# Towards Theoretical Foundations of Clustering

by

Margareta Ackerman

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

# Abstract

Clustering is a central unsupervised learning task with a wide variety of applications. Unlike in supervised learning, different clustering algorithms may yield dramatically different outputs for the same input sets. As such, the choice of algorithm is crucial. When selecting a clustering algorithm, users tend to focus on cost-related considerations, such as running times, software purchasing costs, etc. Yet differences concerning the output of the algorithms are a more primal consideration. We propose an approach for selecting clustering algorithms based on differences in their input-output behaviour. This approach relies on identifying significant properties of clustering algorithms and classifying algorithms based on the properties that they satisfy.

We begin with Kleinberg's impossibility result, which relies on concise abstract properties that are well-suited for our approach. Kleinberg showed that three specific properties cannot be satisfied by the same algorithm. We illustrate that the impossibility result is a consequence of the formalism used, proving that these properties can be formulated without leading to inconsistency in the context of clustering quality measures or algorithms whose input requires the number of clusters.

Combining Kleinberg's properties with newly proposed ones, we provide an extensive property-base classification of common clustering paradigms. We use some of these properties to provide a novel characterization of the class of linkage-based algorithms. That is, we distil a small set of properties that uniquely identify this family of algorithms.

Lastly, we investigate how the output of algorithms is affected by the addition of small, potentially adversarial, sets of points. We prove that given clusterable input, the output of $k$-means is robust to the addition of a small number of data points. On the other hand, clusterings produced by many well-known methods, including linkage-based techniques, can be changed radically by adding a small number of elements.

# Acknowledgements

It is with great joy that I put together this thesis. It has been a most extraordinary experience of discovery and personal growth. The years during which this research was conducted were truly wonderful, in large part thanks to many great people. First and foremost, I would like to thank an extraordinary supervisor and exceptional researcher, Professor Shai Ben-David, for all of his guidance and support. But most importantly, Shai taught me how to do research, and for that I am forever grateful.

I was also very lucky to have an exceptional internal committee. On multiple occasions, Dan Brown went out of his way to help me. His guidance and support made a tremendous difference. I am also very grateful to have had a chance to work with Alejandro Lopez Ortiz. Working with Alex has been a true inