure* is a function assessing resemblance between objects. Contrary to a distance function, a similarity measure is a real-valued function that can give negative values and it is commonly assumed that similarity is an inverse of distance. More similar objects (high value of similarity) would have lower distance between them, and distant objects (higher metric value) have lower similarity. Even though similarity measure is symmetric it does not necessarily meet other distance properties (provided at the beginning of Sect. 3.2.3.1). Furthermore, in most of the cases, a similarity function is not additive. In such a case, adding or subtracting similarity values or computing an average is invalid, but similarities can be multiplied (scaled) for better comparison purposes, for instance.

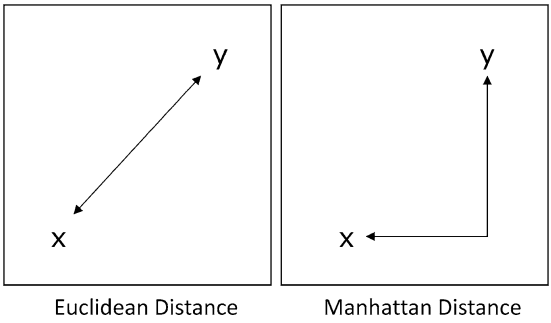

Fig. 3.3 A figure representation of the Euclidean and Manhattan distances which shows how the Euclidean distance is "how the crow flies" and the Manhattan distance "going around the city block" following a grid-like path between two data points

An example of a similarity measure is *Cosine Similarity* (s_{cos}) that uses vectors created from data points (a and b) in such a way that a vector begins in an arbitrary point (usually $z = (0, 0)$) and ends in point indicated by a or b. It is formulated as follows:

$$s_{cos}(a, b) = \cos(\Theta) = \frac{a \cdot b}{||a||\ ||b||}, \tag{3.4}$$

where Θ is the angle between vectors created by a and b, $||a||$ and $||b||$ are the lengths of vectors $d_{euc}(a, b)$ and $d_{euc}(a, b)$, respectively, $a \cdot b$ is a *dot product*

$$a \cdot b = \sum_{i=1}^{n} a_i b_i. \tag{3.5}$$

Notably, the cosine measure considers similarity as an angle (Θ) between the vectors. The magnitude of the vectors is not considered. Vectors with the same orientation are regarded as similar giving a maximum value $s_{cos}(a, b) = 1$, whereas perpendicular vectors have 0 similarity. On the other hand opposite vectors will result in lowest similarity $s_{cos}(a, b) = -1$.

Cosine similarity is often used in information retrieval and text mining [62, 63]. From business analytic perspective this could be valuable in product review analysis [11] or automatic product recommendation. Assuming that two text documents are described by vectors of word occurrences, this measure will give the similarity between those documents, regardless of their sizes. In other words, cosine similarity indicates how similar the subject is of the two given texts. Furthermore, this measure is efficient to calculate on *sparse vectors* (vectors having many zeros) since only non-zero values are important.

Since similarity and distance are opposite to each other, distance can be transformed into similarity, but it is important to note that a reverse operation is not always possible. As a consequence, Euclidean distance can be used to describe objects' similarity, but cosine similarity cannot be used as a distance function. Assuming that distance takes values in the closed interval [0, 1], a common way to transform distance function value (d) into similarity (s) is

$$d = 1 - s. \tag{3.6}$$

Another way to compute similarities is to use correlation metrics. These will in fact produce a number that relates to similarity (correlation) between points. Two of the most common correlation metrics are Spearman and Pearson correlation. The Pearson correlation coefficient should be used to cluster objects with similar behaviour patterns as those with opposite behaviours are assigned to different clusters. A variation of the Pearson correlation is the Absolute Pearson correlation where the absolute value of the Pearson correlation coefficient is used; hence, the corresponding distance lies between 0 and 1. Spearman correlation clusters together those objects who's profiles have similar shapes, that is, their trends are similar but the actual values may be quite different. The authors have previously used various distance metrics and correlation metrics in a clustering study and compared the effects on the outcome of using various similarity and distance metrics [75].

3.3 Measuring Clustering Quality

Some clustering methods are designed to take advantage of extra sources of information. For instance, if the number of clusters a user expects to find in the data is known (if that number is known a priori or can reasonably be guessed or predicted within reasonable bounds). In other cases, the number of clusters could be a user-defined request (due to reasons which do not belong to the data study in question). This information can be provided either explicitly, by the value of the k parameter as it is in the *k-means* algorithm (described in Sect. 3.5.1) or more implicitly, by providing a density of clusters that the method will be looking for (e.g. [34]). Moreover, considering Fig. 3.8 (in a section further in this chapter) it is visible that the k parameter has a huge influence on the method's result. After all, the vast majority of clustering algorithms have some parameters that should be (more or less) carefully tuned. This situation raises a question, *how to choose the appropriate k (and possibly other parameter's) value?*

One obvious technique is to execute the method with different parameter values and then *compare* their results in order to finally choose *"the best one"* (as the authors were able to do in [75], thanks to class labels that could be used for post hoc statistical analysis), but the process of a comparison might not be so straightforward. There could be a temptation to simply look at the results and subjectively decide which clustering result is better, but considering a real-life situation where the number of features is more than three, and the number of points is 10,000 or more, attempting to do this via a visualization approach becomes difficult. Furthermore, what happens when there are 1000 clustering results? One possible solution is to visualize the clustering in a grid of pairwise two-dimensional plots. In such a grid, every feature is plotted against each other giving a set of plots that are easier to analyse. The plot's size grows quadratically with the number of dimensions. For instance, having five-dimensional data, the grid would contain $5^2 = 25$ plots. How hard and time-consuming would it be to analyse such plots? Even though it is possible to reduce the dimensionality of the input data by aggregation or to use a dedicated method, such as *Principal Component Analysis (PCA)* [12], there is a risk of losing valuable information.

Considering all mentioned problems, it becomes clear that there is a general need for a tool that assesses which clustering is better than others, and it is desirable for the tool to be fully automated. Due to this, researchers devised the idea of *Quality Measures*. These are mathematical formulations aiming at expressing the quality of a clustering result as a number in a predefined interval. Quality measures will be discussed more in Sect. 3.3.1. Having two clustering results, we can compute their quality using a particular measure, and by comparing these measures, we can decide which one is of *higher quality*. This process is going to be discussed in Sect. 3.3.2. However, what does it mean to have *better quality*? One could define quality in

a variety of ways, and because of this, there are many measures and measure paradigms aiming at assessing clustering. To choose the most suited measure, in Sect. 3.4 we elaborate on types of clustering measures and introduce the most common measures in a systematic way.

As will be further elaborated on in Sect. 3.6, apart from *hard* (*crisp*) clustering, there is a *fuzzy* approach, however in this section, we are going to focus only on the assessment of non-fuzzy non-hierarchical clustering results. Nevertheless, most of the presented measures can be easily applied to other types of clustering as well.

3.3.1 What Is a Quality Measure?

As was stated, a quality measure is a function that gives a quantitative rating to the outcome of a clustering algorithm. A clustering $\mathcal{C}$ is a set of clusters (i.e. $\mathcal{C} = \{C_1, \ldots, C_p\}$), where C_i is the ith cluster of the clustering. An example of such a measure could be the average size of clusters:

$$Q_{avgSize}(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{\forall C_i \in \mathcal{C}} |C_i|, \tag{3.7}$$

where $Q_{avgSize}$ is a measure, $\mathcal{C}$ is a clusterization result, $|\mathcal{C}|$ is the number of clusters in the particular clustering under study and $|C_i|$ is the number of objects that belong to cluster C_i.

Assume that we have two different clustering outcomes $\mathcal{C}_1$ and $\mathcal{C}_2$ that are rated by $Q_{avgSize}$ in the following way:

$$Q_{avgSize}(\mathcal{C}_1) = 5, \qquad Q_{avgSize}(\mathcal{C}_2) = 17.5.$$

If we would be interested in configuring algorithm parameters in such a way to get (on average) smaller groups, then we would prefer clusterings to have lower $Q_{avgSize}$ values so that we would pick $\mathcal{C}_1$ as a final result. That interest could be present when creating personalized music recommendations for customers based on a music database. That if someone has listened to one of the songs, the remaining songs in the cluster will more likely suit their taste. As such, smaller clusters will reduce the chance of suggesting unrelated songs. Contrary, if someone else would prefer bigger groups, e.g. when trying to find main general music taste among all customers, then it would be better to pick $\mathcal{C}_2$. As a third option, if someone is not interested in the group sizes, but in a different aspect of clustering, they should rather look for another measure that would reflect his demand on the clustering.

This example shows that *better quality* can be treated differently by people and sometimes the clustering that is right for one application does not fit the needs of another. Due of this, there are many different quality measures, promoting various aspects of clustering and the meaning of *the best* clustering should be indicated by the task's specific requirements on a case-by-case basis (its application). Therefore, in Sect. 3.4, we present several measures and will explain the intuition behind them.

The usage of a quality measure should be application driven. Due to this, it is critical to understand both the clustering problem and the possible quality measures that could be used. It is vital to do this before starting to implement a solution. In order to do this, there are several important questions to answer before using any particular quality measure:

1. What clustering aspects are promoted by that measure and what aspects are ignored?
2. What are the possible values of that measure (i.e. what are the measure's boundaries)?
3. What values we are looking for and what we would like to avoid? (In other words, what values indicate better and worse clustering?)

The first question forces us to *understand* the measure. After doing this, it is much easier to address the remaining questions. As we have already noticed, the $Q_{avgSize}$ (Eq. (3.7)) measure is focused only on the average group sizes. So this measure will not tell us anything about the number of clusters or their distribution. At this point, we can decide whether this quality measure suits our needs or we should rather search for another one.

The second question can tell us what values we can expect to obtain. Sometimes the values vary between 0 and infinity (∞), in other words, the boundaries are, theoretically, $(0, \infty)$ (e.g. any positive real number). If it can't reach 0 or $[0; \infty)$ if it can. However, in the general case, the boundaries may be different, e.g. $(-\infty; \infty)$, $(-\infty; 0)$, $[1; \infty)$ or even between some arbitrary values (e.g. [3; 50.5]). By looking at our measure in Eq. (3.7), we can derive that it is impossible for the measure to be negative since clusters sizes cannot be below 0. This measure can give 0 only in a situation when all the clusters will be empty, so no points were grouped. For the sake of simplicity, we can assume that it is not a valid clusterization, and by this, the lowest possible value for our measure is not going to be equal or below 0. What about the highest value? It could be any positive number. One can imagine that if we have only one massive cluster, e.g. 10^{25} points, then the $Q_{avgSize}$ value would be very high (exactly 10^{25}). So the upper-bound is ∞ which gives the final boundaries of this measure as: $(0; \infty)$.

By now, we know what values to expect, but what numbers indicate a preferable result? This leads to the last question. As it was stated earlier in this section, sometimes we would like to *minimize* that quality measure value and

sometimes *maximize* it with respect to the provided boundaries. In most of the cases, we are trying to reach the boundary value (either upper or lower one). However, with some other quality measures, we should reach a specific value within the boundaries, e.g. 0 or 4. Coming back to the $Q_{avgSize}$, being interested in smaller clusters, what we would be doing is to *minimize* the value of $Q_{avgSize}$.

3.3.2 How to Use a Quality Measure?

Now it is the time to use a quality measure in practice. Assume that we have the well-known *Iris dataset* [37] and we would like to tune k-means algorithm on this data. As it was stated in Sect. 3.5.1, the most important k-means parameter is actually the value of an integer $k > 0$.

The dataset that we are going to use contains measurements of three species of Iris Flowers: *Iris Setosa, Iris Virginica* and *Iris Versicolor*. For each class (species) we have 50 samples (varieties), and for every example we have four features measured, i.e. *Sepal Length, Sepal Width, Petal Length* and *Petal Width* (or five, if we consider the class being the target feature as discussed in the first chapter). For the sake of simplicity, we are going to use only two dimensions: *Sepal Length* and *Petal Length*. The dataset is visualized in Fig. 3.4. From the figure, one can see that *Iris setosa* (green triangles) samples are easily distinguishable from the rest, whereas when it comes to the *Iris Versicolor* (red circles) and *Iris Virginica* (blue squares), it is not that easy. In mathematical terms we could say that *Iris Setosa* is *linearly separable* from both the *Iris Versicolor* and *Iris Virginica* samples, which means (without going deeply into the details) that one could easily draw a straight line on Fig. 3.4 to have all the *Iris Setosa* samples on the one side of the line and all the others on the other side. On the other hand the *Iris Versicolor* and *Iris Virginica* species are ***not*** *linearly separable* because it is impossible to draw a straight line to separate those two classes using only these two features (see Sect. 1.6.3 if you are interested to know what to do in case that linear separability is not possible).

Our possible goal is uncover from the data the number of natural clusters present, which in turn in order to create a clustering outcome that would be the most informative. Such a clustering can then help us to *classify* unknown data samples as belonging to one of these classes. Since we know the target feature (what is their species), we expect to have some correlation between the result of our clustering and the target feature values. If there is total agreement, perhaps we would be to have only three clusters with 50 samples in each one.

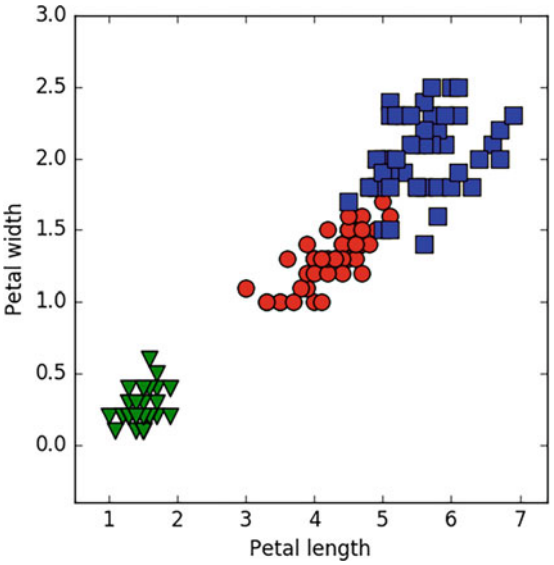

Fig. 3.4 Visualization of the Iris dataset. Each shape represents different species. Green triangle—*Iris Setosa*, red circle—*Iris Versicolor*, blue square—*Iris Virginica*

The quality measure we can use is *Within Cluster Sum of Squares* (*WCSS*) that is defined as

$$WCSS = \sum_{C \in \mathcal{C}} \sum_{x \in C} d\left(x, x\left(C\right)\right)^2, \tag{3.8}$$

where C is a particular cluster, $\mathcal{C}$ is a set of all clusters, x is a particular element attributed to cluster C, $x(C)$ is a centroid (central point) of cluster C and $d(x, x(C))$ is the distance measured between point x and $x(C)$. As a distance measure, for this problem we use the *Euclidean Distance* (Eq. (3.1)), but depending on the application others could be used, e.g. *Manhattan Distance* or *Cosine Similarity Measure*. Having defined the quality measure, the answers to three questions stated in Sect. 3.3.1 are straightforward.

1. This measure promotes clustering, where points in every cluster are close to the cluster centre (dense clusters) so, as a consequence, the clustering result with only one-object clusters is the most desired solution from the measure perspective. However, from the user perspective, in most of the cases, it is not desired because this does not give any knowledge about the data. Furthermore, this measure ignores the separation between clusters or how the clusters are distributed in the feature space. Using this measure, we do not care about the spatial relation between clusters which might sometimes be important (as stated in Sect. 3.4.1). Also, the number of clusters is not taken into consideration by this measure, so more clusters will, in general, give a better score since it gives a better fit into data, lowering the average distance of points to their centres.

2. The boundaries are $[0; \infty)$, where $WCSS = 0$ can be reached in a situation, where there are no clusters or every point is in its own cluster. Any other situation should end up with a positive value.
3. In general, we would like to *minimize* the value of the measure, because the lower the value is, the denser and more consistent the clusters are. However, on the other hand, we would like to avoid the situation when $WCSS = 0$.

After giving these answers, we can conduct a series of experiments (using k-means, for instance) with different parameter values (for k) that can vary from one up to four. The results, together with the measure value, are presented in Fig. 3.5. From this figure, we can see that for $k = 1$ there is only one cluster containing all the samples. Because of this, there are many objects that are far from the cluster centre, and because of that, the $WCSS$ value is the highest. On the other hand, when $k = 2$, the result correlates with our understanding that there is one group of samples that is highly different than the other. We point out, however, that one sample near the centroid of the blue group has been attributed to the other group. This shows a divergence between the result of k-means and our intuition, since we perceive that many "blue elements" are actually closer to that sample attributed to the other cluster.

Focusing only on the value of $WCSS$, the solution with $k = 2$ is preferable over the one with $k = 1$. Despite the fact that the result for $k = 2$ complies with human intuition, we know that experts agree in identifying at least three different iris species, not two. Following that fact, when $k = 3$ the result is similar to the *ground truth* (the expert opinion) shown in Fig. 3.4 except several samples. This is because k-means in its basic form is not capable of separating linearly non-separable groups (as Fig. 3.2 also shows).

Perhaps the most important observation is that, even though the number of created groups and the result of the k-means method is similar to Fig. 3.4, the quality measure $WCSS = 35.39$ is not the lowest of the four. The measure gives the minimum value when $k = 4$ and it is $WCSS = 19.55$, this is obvious when one looks at Eq. (3.8) and our answer to the first question given above. The more the clusters in a clustering result, the shorter the distances of points to their centres become. Since we should minimize the $WCSS$ value, if we follow this criteria we should choose the clustering with $k = 4$. Is it a solution that we are looking for? From Fig. 3.5, we can see that four-cluster solutions just fit additional cluster in the right-upper part of the plot. These three groups together do not look like well-separated groups.

From the above paragraph we can see that the procedure of choosing the best clustering is far from being trivial. The used measure, even though it focuses on the coherence of the clusters, ignores the separation between groups, thus simply promoting results with more clusters. Some researchers [45] say that, when operating with a quality measure with such a tendency, one should plot the obtained quality measures against the value of the parameter that we are tuning. Then one should look for a point giving the biggest change (increase or decrease) on the plot. This point is sometimes referred as a *knee* or *elbow point*, and the corresponding parameter value

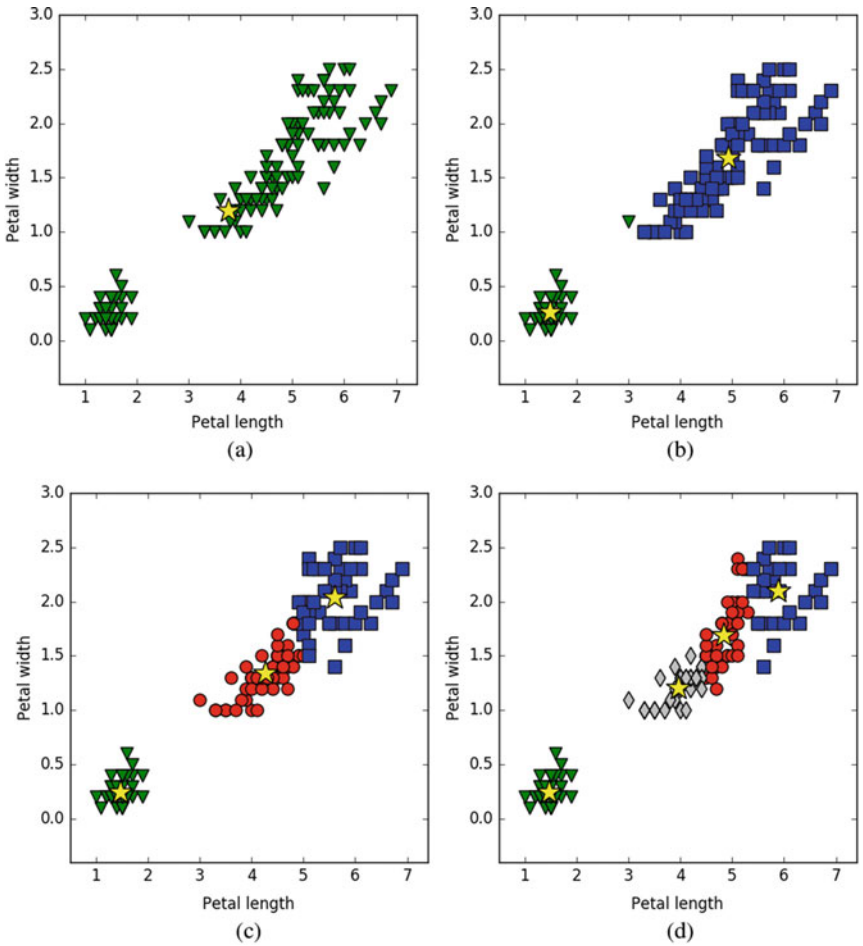

Fig. 3.5 Visualization of k-means results on the *Iris dataset*, where $k = 1, 2, 3, 4$. Different markers represent different clusters, and yellow star indicates cluster centres. The $WCSS$ values are as follows: (**a**) for $k = 1$ the $WCSS = 550.64$, (**b**) for $k = 2$ the $WCSS = 86.40$, (**c**) for $k = 3$ the $WCSS = 35.39$ and (**d**) for $k = 4$ the $WCSS = 19.55$

for that point should be chosen. From the plot for our example, shown in Fig. 3.6, one can see the *knee point* exists at $k = 2$. So, using this technique, this value should be our choice. However, what if there is no significant change in the plot? This could be an indication that there is no clear clustering structure in the data. Due to these problems, in a practical application, it is recommended to consider either using *several* quality measures that focus on different aspects of clustering or find a measure that combines several aspects in one equation [83]. This gives a more comprehensive view on the obtained results. Luckily there are many different quality measures from various groups that try to describe clustering from different perspectives.

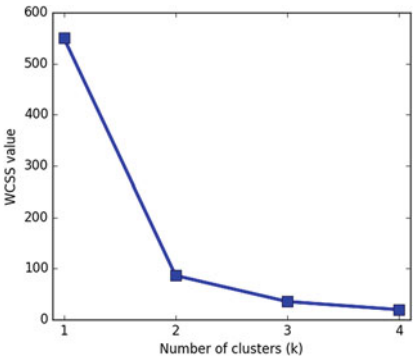

Fig. 3.6 Within cluster sum of squares (WCSS) of clusterization results for different values of k. The knee point is visible for $k = 2$

Moreover, based on the visual assessment of obtained results, and from the existence of the *knee point*, one would select a result with $k = 2$ which is not consistent with the real number of three groups. The answer to this problem is more complex. This situation could be caused by several factors:

- a weakness of the clustering method in discovering the proper shape of the groups,
- a weakness of the quality measure in selecting the clustering that is compliant with true classes assignment,
- the features used to describe samples make it impossible to distinguish between objects from different groups (i.e. the available features and data do not *tell the whole story*).

In most cases, the problem is a combination of all the three factors. That is why, when it is possible, one should choose the final clustering relying not only on the solution indicated by the used quality measure. An inspection of several (e.g. top five) clustering results is highly beneficial. Furthermore, sometimes it could happen that even if we manage to obtain a clustering with a "proper" number of clusters, the clustering does not perfectly match the ground truth. This situation is often caused by the clustering method, and a simple solution is to use several methods, produce a few results and then conduct a detailed analysis of what they would bring in terms of adding knowledge to the user. On top of that, the measures and dataset can simply make it impossible to obtain a clustering that fully complies with our expectations.

To sum up, the problem of choosing the best clustering result is complex. The best result indicated by the quality measure may not necessarily be the best for other people, and it may not even appear to be the best among all obtained results. This stems from the fact that clustering is unsupervised, and the assessment process is both subjective, and application dependent. The best clustering result for one application could be the worst for another.

3.4 Classification of Quality Measures

The central point in the assessment of clustering is the quality measure. Due to this, there are different measures and measure paradigms [45]. Since the number of measures is high, there are several ways to organize them into groups. The main division considers *Internal* and *External* measures. It was made based on whether we have any additional (*external*) knowledge about the data or we base only on the (*internal*) structure of clusters. In most of the cases, the external knowledge is *the ground truth* assignment with which the measures will try to compare the clustering assignment. Several external measures are provided in Sect. 3.4.2. On the other hand, internal measures are focused on the way that the points are clustered. Since we want to have clusters that are both compact and separated from each other, the internal measures often balance these two requirements. Example internal measures are provided in Sect. 3.4.1. The presented measures are a subjectively chosen representation, based on how frequently they appear in the literature.

It is also worth noting that there are other different ways to divide measures into groups. In addition to *internal* and *external* measures, one can also distinguish measures for *crisp* or *fuzzy* clustering or *hierarchical* clustering measures.

In the following sections, we will present the most commonly used quality measures in the field of cluster analysis. In order to make them easily readable all presented equations will obey the following variable definitions:

X	the set of all data points,
$\mathcal{C}$	the set of all clusters in a particular clustering,
C	a particular cluster in $\mathcal{C}$, it can have an additional lower index indicating which cluster it is, e.g. C_j is the jth cluster,
X_C	the set of objects in cluster C,
x	a particular object from a cluster, it can have an additional lower index indicating which point it is, e.g. x_j is the jth observation,
$d(a, b)$	distance between objects (or centroids) a and b,
K	set of all classes (i.e. from a *ground truth* assignment),
k	a specific element from K (a class label),
X_k	set of all objects of class k,
$\overline{X_C}$	the centroid of a cluster C,
$\|S\|$	the number of objects in a set S.

3.4.1 Internal Measures

One of the simplest internal measures is *Within Cluster Sum of Squares* (*WCSS*). It was used in Sect. 3.3.2, and its formula is in Eq. (3.8). However, as shown in the

mentioned section, its main drawback is that it focuses only on the *compactness* of clusters, ignoring the *separation* that exists between pairs of clusters. Another simple measure is *Within-Between Index (WBI)*. It is formulated as follows:

$$WBI = \frac{\max\limits_{C \in \mathcal{C}} \{ \max\limits_{x_i, x_j \in X_C} \{d(x_i, x_j)\}\}}{\min\limits_{C_a, C_b \in \mathcal{C}, a \neq b} \{d(\overline{X}_{C_a}, \overline{X}_{C_b})\}}. \tag{3.9}$$

It expresses the ratio between compactness (in the numerator) and separation (in the denominator). The compactness of the clustering is measured by the distance between the farthest pair of points belonging to one cluster. On the other hand, the separation part is the minimum distance from cluster centroids among all the possible pairs of clusters. Therefore, this measure should be minimized, and its values fall into $[0; \infty)$.

Another quality measure that utilizes the compactness–separation ratio is *Dunn Index (DI)* [32]. It has several different forms in the literature [6] providing different characteristics, e.g. robustness to noise. One exemplar form is:

$$DI = \frac{\min\limits_{C_a, C_b \in \mathcal{C}, a \neq b} \{ \min\limits_{x_a \in X_{C_a}, x_b \in X_{C_b}} \{d(x_a, x_b)\}\}}{\max\limits_{C \in \mathcal{C}} \{ \frac{1}{|X_C|(|X_C|-1)} \sum\limits_{x_i, x_j \in X_C, i \neq j} d(x_i, x_j)\}}, \tag{3.10}$$

that expresses the ratio between the closest clusters, expressed as the shortest distance between points from different clusters, in the numerator (separation) and the biggest pairwise distance in between points in a cluster in the denominator (compactness). Moreover, the value in the denominator is averaged guaranteeing that the size of the cluster is not taken into account. The value of DI varies from 0 to ∞, and the index should be as high as possible.

Both of the measures WBI and DI promote solutions with dense, well-separated clusters. It is also worth noting that they concentrate on the worst aspects of a clustering. They utilize the least separated clusters and the least compact one. It follows the rule: *a clustering is as good as its weakest part*. However, sometimes this may not be the case. In such situations, one could use *Davies–Bouldin Index (DBI)* [23, 27] or *Calinski–Harabasz Index (CHI)* [14, 27]. They take into account the sum or average of all the clusters but then it suffers from the fact that their result is biased when assessing clusterings with varying diameters. Other internal measures that are worth considering are *Silhouette Index (SI)* [95], or more recently *SD Index* [44], *S_Dbw* [46] or *Clustering Validation Index Based on Nearest Neighbours (CVNN)* [73]. Additionally, some internal measures have been updated in order to perform better, e.g. DBI [64].

3.4.2 External Measures

External measures commonly use a *ground truth* cluster assignment. In the previous example, the three species of Irises is a user-defined characteristic (target feature). External measures would then use this labelling to evaluate the quality of a clustering method.

3.4.2.1 Confusion Matrix

The external measures can be divided into several types. One group relies on the computation of a *confusion matrix (error matrix)* that serves as a data structure to quantify the performance of a classification method. This group is sometimes known as *pair counting* methods [93]. The confusion matrix consists of four numbers called the *true positive*, *true negative*, *false positive* and *false negative* rates. They can be calculated by considering all possible pairs of objects in the sets and their assignments in the clustering result. We will also use the ground truth information about those objects [27]. In order to calculate the rates above, one has to consider all $\frac{n(n-1)}{2}$ possible pairs, where n is the number of objects. Assuming that the clustering result we want to assess is $\mathcal{C}_a$ and the ground truth grouping is $\mathcal{C}_{gt}$ one could compute

- *True positive (TP)* as a number of pairs that belong to the same cluster in both $\mathcal{C}_a$ and $\mathcal{C}_{gt}$,
- *True negative (TN)* as a number of pairs that **do not** belong to the same cluster in both $\mathcal{C}_a$ and $\mathcal{C}_{gt}$,
- *False positive (FP)* as a number of pairs that belong to the same cluster in $\mathcal{C}_a$ but **do not** belong to the same cluster in $\mathcal{C}_{gt}$,
- *False negative (FN)* as a number of pairs that **do not** belong to the same cluster in $\mathcal{C}_a$ but belong to the same cluster in $\mathcal{C}_{gt}$.

The *TP* and *TN* are expressing the ability of our clustering method to properly group objects that should possibly be together and separate objects that are assumed to belong to different groups. We can call them *good decisions*. On the other hand, *FP* and *FN* are showing how many "mistakes" were made by our model. In Statistics, *FP* is often referred to as the number of *type I errors*, and *FN* as the number of *type II errors*.

Having computed all the values of the confusion matrix, one can calculate different external measures, e.g. *Specificity* (*True Negative Rate*), *Accuracy*, *Precision*, *Recall*, *F-Score* [107], *Jaccard Index*, *Rand Index*, *Fowlkes–Mallows Index* [40]. All of these indices can be found in [27]. Another external measure worth to be noted is *Matthews Correlation Coefficient* (*MCC*) [79] with its later generalization into multi-class version in [43, 58]. The advantage of using this index is its invariance to different class sizes. An example of using this measure in the marketing domain

is presented in Chap. 20 of this volume. The most commonly used quality measure is the *F-Score* [101], also known as *F-Measure*. It is the harmonic mean of the *Precision* (*Prec.*) and *Recall* (*Rec.*) (also known as *Sensitivity* or *True Positive Rate*) and it is computed as follows:

$$Prec. = \frac{TP}{TP + FP} \tag{3.11}$$

$$Rec. = \frac{TP}{TP + FN} \tag{3.12}$$

$$F\text{-}Score = \frac{2 * P * R}{P + R} = \frac{2 * TP}{2 * TP + FP + FN}. \tag{3.13}$$

The bounds of the *F-Score* is the closed interval $[0, 1]$ and the larger the *F-Score* gets, the more similar the model is to the ground truth. The intuition behind formula (3.13) is that when clustering, we would like to get high precision and recall, but since these two measures are in contradiction to each other, we will take a harmonic mean of them.

One notable fact about the *F-Score* is that it does not explicitly depend on *TN*. The consequence of this is that the assessment is positively influenced only by *TP*, so bigger clusters impact the measure more. Measures that have an explicit dependence on *TN* include the *Rand Index* [40] (a variant which is called the *Adjusted Rand Index* was employed in the network alignment study of Chap. 12 and in [83]).

3.4.2.2 Inter-Rater Reliability

Another group of external measures is connected with the statistical concept of *Inter-rater Reliability* (*Inter-rater Agreement*) [47]. The intuition behind that is connected with a situation when several raters (e.g. psychiatrists) assessed *subjects* (e.g. patients) into predefined *categories* (e.g. diseases). In such scenario a relevant question is about the agreement among raters. In other words how consistent the psychologists are in their diagnosis. Based on their assessments it is possible to build a *Contingency Table* [67] which is a generalization over a confusion matrix used in the previous subsection. In such a table the rows relate to cases (*subjects*), the columns to mental disorders (*categories*) and the elements are the number of raters that classify corresponding subject to the corresponding category. Based on such a structure one can compute the number of statistics measuring agreement among raters counting factors like agreement occurring by fortune. If there are only two raters, the *Scott's Pi* [98] or the *Cohen's Kappa* [21] could be computed. In a case of more than two raters the *Fleiss' Kappa* [38], which is based on Scott's Pi, is recommended.

Analogously, in clustering, given a clustering result and the ground truth the contingency table can be built in a way that the columns represent clusters from ground truth and the rows represent obtained clustering result groups. The elements of the matrix are the number of objects that are the same among particular groups. In such a situation the statistics from the previous paragraph can indicate how similar the obtained clustering is to ground truth. An example of how to use Inter-rater reliability indices is shown in [25].

3.4.2.3 Purity

The other group of external measures worth noting is based on the concept of *cluster purity*, that is, they focus in how homogeneous the created clusters are. A pure cluster is one that contains only points belonging to one class in the ground truth model. Cluster purity can be calculated as shown below

$$CP = \frac{\sum_{C \in \mathcal{C}} \left(\max_{k \in K} \{|X_k \cap X_C|\} \right)}{|X|}. \tag{3.14}$$

For every cluster from our result, this measure tries to find a class from the ground truth model which shares the largest number of points with that cluster. This number is then normalized by the size of input data to give the value of CP. The boundaries of this measure are [0; 1], and results closer to 1 are preferred. Its main drawback is that this measure is blind to the situation in which the clustering result has more clusters than the ground truth. It does not penalize cases when a class from the ground truth is fragmented into smaller groups in the clustering. This is because it does not take into account what fraction of particular class' points are within the considered cluster. Moreover, cluster purity concentrates only on the points that comply with the ground truth, omitting the points that do not, so one big cluster containing all the data would maximize the CP value.

The main advantage of this type of measure is that we can compare an obtained clustering result with the one that we have as a reference. If, for instance, we need a classification algorithm that relies on the results of a clustering method acting as a subroutine, it is important that the ground truth information guides the clustering algorithm, as in turn may indicate which are the best features to be used for the classification final objective. In those circumstances, external measures become really important.

There are circumstances, however, in which obtaining the ground truth is hard, expensive, or not possible for a percentage or even all the samples, e.g. when we know nothing about the ground truth clustering, and the task is to actually *discover* these groups (like in the marketing segmentation study of [26]). Additionally, there are also circumstances in which we should take extreme care when creating a ground truth model, since its form will influence the rest of the research.

3.4.3 Other Measures

There are other groups of external measures. One example are measures that originate from the *Theory of Information* [99] and are based on the concept of *Entropy*. Examples of such measures are *Normalized Mutual Information (NMI)* (used in the study of Chap. 9), *Information Gain (IG)*, or more recently *Variation of Information (VI)* [80] and *Confusion Entropy Confusion Entropy* [58].

Another group is the *Set Matching* indexes, where the purpose is to match groups from ground truth to the created groups in our model [93]. Example measures within this group are *Normalized Van Dongen (NVD)* [108] or *Pair Sets Index (PSI)* [93]. Additionally, the *Purity* and *F-Score* measures can also be viewed as *Set Matching* measures [88].

3.5 Some Partitional Clustering Methodologies in More Detail

Now that we have provided a general introduction to clustering, its key aspects and how to check quality measures we take a deeper look at some specific clustering methodologies. Specifically, most commonly known k-means algorithm is presented followed by k-Nearest Neighbour (k-NN) approaches and a variant of the k-NN. Throughout, we have also focussed on finding those business and marketing applications and domains in which these methods are heavily used and championed.

3.5.1 The k-Means Approach

It is safe to say that the k-means algorithm (first proposed in the 1960s [39, 76]) is one of the best known (and most used) partitioning clustering algorithms. The k-means method is still very much used today and one main reason for this is that it is also one of the simplest partitioning techniques available [56]. The algorithm finds a partition such that the squared error between the empirical mean of a cluster and the points in the cluster is minimized. With a k-means algorithm the user has to set a value for k a priori which will be the number of clusters the algorithm will divide the dataset in. K-means then selects k random points which are called cluster centroids (or *seeds*). Therefore, if k is set to 4 (as in the Iris dataset experiment described before), the number of centroids will be four. Then, the algorithm goes through each of the data points and it assigns each data point to the centroid that it is "closer" to (either in a graph or based on the distance matrix). Next, the algorithm calculates the cluster average for each cluster and moves the cluster centroids to the cluster average location. This action is repeated (iterated)

until there are no further changes made to the clustering (or until an alternative stopping condition is met). Figure 3.7 shows the k-means algorithm graphically in a very simple example. Here we can easily visually see that the data has four natural groupings in Fig. 3.7a and how the centroids are moved closer to the cluster average in Fig. 3.7b.

The basic k-means algorithm has a few simple steps. They are shown in the algorithm below.

Algorithm 1: The basic k-means algorithm

Input : A set of points equipped with a distance metric.
Output: A set of k clusters.

1 Select k points as the initial centroids.
2 **repeat**
3 Calculate the distance between each data point and cluster centroids.
4 Form k clusters by assigning all points to the cluster that corresponds to their closest centroid.
5 Recompute the centroids for each cluster.
6 **until** *The generated clusters stop changing.*

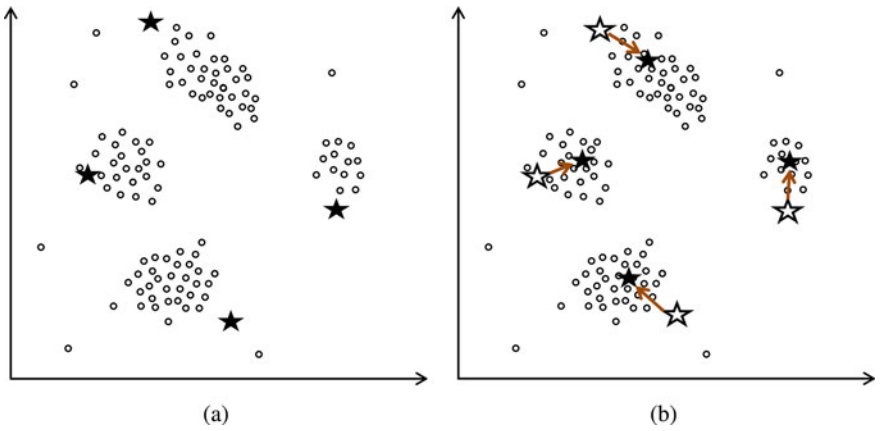

Fig. 3.7 Here we can see the k-means algorithm simplified to one simple step. In (**a**) the initial centroids/seeds (shown by stars) are randomly selected by the k-means algorithm and in (**b**) they are moved closer to the cluster average location. This step is repeated iteratively until all the squared errors between the empirical mean of a cluster and the data points in that cluster are minimized. Usually this process will be continued until no more changes are made or alternatively another stopping criteria such as the running time of the algorithm

Although k-means is a quick, efficient and simple clustering algorithm, this method is stochastic in nature and has known several disadvantages. Firstly, the a priori selection of k by the user constricts the quality of the algorithm's outcome to this user-determined parameter [56]. Further, k-means tends to have problems when the underlying clusters are of different sizes, densities or when they are non-globular shapes. The example in Fig. 3.2 already showed an instance in which k-means clustering would not be able to find the naturally occurring clusters in the two-dimensional graph space due to their non-globular shapes, density based separation and different sizes. Furthermore, k-means has problems when the data contains large outliers due to the fact that it is based on the arithmetic mean of data points [56]. Several large outliers could significantly skew the clustering outcome as they will alter the cluster average and this would have an adverse effect on the rest of the clustering outcome as k-means aims to minimize the squared error between the mean of a cluster and the points in that cluster. One more disadvantage of the k-means algorithm is that it can only be used for numerical datasets (i.e. not for categorical information).

Having recognized its disadvantages, it is still reasonable to consider k-means as one of the easiest to use clustering algorithms and the base for many other approaches of clustering. A variant of the k-means algorithm that was developed in order to deal with the disadvantage of being limited to numerical data is the "k-modes" approach [18]. As the name suggests, this algorithm takes the modes instead of averages. This means that it can be used for categorical data, or data of mixed types, and it is also a lot faster. With the increase of data mining throughout the years and the increased adoption of data analytics methods by the social sciences, categorical datasets and datasets of mixed types have become a lot more common. Variations to the k-modes have already previously been published such as Hartigan's method for k-modes [115]. Besides, k-modes, there are many other variations of the k-means algorithm including k-medoids (PAM) [91], CLARA [60, 61], CLARANS [86] among many others. It is likely to see many more variants to the k-means method to be developed and brought forward as research and data analysis capabilities continue to grow and expand.

3.5.1.1 *k*-Means in Marketing and Business Analytics

As stated, the k-means algorithm is by far one of the most widely used clustering algorithms and it is implemented in many different analysis software. Researchers from many different domains use k-means either as a sole clustering analysis, or as a comparison method to their own new algorithms. This also counts for the field of marketing and business analytics where many applications can be

found using *k*-means. Most commonly, marketing researchers likely use the *k*-means algorithm to cluster customers, whether it is in tourism applications [4, 30, 59], banking applications [77, 84, 106, 109, 114], telecommunications [71] or customer behaviours relating to their weightloss and beauty preferences [55], the focus is to cluster consumers and more accurately target market for each of them.

The paper by Kau and Lim [59] provides an interesting example of *k*-means clustering used in a tourism application. They investigated the motivations of Chinese tourists who travel to and visit Singapore. Their study shows that a technique as common and as simple as the *k*-means algorithm can provide great insights into an industry application such as investigating tourism motivation for Singapore's third largest tourist generating country. Another marketing paper using the *k*-means method is that of Kleijnen et al. [66] who investigated consumers' adoption of wireless technologies in the earlier 2000s. In their study they found three different segments of consumers when it comes to the adoption of new wireless services and products showing that manufacturers and brand managers of these products can (and should) target these consumers differently.

Besides clustering and segmenting consumers, the *k*-means algorithm is also used in other business applications. For instance, Nanda et al. [85] conducted a clustering analysis comparing *k*-means with fuzzy *c*-means and other methods analysing Indian stock market data with the purpose of improving portfolio management. The idea of their study was to select the optimal combination of stocks to create a portfolio where portfolio risk is minimized and compared to the benchmark index. Their study is a good example of the *k*-means algorithm being used in a financial application domain and providing insights for stock market traders and investors.

3.5.1.2 Variations of *k*-Means Algorithm

New advances to the *k*-means algorithm are made using marketing and business applications. For instance, Kim and Anh introduce a Genetic Algorithm (GA) optimized approach for *k*-means clustering [55] when investigating demographic and behavioural information of Korean consumers related to health, beauty and weight characteristics about themselves. Other publications also saw researchers including and integrating Self-Organizing Feature Maps (SOMS/SOFMS) together with *k*-means clustering to improve market segmentation [68, 69]. SOMS are a type of artificial neural network that provide a discretized representation of the input

space (which is called a map). Battiti and Brunato [5] provide a good introduction on SOMS in Chap. 14 of their book *The Lion Way: Machine Learning Plus Intelligent Optimization*[1] for those readers wishing to learn further about SOMS.

Other ways in which researchers have attempted to improve the k-means clustering method are by combining approaches, through, for instance, generating a hybrid approach. Wang et al. [110] propose the "K-means SVM (KMSVM) algorithm" in which Support Vector Machines (SVM) and the k-means algorithm are combined to generate better results for real-time business intelligence systems. Niknam and Amiri propose a hybrid approach combining k-means with FAPSO (fuzzy adaptive particle swarm optimization), ACO (ant colony optimization) called FAPSO-ACO-K [87].

Another example of making improvements to the clustering outcome of k-means comes from researchers combining variable selection (or weighting) techniques with the k-means algorithm. For instance, Carmone et al. also looked at the issues surrounding how to weight the variables or pre-select them for clustering [15]. They propose a new algorithm termed HINoV (the Heuristic Identification of Noisy Variables) to solve these issues and find that clustering when implementing HINoV improves the clustering outcome in terms of stability and robustness. More recently, Brusco and Cradit [13] also proposed a variable-selection heuristic for nonhierarchical (k-means) cluster analysis with the objective to include variables that truly define cluster structure and eliminate those that do not or even mask the structure. They applied their method on financial data and found that the method including a variable selection step to uncover variables that mask the structure provided an outcome with greater cluster stability than a simple clustering approach without variable selection. Steinley and Brusco [103] later built on this method proposing a variance-to-range ratio variable weighting procedure.

As this book highlights, marketing and business researchers and practitioners are increasingly adopting newer, better and more computationally complex approaches. Another example is the work by Liu et al. [74] who incorporate a multi-objective algorithm in their clustering approach combined with k-means. They propose MMSEA (a Multi-criterion Market Segmentation algorithm) and apply it on a cellphone network provider dataset (similar to the one analysed in Chap. 20 of this book) and a retail customer dataset. One of the main benefits of their method is that no multicriterion aggregation or trade-offs (of objectives) are required before the users see the full spectrum of the solution space allowing for greater flexibility and improved business decision making.

As can be seen, applications and research from the marketing and business fields have brought forward many k-means contributions. Here we have only "scratched the surface" of presenting studies that may use (some form of) the k-means algorithm; however, we have provided some ranging examples and a basic

[1] http://intelligent-optimization.org/LIONbook/.

understanding of this hugely popular approach. Further, Table 3.1 later in this chapter shows a small survey of clustering methodologies in consumer analytics and business applications for further reading. We focus on studies that are published between 2000 and 2017 in order to provide an up-to-date view of the field and so to avoid too much repetition with the survey and review works of Jain [56], Punj and Stewart [92], Steinley [102] and others.

3.5.2 Who Is Your k-Nearest Neighbour?

We have already highlighted the difference between supervised classification methods and unsupervised clustering methods. k-Nearest Neighbour (kNN) approaches can be said to be a "bridge" between these two methods. They combine parametric approaches that need a priori knowledge of the distributions underlying the data, and non-parametric approaches that assume the functional form of the discriminant surfaces partitioning the different pattern classes [22]. The kNN approach has the basic principle that an unknown entity (object in the data) is best to be assigned to the category (or cluster) to which it is closest to in a suitably defined information space (dataset) through an appropriate metric (i.e. using a distance matrix).

When k is not explicitly defined, k-NN techniques assume that $k = 1$ for the approach. That is, node a is connected to node b if b is one of node a's nearest neighbours (in the general case there may be more than one), or if node a is one of the nearest nodes of b. However, often, researchers use a k-NN approach in which the value of k will be set to suit the specific requirements. Consequently, the value of k has a significant effect on the density of the clustering result. Where a lower value of k is selected, nodes will only be connected to one or two other nodes. However, if the value of k is fixed, e.g. $k = 4$, then each data point will be connected to its $k = 4$ nearest neighbours. This is illustrated in Fig. 3.8, where on a small simulated dataset $k = 1$, $k = 3$ and $k = 6$ nearest neighbour graphs are computed and shown in the subfigures Fig. 3.8a–c, respectively.

Due to the significant effect of the value of k on the clustering outcome, there is a lot of debate about finding an "automatic" selection of k or finding the "optimum" number for the particular instance at hand. As we stated at the start of this chapter, the number of a priori parameters set by the user should ideally be kept to a minimum with unsupervised learning as this allows results to be completely data-driven.

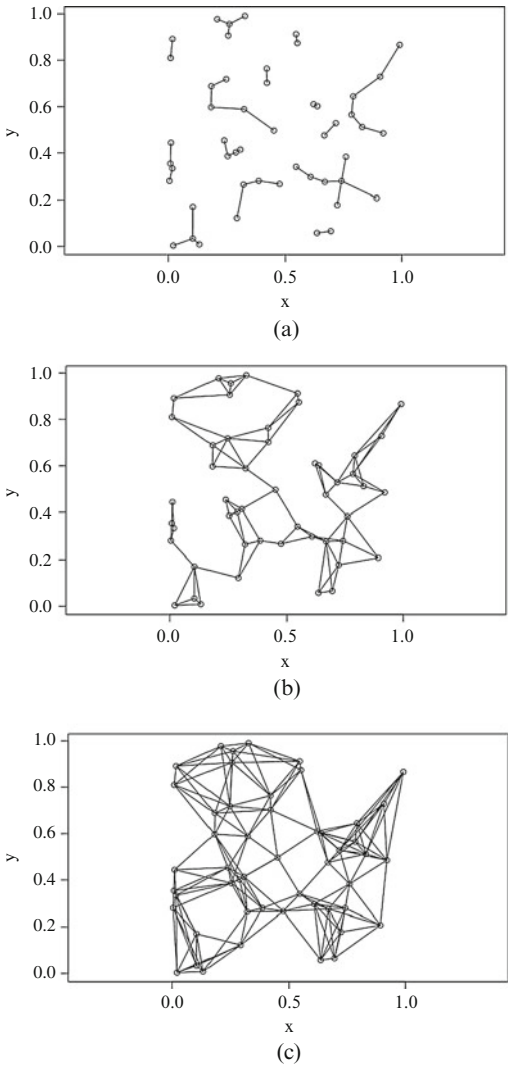

Fig. 3.8 In this figure, the effect of selecting a different value for k on the resulting nearest-neighbour graph is shown. The graph becomes a lot less or a lot more dense depending on whether nodes are connected only to their 1-nearest neighbour, their 3-nearest neighbours or to their 6-nearest neighbours. (**a**) Shows the results of running a 1-NN algorithm on a random simulated graph, in (**b**) the value of k is set to 3 and in (**c**) k is equal to 6. As can be seen in this image, the value of k has a large impact on the outcome of the kNN algorithm and thus the resulting clusters. In (**a**) we still have many partitioned small clusters, whereas when k is increased to 3, the graph is already one completely connected network and when $k = 6$, it becomes quite a densely interconnected graph

One common approach is through the use of a type of validation process (for example, cross-validation or leave-one-out). Generally, as the value of k increases, error would decrease until it stabilizes and then starts raising again as k is further increased. The rule-of-thumb is then to set k at the start of the "stable" zone in the error curve. Another rule-of-thumb approach to selecting a value for k, sometimes used in some machine learning scenarios, is to take the square root of the number of training patterns/samples (n) as this would lead to better results [31].

It is difficult to give a generic mathematically well-principled answer to which would be the best approach to select the value of k. It may be an ill-posed task because it depends not only on the problem, but also on the problem instance/input we are working with, the metric being used and other considerations. For instance, in [83] the authors calculate graphs with k ranging from 1 to 10 and then they apply their new methodology that makes use of the Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI). They report results with all these graphs and then particularize the discussion on the one that according to their proposal maximizes the product of both indexes (which turned to be $k = 5$ in their study).

3.5.2.1 Introduction to the MST-*k*NN

Recently, one of the editors of this book and his colleagues presented a new clustering methodology that uses the generation of a k-NN graph combined with a sparsification of the *Minimum Spanning Tree* (MST). They named this approach the MST-kNN [53] method. Since we will be using this approach in subsequent chapters, we introduce it here. For further details of proximity graphs that are supersets of the nearest neighbourhood graph, we refer to Chap. 4. The MST-kNN method has been tested on comprehensive studies on large-scale biological weighted networks and it has been successfully applied in various areas, see, for instance, Arefin et al. [3].

The MST-kNN algorithm has led to results that seem to be superior to known classical clustering algorithms (e.g. k-means and SOMs) in terms of homogeneity and separation [53, 54] in spite of not using an explicitly defined objective function. Due to its characteristics, it performs well even if the dataset has clusters of different mixed types (i.e. MST-kNN is not biased to "prefer" convex clusters or when the data has clusters that are embedded in subspaces of different dimensionalities). Most importantly, the MST-kNN algorithm scales very well, allowing the possibility that the methods that are based on it can be extended to the analysis of very large datasets. This opens a door for new methods in marketing that involve the analysis of datasets with millions of samples, e.g. as those arising from online behaviours, products, web pages, etc. A Graphics Processing Unit-based implementation has been made available [2].

The MST-kNN approach is basically a constructive heuristic that is not biased for the choice of a particular objective function, yet it provides a strong guarantee of optimality of a property of the final solution [53]. We explain this property after we explain the algorithm. First, the algorithm's input can be either a distance matrix between all pairs of nodes or a weighted graph. As an example, we take a dissimilarity matrix that is computed from the Spearman rank correlation matrix as the input for the algorithm (as is done in Chap. 5). Formally, if $r(a, b)$ is

the Spearman rank correlation between two objects (nodes) a and b over a set of features, then the corresponding distance matrix $D = [d(a, b)]$ with each coefficient is calculated as $d(a, b) = 1 - r(a, b)$. Given this input matrix D, the output of the MST-kNN algorithm is a forest. This means that the MST-kNN generates a partition of a set of nodes given as an input using the information of similarities/dissimilarities between each pair.

We mentioned that the algorithm returns a forest that satisfies a property. The set of nodes are the ones that are part of the input. In the forest given as output, any edge of the forest that connects two nodes does so if the edge is one of the edges of the minimum spanning tree ($MST(G)$) and, at the same time, it is also an edge present in the set of edges of the k-nearest neighbour graph ($kNN(G)$). The k-NN graph is the graph that has one node per object and that has an edge between each pair of nodes, for example, a and b, if either a is one of the k nearest neighbours of b or if b is one of the k nearest neighbours of a, or both. We note that edges of the minimum spanning tree are not bound to have this property regarding "k-neighbourness". The addition of this extra constraint has the effect of disconnecting the MST, thus creating a multi-tree forest and consequently leading to a natural partitioning of the set of nodes.

There are several variations of this scheme. In one of them, the value of k is set up to a relatively large value which is linked to the total number of nodes, and then, when the MST is fragmented in different components, a different value is selected for the different connected components using the same formula but now having for each of the connected components the number of nodes in each of them as input, thus leading to different values of k for each component. Another approach is when a value of k is fixed or when multiple values for k are trialled as done in de Vries et al. [24]. The MST-kNN will reappear in subsequent chapters in this section where outcomes of the method can be found and interpretations are explained.

An example of the MST-kNN method has been shown in Fig. 3.9. In this figure, the Wine Qualities dataset (also discussed and presented in Chaps. 16 and 26) has been clustered using the MST-kNN approach and visualized using red colours for "bad" quality wines and green colours for "good" quality wines. The quality measure is a rating given by a wine connoisseur and the features are physico-chemical properties of the wines (for instance, sugar level, acidity level, alcohol level, etc.). The figure shows the properties that we have explained of how a Minimum Spanning Tree is further subdivided leaving separate trees only connected if they are nearest neighbours. This figure shows that "bad" and "good" quality wines cannot easily be separated and wines may be similar or dissimilar based on other values. It shows that wine quality is rather heterogeneous in nature and perhaps also indicated the human subjectiveness in wine quality compared to physical properties of wine.

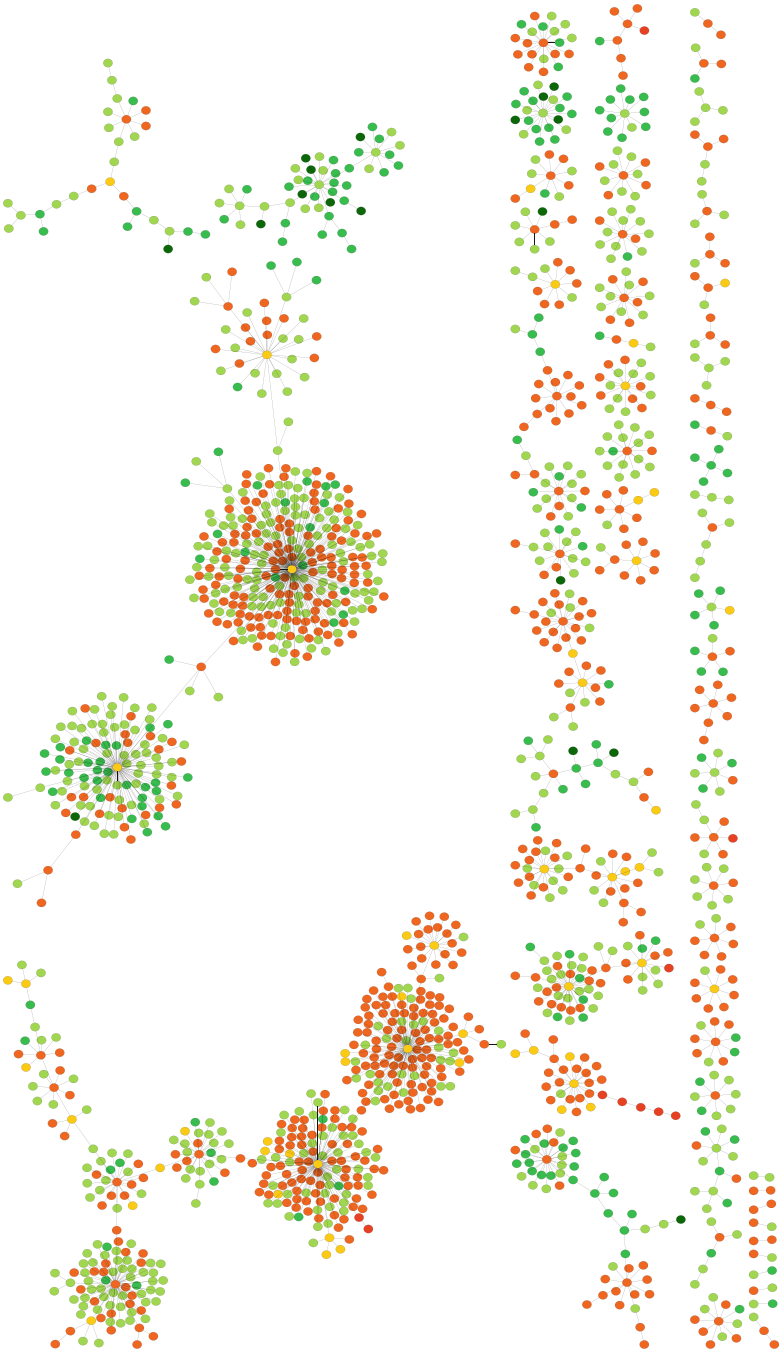

Fig. 3.9 Outcome of the MST-*k*NN algorithm on the Portuguese wine quality dataset (see Sect. 16.4 for details). Colours indicate the quality of the wine, as evaluated by specialists, showing the heterogeneity of responses between clustering outcomes and quality

3.6 All Things Fuzzy

Fuzzy clustering is also referred to as "soft clustering" as each object has a level, or percentage, membership to more than one cluster rather than a "hard division" between clusters. Soft clustering produces a membership clustering outcome [65], is commonly known as *fuzzy clustering* and is based on *fuzzy logic* [104]. With roots in control theory and artificial intelligence, fuzzy clustering is a relatively new approach in the marketing literature. Clustering methodologies used by marketers and consumer behaviour researchers encompass more "typical" clustering approaches such as k-means clustering. New studies, however, are seeing an improvement in market segmentation using "fuzzification" methods [16]. In this volume Chap. 22 provides an example of the use of a fuzzy clustering approach in a tourism application. In addition, another contribution presented in Chap. 24 uses fuzzy logic to analyse data from a tourism sustainability application.

Fuzzy logic and fuzzy sets, including fuzzy clustering, is a relatively well-known field among applied mathematicians and computer scientists with a journal dedicated to the topic since 1978 (i.e. *Fuzzy Sets and Systems*). It has been recognized in Marketing as a tool for market segmentation since the 1980s as well (see, for instance [1, 49] or [111]), but with the catalytic growth in online applications, advanced analytical approaches such as fuzzy clustering are able to make significant multi-disciplinary contributions in areas like business and marketing.

3.6.1 Fuzzy Clustering Fundamentals

Fuzzy clustering was first introduced by Bezdek in 1973 [7] and the first work leading to the "Fuzzy C-Means" (FCM) algorithm was developed and brought forward by Bezdek et al. in 1981 in a two-part publication; [8] and [9]. Bezdek used membership function matrices associated with fuzzy c-partitions of X (a set of objects). An Euclidean norm is used and fuzzy clusters are obtained. The actual FCM algorithm (and its FORTRAN coding) was published by Bezdek et al. in 1984 [10]. The FCM algorithm was based on the already popular k-means clustering methodology. In this paper, Bezdek and colleagues use a geological application to illustrate their method. Some other early seminal papers on fuzzy clustering for the interested reader can be found in [32, 96] and [112].

With the fuzzy clustering paradigm, we assume that each object/data point belongs to a cluster with a certain "degree of membership" which is represented as a number in the closed interval [0,1]. Intuitively, data points on the edge of a cluster may have a lower degree of membership than other points of the cluster. A simple example, just for illustration only, of the difference between "hard" clusters and "soft" (fuzzy) clusters is shown in Fig. 3.10. In Fig. 3.10a the clusters are completely partitioned with no overlapping data points being part of more than one cluster. In Fig. 3.10b however, we can see that some nodes (vertices) are part of two clusters.

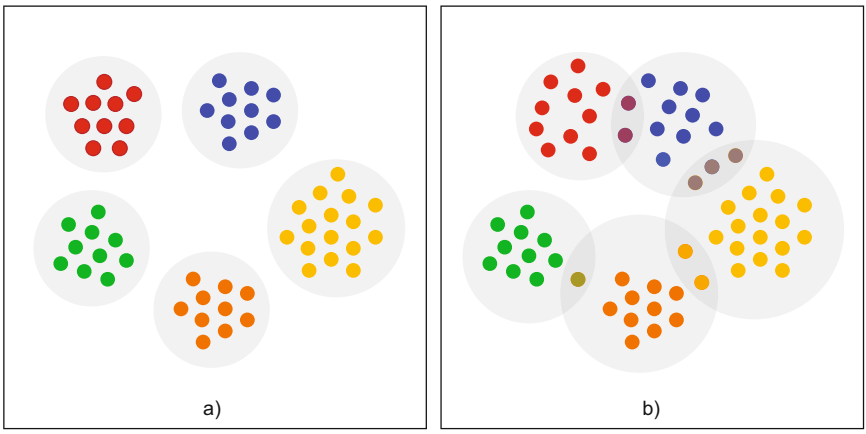

Fig. 3.10 This figure shows the difference between a hard clustering partition and a soft (fuzzy) clustering partition. On the left in figure (**a**) the clusters are clearly separated and defined by their own boundaries. On the right in figure (**b**) however, some nodes belong to two clusters at the same time. They may be 50/50 per cent split between the two clusters, or any other degree between [0,1] with a sum of 1. In this example the nodes that are part of the "fuzzy" outcome only belong to a total of two clusters; however in reality, some nodes may belong, to some degree, to as many clusters there are in the data

In this particular example there are nodes that only have a shared non-zero membership between a pair of clusters. In general, it is possible for some nodes to have a certain non-zero membership in all or in most of the clusters. Consequently, fuzzy methodologies are slightly more complex and often take longer to compute [41].

3.6.1.1 Fuzzy Clustering in Marketing and Business Analytics

As fuzzy clustering attracted more attention, it became apparent that it has many useful applications in different domains. Fuzzy clustering has successfully been used in marketing simply for market definition and segmentation. For instance, Hruschka et al. [49] found that fuzzy methods performed better in segmenting the market than "hard" clustering methodologies in terms of internal validity. They go on to state that in fact, fuzzy partitions provided more insights on segments and markets than their "hard" counterparts and the ease of interpreting outcomes from fuzzy or overlapping results was "satisfactory".

Fuzzy clusterwise regression (FCR) has been used as a benefit segmentation strategy in marketing [111]. As explained in Chap. 1, benefit segmentation separates consumers into groups who are similar to each other in terms of the benefits derived from, and the reasons for using a particular product or service. In this work, Wedel et al. [111] develop a method that estimates the models relating preference to

product dimensions within each cluster while estimating the parameters indicating the degree of membership of individuals in these clusters at the same time. Wedel et al. found their fuzzy approach to be a powerful method that uncovers a large amount of useful information. As they state, non-overlapping or "hard" partitioning clustering methodologies ignore the presence of heterogeneity that may be present in a segment and among consumers. This is one of the reasons why fuzzy or overlapping clustering methodologies is increasingly proving to be useful in marketing and consumer analytics domains. A useful review and introduction to various extensions of the fuzzy k-means algorithm are introduced and explained in Ferraro and Giordani [36].

3.6.1.2 Variations of Fuzzy Clustering

One example of a variation to the method that has been introduced is termed the "fuzzy k-modes method". As with its non-fuzzy variant, this method actually tackles some of the problems faced when working with categorical data or mixed data types. Introduced by Huang [51] and Huang and Ng [52] in 1998, it provides us with an alternative of the "standard" fuzzy C-means [33] that can cluster datasets with categorical values as well as those of mixed numerical/categorical natures. As explained by Huang, the k-modes algorithm replaces the means of clusters with modes and uses a frequency-based method to update modes in the clustering process to minimize the clustering cost function. One other advantage outlined by Huang and Ng [52] is that (in their experiments) the fuzzy k-modes algorithm performed much quicker, with less CPU time than the fuzzy k-means algorithm. Similar to the non-fuzzy k-means algorithm, we will likely see many more variants (or completely new methods) of fuzzy approaches to grouping and clustering as the business and marketing fields become increasingly intertwined with computer science and data science approaches.

3.7 Examples of Clustering Techniques in Marketing and Consumer and Business Analytics

As we have already stated, many reviews, surveys and introductions already exist to the field of clustering, segmentation and grouping. However, considering this field is such a fragmented and complex mine field, we have provided the reader with a brief survey of clustering methodologies with applications related to business, marketing and consumer analytics, focussing on those published between 2000 and 2017 for a current view of the field. This small collection of articles is shown in Table 3.1 and provides further reading material for the interested reader.

Table 3.1 A selected sample of some clustering methodologies and applications in marketing from 2000 to 2014 (ordered by year) that use some of the techniques described in this chapter

Application area & year	Clustering technique	Key characteristics
Tourism segmentation of customers to a B&B [4] (2001)	k-Means clustering	The popular k-means algorithm is used to cluster visitors to a Bed and Breakfast using multistate categorical survey data
Segmentation of customers of online music services [90] (2001)	Fuzzy c-means clustering algorithm	Users of online music services are clustered according to the fuzzy c-means and the study provided interpretable results for practitioners
Tourism segmentation using the Austrian National Guest Survey [30] (2004)	Bagged clustering	Bagged clustering is introduced as a new clustering approach for post hoc marketing segmentation drawing benefits from both partitional and hierarchical clustering methods
Segmentation of consumers regarding their adoption of wireless technologies/services [66] (2004)	k-Means clustering	Three different segments of consumers when it comes to the adoption of new wireless technologies were found using k-means providing business insights to better target market to these consumers
Clustering of Chinese tourists based on their motivations for travel in Singapore [59] (2005)	k-Means clustering	The popular k-means clustering algorithm is used on survey data of tourists using many variables related to motivation
Clustering and model-building of customers using credit card consumption data [114] (2005)	Combination of marketing RFM analysis and k-means clustering	Different values of k are trialled for k-means clustering and $k = 6$ is selected as providing the clustering results with the highest level of difference between clusters. Different consumption patterns are found between clusters of consumers
e-Banking customers in Thailand and their usage/motivations patterns [109] (2006)	k-Means clustering, SOMS and marketing RFM analysis (recency, frequency, monetary)	The resistance of Thai customers to adopt internet and e-Banking is investigated using a variety of clustering and grouping methodologies
Clustering customers of a drink company [50] (2007)	Support vector clustering (SVC)	An SVC approach is shown to outperform k-means and self-organizing feature map (SOFM) methods in providing a solid customer segmentation approach
Clustering customers to improve a recommender system based on demographic and personal information, related to weightloss and beauty needs and their related behaviours [55] (2008)	Genetic Algorithm (GA) optimized k-means clustering approach	The input for the k-means algorithm is optimized using GA approaches and compared with simple k-means and SOMS. The findings show that the GA k-means improves the segmentation of customers
Mobile phone provider customer usage behaviour clustering for targeted marketing purposes [71] (2009)	k-Means clustering	600,000 mobile customers are analysed and a real company in China implemented the results and saw their customer base increase by 64% from 2006 to 2007

(continued)

Table 3.1 (continued)

Application area & year	Clustering technique	Key characteristics
Analysing "anti-consumption" (why consumers *do not* buy something) in the case of toy libraries [89] (2010)	Hierarchical clustering followed by *k*-means	Four groups of consumers were found in terms of their "anti-consumption" behaviour regarding toys and they displayed different motivations and behaviours regarding their use of toy sharing libraries
Clustering using customer relationship information from an Iranian bank [84] (2010)	Combination of marketing RFM analysis and *k*-means clustering	A new framework for segmenting banking customers is proposed using two stages of analysis: clustering using *k*-means and use of demographic values followed by the creation of a customer profile for target marketing purposes
Customer relationship systems of airlines for Taiwanese travellers [19] (2011)	Fuzzy *k*-means algorithm	A fuzzy decision rules approach using fuzzy *k*-means in its approach is applied in a tourism application. The fuzzy *k*-means algorithm is used as a step in creating fuzzy decision rules
Clustering customers' requirements for product design [17] (2012)	Genetic algorithm (GA) for fuzzy clustering	The method integrates a fuzzy compression technique for multi-dimension reduction and a fuzzy clustering technique. Subsequently the centre points of market segments are used as "ideal points" for new product development
Analysis of Russian credit institutions (banks) [106] (2014)	*k*-means clustering and Kohonen's network	Through the clustering analysis the study confirms their expected hypothesis of an institutional misalignment existing in the Russian banking system and possible development ways of the interconnection and interaction between banks and the other economic sectors are suggested, thanks to the clustering outcomes
Wine consumption trends of Italian consumers analysis [29] (2014)	Hierarchical clustering followed by *k*-means	Three different clusters of consumers were so identified in terms of their wine consumption behaviours and attitudes (occasional and choosy consumers, basic consumers and high quality demanding purchasers)
Analysis of a large study on attitudes towards and habits of food consumption in Germany [82] (2014)	Hierarchical clustering followed by *k*-means	Their study investigating market segment stability in the German market of food consumption (over the period of 2005–2008) using clustering showed that neither the internal nor the dynamic stability of market segments should be taken for granted. This means that marketers face the challenge of designing segment-specific marketing strategies that allow changes to be made to them to remain flexible and keep up with changing consumer trends and segments

3.8 About Other Chapters That Relate to Clustering and Some Final Conclusions

It is hard to find a chapter in this book that does not have some sort of connection with clustering, either as a pre-processing or a post-processing technique to reveal structure in data. For instance, in the introductory first chapter, when we discussed the case of the US Presidential elections, feature selection techniques allow to "cluster" samples in particular groups (see Table 1.4 in Chap. 1). Analogously, techniques like those for *Frequent Itemset Mining*, described by Cafaro and Pulimento in Chap. 6, allow to "compress" information by identifying samples that share common characteristics, enabling the post-processing of large databases and the identification of interesting clusters in the outputs of itemset mining algorithms. The chapter includes techniques for parallel processing allowing the possibility of using large computing cloud systems and high-performance supercomputers.

Mathieson and Moscato, in Chap. 4, generalize the discussion on k-nearest neighbour graphs, and the MST-kNN algorithm, by considering several other "proximity graphs", which in turn can help to reveal clusters in the data. In "*Clustering Consumers and Cluster-Specific Behavioural Models*" (Chap. 5), the authors use the MST-kNN algorithm to cluster social media users. Based on the clusters found, consumer behaviours relating to the user engagement with the pages are investigated using symbolic regression analysis powered by the genetic programming techniques introduced in Chap. 1. The authors conclude that these models obtained better "*inform possible personalised marketing strategies after proper segmentation of the customers based on their online consumer behaviour, rather than simple demographic characteristics*".

Clustering methods in which membership is shared by several sets also are gaining popularity in business and customer analytics. Chapter 22 presents an application of *fuzzy clustering* in tourism analytics, presenting two different types of algorithms for fuzzy data and two empirical case studies. Chapter 21 presents a result on a hierarchical clustering algorithm, powered by metaheuristic-based optimization, on a dataset of hotel ratings. Also on the theme of tourism analytics, Chap. 23 presents a study on the *bundle design problem* which occurs when a company wants to set up offers based on sets of services and they do so based on evidence collected from consumer redemption data. More specifically, they address a problem in which a company manages multiple service providers, each responsible for an attraction, in a leisure park in Asia.

Finally, we have seen with the example presented in Fig. 3.9 that the visual presentation of the result of clustering brings some interesting challenges to computer scientists. Usually displayed in two dimensions, the output of some software currently available does not take into consideration the inter-cluster similarity. In Chap. 16, this issue is addressed presenting different alternatives for the same dataset on wine quality of Fig. 3.9 (see Fig. 16.2). The authors also include a clustering of the set of characters in the Marvel Universe (Fig. 16.1) and of an

interesting dataset on customer churn behaviour in telecoms, again showing, like in Chap. 5, the need of establishing cluster-specific models of customer behaviour for service personalization and market segmentation.

In conclusion, clustering "is more than a religion", it is actually a very necessary step for the analysis of data in business and customer analytics. Although our brief chapter cannot cover all possible techniques that exist, we hope it can motivate the reader to further study the methods presented here (as well as in other chapters). We also hope the readers will explore other methods and techniques. The diversity of points of view on "how to group things" is essential in the exploration of the structure present in large datasets.

Acknowledgements We would like to thank Ademir Cristiano Gabardo, Luke Mathieson and Shannon Fenn for their technical and proofreading help with this chapter. Łukasz P. Olech is supported by "THELXINOE: Erasmus Euro-Oceanian Smart City Network", Erasmus Mundus Action-2, Strand-2 (EMA2/S2), project funded by the European Union. Pablo Moscato acknowledges previous support from the Australian Research Council Future Fellowship FT120100060 and Australian Research Council Discovery Projects DP120102576 and DP140104183, and his academic partners in their funded project THELXINOE.

References

1. Phipps Arabie, J. Douglas Carroll, Wayne DeSarbo, and Jerry Wind. Overlapping clustering: A new method for product positioning. *Journal of Marketing Research*, 18(3):310–317, 1981.
2. Ahmed Shamsul Arefin, Carlos Riveros, Regina Berretta, and Pablo Moscato. GPU-FS-kNN: A software tool for fast and scalable kNN computation using GPUs. *PLOS ONE*, 7(8):1–13, 08 2012.
3. AhmedShamsul Arefin, Mario Inostroza-Ponta, Luke Mathieson, Regina Berretta, and Pablo Moscato. *Clustering Nodes in Large-Scale Biological Networks Using External Memory Algorithms*, volume 7017 of *Lecture Notes in Computer Science*, book section 36, pages 375–386. Springer Berlin Heidelberg, 2011.
4. George Arimond and Abdulaziz Elfessi. A clustering method for categorical data in tourism market segmentation research. *Journal of Travel Research*, 39(4):391–397, 2001.
5. Roberto Battiti and Mauro Brunato. *The Lion Way: Machine Learning Plus Intelligent Optimization*. LIONlab, University of Trento, Italy, 2014.
6. J. C. Bezdek and N. R. Pal. Cluster validation with generalized Dunn's indices. In *Proceedings 1995 Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, pages 190–193, Nov 1995.
7. James C. Bezdek. Cluster validity with fuzzy sets. *Journal of Cybernetics*, 3(3):58–73, 1973.
8. James C. Bezdek, Chris Coray, Robert Gunderson, and James Watson. Detection and characterization of cluster substructure I. linear structure: Fuzzy c-lines. *SIAM Journal on Applied Mathematics*, 40(2):339–357, 1981.
9. James C. Bezdek, Chris Coray, Robert Gunderson, and James Watson. Detection and characterization of cluster substructure II. Fuzzy c-varieties and convex combinations thereof. *SIAM Journal on Applied Mathematics*, 40(2):358-372, 1981.
10. James C. Bezdek, Robert Ehrlich, and William Full. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2):191 – 203, 1984.
11. Saprativa Bhattacharjee, Anirban Das, Ujjwal Bhattacharya, Swapan K. Parui, and Sudipta Roy. Sentiment analysis using cosine similarity measure. In *2nd IEEE International Conference on Recent Trends in Information Systems, ReTIS 2015, Kolkata, India, July 9-11, 2015*, pages 27–32. IEEE, 2015.

12. CM Bishop. *Bishop Pattern Recognition and Machine Learning*. Springer, New York, 2001.
13. Michael J. Brusco and J. Dennis Cradit. A variable-selection heuristic for k-means clustering. *Psychometrika*, 66(2):249–270, 2001.
14. Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.
15. Frank J. Carmone, Ali Kara, and Sarah Maxwell. HINoV: A new model to improve market segment definition by identifying noisy variables. *Journal of Marketing Research*, 36(4):501–509, 1999.
16. Mònica Casabayó, Núria Agell, and Germán Sánchez-Hernández. Improved market segmentation by fuzzifying crisp clusters: A case study of the energy market in Spain. *Expert Systems with Applications*, 42(3):1637 – 1643, 2015.
17. Kit Yan Chan, C.K. Kwong, and B.Q. Hu. Market segmentation and ideal point identification for new product design using fuzzy data compression and fuzzy clustering methods. *Applied Soft Computing*, 12(4):1371 – 1378, 2012.
18. Anil Chaturvedi, E. Paul Green, and Douglas J. Caroll. K-modes clustering. *Journal of Classification*, 18(1):35–55, 2001.
19. Wen-Yu Chiang. Establishment and application of fuzzy decision rules: an empirical case of the air passenger market in Taiwan. *International Journal of Tourism Research*, 13(5):447–456, 2011.
20. Prabhakar Raghavan Christopher D. Manning and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
21. Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.
22. Belur V. Dasarathy. *Handbook of Data Mining and Knowledge Discovery*, chapter Nearest-Neighbor Approaches, pages 288–298. Oxford University Press, 2002.
23. David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.
24. Natalie J de Vries, Ahmed S Arefin, and Pablo Moscato. Gauging heterogeneity in online consumer behaviour data: A proximity graph approach. In *2014 IEEE Fourth International Conference on Big Data and Cloud Computing (BDCloud)*, pages 485–492. IEEE, 2014.
25. Natalie Jane de Vries, Jamie Carlson, and Pablo Moscato. A data-driven approach to reverse engineering customer engagement models: Towards functional constructs. *PloS one*, 9(7):e102768, 2014.
26. Natalie Jane de Vries, Rodrigo Reis, and Pablo Moscato. Clustering consumers based on trust, confidence and giving behaviour: data-driven model building for charitable involvement in the Australian not-for-profit sector. *PloS one*, 10(4):e0122133, 2015.
27. Bernard Desgraupes. Clustering indices. 2013.
28. Michel Marie Deza and Elena Deza. *Encyclopedia of Distances*. Data-Centric Systems and Applications. Springer-Verlag, 3rd edition, 2014.
29. Giuseppe Di Vita, Gaetano Chinnici, and Mario D'Amico. Clustering attitudes and behaviours of Italian wine consumers. *Calitatea*, 15:54–61, 03, 2014. Copyright - Copyright Romanian Society for Quality Assurance Mar 2014; Document feature - Tables; Equations; Graphs; Last updated - 2014-03-24.
30. Sara Dolnicar and Friedrich Leisch. Segmenting markets by bagged clustering. *Australasian Marketing Journal (AMJ)*, 12(1):51 – 65, 2004.
31. Margaret H. Dunham. *Data Mining Introductory and Advanced Topics*. Pearson Education, 2nd edition, 2003.
32. J. C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Cybernetics and Systems*, 3(3):32–57, 1973.
33. Pierpaolo D'Urso and Paolo Giordani. A weighted fuzzy c-means clustering model for fuzzy data. *Computational Statistics & Data Analysis*, 50(6):1496 – 1523, 2006.
34. Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

35. Alberto Fernández and Sergio Gómez. Solving non-uniqueness in agglomerative hierarchical clustering using multidendrograms. *Journal of Classification*, 25(1):43–65, 2008.
36. Maria Brigida Ferraro and Paolo Giordani. A toolbox for fuzzy clustering using the r programming language. *Fuzzy Sets and Systems*, 279:1 – 16, 2015. Theme: Data, Audio and Image Analysis.
37. R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.
38. Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.
39. E.W. Forgy. Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. *Biometrics*, 21:768–769, 1965.
40. Edward B Fowlkes and Colin L Mallows. A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, 78(383):553–569, 1983.
41. Hichem Frigui and Raghu Krishnapuram. Clustering by competitive agglomeration. *Pattern Recognition*, 30(7):1109 – 1119, 1997.
42. Guojun Gan, Chaoqun Ma, and Jianhong Wu. *Data clustering: theory, algorithms, and applications*, volume 20 of *ASA-SIAM Series on Statistics and Applied Probability*. Siam, Philadelphia, 2007.
43. Jan Gorodkin. Comparing two k-category assignments by a k-category correlation coefficient. *Computational biology and chemistry*, 28(5):367–374, 2004.
44. M. Halkidi, M. Vazirgiannis, and Y. Batistakis. *Quality Scheme Assessment in the Clustering Process*, pages 265–276. Springer, Berlin, Heidelberg, 2000.
45. Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2):107–145, 2001.
46. Maria Halkidi and Michalis Vazirgiannis. Clustering validity assessment: Finding the optimal partitioning of a data set. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 187–194, Washington, DC, USA, 2001. IEEE Computer Society.
47. Kevin A Hallgren. Computing inter-rater reliability for observational data: an overview and tutorial. *Tutorials in quantitative methods for psychology*, 8(1):23, 2012.
48. Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems.
49. H. Hruschka. Market definition and segmentation using fuzzy clustering methods. *International Journal of Research in Marketing*, 3(2):117 – 134, 1986.
50. Jih-Jeng Huang, Gwo-Hshiung Tzeng, and Chorng-Shyong Ong. Marketing segmentation using support vector clustering. *Expert Systems with Applications*, 32(2):313 – 317, 2007.
51. Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(3):283–304, 1998.
52. Zhexue Huang and Michael K. Ng. A fuzzy k-modes algorithm for clustering categorical data. *IEEE Transactions on Fuzzy Systems*, 7(4):446–452, 1999.
53. Mario Inostroza-Ponta, Regina Berretta, Alexandre Mendes, and Pablo Moscato. *An automatic graph layout procedure to visualize correlated data*, pages 179–188. Springer, 2006.
54. Mario Inostroza-Ponta, Alexandre Mendes, Regina Berretta, and Pablo Moscato. *An integrated QAP-based approach to visualize patterns of gene expression similarity*, pages 156–167. Springer, 2007.
55. Kyoung jae Kim and Hyunchul Ahn. A recommender system using {GA} k-means clustering in an online shopping market. *Expert Systems with Applications*, 34(2):1200 – 1209, 2008.
56. Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651 – 666, 2010. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR)19th International Conference in Pattern Recognition (ICPR).
57. Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, 1999.
58. Giuseppe Jurman, Samantha Riccadonna, and Cesare Furlanello. A comparison of MCC and CEN error measures in multi-class prediction. *PLOS ONE*, 7(8):1–8, 08 2012.

59. Ah Keng Kau and Pei Shan Lim. Clustering of Chinese tourists to Singapore: an analysis of their motivations, values and satisfaction. *International Journal of Tourism Research*, 7(4-5):231–248, 2005.
60. Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, Inc., Hoboken, New Jersey, 1990.
61. Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, Inc., Hoboken, New Jersey, 2005.
62. Navneet Kaur and Craig M. Gelowitz. A tweet grouping methodology utilizing inter and intra cosine similarity. In *IEEE 28th Canadian Conference on Electrical and Computer Engineering, CCECE 2015, Halifax, NS, Canada, May 3-6, 2015*, pages 756–759. IEEE, 2015.
63. D. Kavyasrujana and B. Chakradhara Rao. *Hierarchical Clustering for Sentence Extraction Using Cosine Similarity Measure*, pages 185–191. Springer International Publishing, Cham, 2015.
64. Minho Kim and R.S. Ramakrishna. New indices for cluster validity assessment. *Pattern Recognition Letters*, 26(15):2353 – 2363, 2005.
65. Frank Klawonn, Rudolf Kruse, and Roland Winkler. Fuzzy clustering: More than just fuzzification. *Fuzzy Sets and Systems*, 281:272 – 279, 2015. Special Issue Celebrating the 50th Anniversary of Fuzzy Sets.
66. Mirella Kleijnen, Ko de Ruyter, and Martin Wetzels. Consumer adoption of wireless services: Discovering the rules, while playing the game. *Journal of Interactive Marketing*, 18(2):51 – 61, 2004.
67. Solomon Kullback. *Information theory and statistics*. Courier Corporation, 1997.
68. R.J. Kuo, Y.L. An, H.S. Wang, and W.J. Chung. Integration of self-organizing feature maps neural network and genetic k-means algorithm for market segmentation. *Expert Systems with Applications*, 30(2):313 – 324, 2006.
69. R.J. Kuo, L.M. Ho, and C.M. Hu. Integration of self-organizing feature map and k-means algorithm for market segmentation. *Computers & Operations Research*, 29(11):1475 – 1493, 2002.
70. M. Sh. Levin. Combinatorial clustering: Literature review, methods, examples. *Journal of Communications Technology and Electronics*, 60(12):1403–1428, 2015.
71. Q. Lin and Y. Wan. Mobile customer clustering based on call detail records for marketing campaigns. In *2009 International Conference on Management and Service Science*, pages 1–4, Sept 2009.
72. Bing Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Data-Centric Systems and Applications. Springer-Verlag, 2nd edition, 2008.
73. Ying Liu, Hong Li, Geng Peng, Benfu Lv, and Chong Zhang. Online purchaser segmentation and promotion strategy selection: evidence from Chinese e-commerce market. *Annals of Operations Research*, 233(1):263–279, 2013.
74. Ying Liu, Sudha Ram, Robert F. Lusch, and Michael Brusco. Multicriterion market segmentation: A new model, implementation, and evaluation. *Marketing Science*, 29(5):880–894, 2010.
75. Benjamin Lucas, Ahmed Shamsul Arefin, Natalie Jane de Vries, Regina Berretta, Jamie Carlson, and Pablo Moscato. Engagement in motion: Exploring short term dynamics in page-level social media metrics. In *2014 IEEE Fourth International Conference on Big Data and Cloud Computing, BDCloud 2014, Sydney, Australia, December 3-5, 2014*, pages 334–341, 2014.
76. James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.
77. Katariina Mäenpää. Clustering the consumers on the basis of their perceptions of the internet banking services. *Internet Research*, 16(3):304–322, 2006.

78. Pritha Mahata, Wagner Costa, Carlos Cotta, and Pablo Moscato. Hierarchical clustering, languages and cancer. In Franz Rothlauf, Jürgen Branke, Stefano Cagnoni, Ernesto Costa, Carlos Cotta, Rolf Drechsler, Evelyne Lutton, Penousal Machado, Jason H. Moore, Juan Romero, George D. Smith, Giovanni Squillero, and Hideyuki Takagi, editors, *Applications of Evolutionary Computing, EvoWorkshops 2006: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoINTERACTION, EvoMUSART, and EvoSTOC, Budapest, Hungary, April 10-12, 2006, Proceedings*, volume 3907 of *Lecture Notes in Computer Science*, pages 67–78. Springer, 2006.
79. Brian W Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.
80. Marina Meilă. *Comparing Clusterings by the Variation of Information*, pages 173–187. Springer, Berlin, Heidelberg, 2003.
81. Volodymyr Melnykov and Ranjan Maitra. Finite mixture models and model-based clustering. *Statist. Surv.*, 4:80–116, 2010.
82. Henriette Müller and Ulrich Hamm. Stability of market segmentation with cluster analysis – a methodological approach. *Food Quality and Preference*, 34:70 – 78, 2014.
83. Leila M Naeni, Hugh Craig, Regina Berretta, and Pablo Moscato. A novel clustering methodology based on modularity optimisation for detecting authorship affinities in Shakespearean era plays. *PLoS One*, 11(8):e0157988, 2016.
84. Morteza Namvar and Mohammad R. Gholamian. A two phase clustering method for intelligent customer segmentation. In *Proceedings of the International Conference on Intelligent Systems, Modelling and Simulation*, pages 61–68, Liverpool, UK, 2010. IEEE.
85. S.R. Nanda, B. Mahanty, and M.K. Tiwari. Clustering Indian stock market data for portfolio management. *Expert Systems with Applications*, 37(12):8793 – 8798, 2010.
86. R. T. Ng and Jiawei Han. CLARANS: a method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, 14(5):1003–1016, Sep 2002.
87. Taher Niknam and Babak Amiri. An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. *Applied Soft Computing*, 10(1):183 – 197, 2010.
88. Łukasz P. Olech and Mariusz Paradowski. *Hierarchical Gaussian Mixture Model with Objects Attached to Terminal and Non-terminal Dendrogram Nodes*, pages 191–201. Springer International Publishing, Cham, 2016.
89. Lucie K. Ozanne and Paul W. Ballantine. Sharing as a form of anti-consumption? An examination of toy library users. *Journal of Consumer Behaviour*, 9(6):485–498, 2010.
90. Muammer Ozer. User segmentation of online music services using fuzzy clustering. *Omega*, 29(2):193 – 206, 2001.
91. Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36(2, Part 2):3336 – 3341, 2009.
92. Girish Punj and David W. Stewart. Cluster analysis in marketing research: Review and suggestions for application. *Journal of Marketing Research*, 20(2):pp. 134–148, 1983.
93. M. Rezaei and P. Fränti. Set matching measures for external cluster validity. *IEEE Transactions on Knowledge and Data Engineering*, 28(8):2173–2186, Aug 2016.
94. Romeo Rizzi, Pritha Mahata, Luke Mathieson, and Pablo Moscato. Hierarchical clustering using the arithmetic-harmonic cut: Complexity and experiments. *PLOS ONE*, 5(12):1–8, 12 2010.
95. Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
96. Enrique H. Ruspini. A new approach to clustering. *Information and Control*, 15(1):22 – 32, 1969.
97. Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications. *Data Mining and Knowledge Discovery*, 2(2):169–194, 1998.
98. William A Scott. Reliability of content analysis: The case of nominal scale coding. *Public opinion quarterly*, pages 321–325, 1955.

99. Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.
100. Padhraic Smyth. *Handbook of Data Mining and Knowledge Discovery*, chapter 16.5 Clustering, pages 386–388. Oxford University Press, 2002.
101. Michał Spytkowski, Łukasz P. Olech, and Halina Kwaśnicka. *Hierarchy of Groups Evaluation Using Different F-Score Variants*, pages 654–664. Springer, Berlin, Heidelberg, 2016.
102. Douglas. Steinley. K-means clustering: A half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, 59(1):1–34, 2006.
103. Douglas Steinley and Michael J. Brusco. A new variable weighting and selection procedure for k-means cluster analysis. *Multivariate Behavioral Research*, 43(1):77–108, 2008. PMID: 26788973.
104. Michio Sugeno and Takahiro Yasukawa. A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on fuzzy systems*, 1(1):7–31, 1993.
105. Michael Nche Tuma, Reinhold Decker, and Sören Scholz. A survey of the challenges and pitfalls of cluster analysis application in market segmentation. *International Journal of Market Research*, 53(3):391–414, 2011.
106. VI Vagizova, KM Lurie, and Ihor Bogdanovych Ivasiv. Clustering of Russian banks: business models of interaction of the banking sector and the real economy. *Problems and perspectives in management*, 12(1):83–93, 2014.
107. C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979.
108. Silke Wagner and Dorothea Wagner. *Comparing clusterings: an overview*. Universität Karlsruhe, Fakultät für Informatik Karlsruhe, 2007.
109. Anongnart Srivihok Waminee Niyagas and Sukumal Kitisin. Clustering e-banking customer using data mining and marketing segmentation. *Transactions on Computer and Information Technology (ECTI-CIT)*, 2(1), 2006.
110. Jiaqi Wang, Xindong Wu, and Chengqi Zhang. Support vector machines based on k-means clustering for real-time business intelligence systems. *International Journal of Business Intelligence and Data Mining*, 1(1):54–64, 2005.
111. Michel Wedel and Jan-Benedict E.M. Steenkamp. A fuzzy clusterwise regression approach to benefit segmentation. *International Journal of Research in Marketing*, 6(4):241 – 258, 1989.
112. William G. Wee and K. S. Fu. A formulation of fuzzy automata and its application as a model of learning systems. *IEEE Transactions on Systems Science and Cybernetics*, 5(3):215 – 223, 1969.
113. Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Elsevier, 2nd edition, 2005.
114. Jing Wu and Zheng Lin. Research on customer segmentation model by clustering. In *Proceedings of the 7th International Conference on Electronic Commerce*, ICEC '05, pages 316–318, New York, NY, USA, 2005. ACM.
115. Zhengrong Xiang and Md Zahidul Islam. Hartigan's method for k-modes clustering and its advantages. In *Proceedings of the Twelfth Australasian Data Mining Conference (AusDM 2014), Brisbane, Australia*, pages 25–30. Australian Computer Society Inc., 2014.
116. Rui Xu and Donald C. Wunsch II. *Clustering*. IEEE Press Series on Computational Intelligence. John Wiley & Sons, Inc., Hoboken, New Jersey, 2009.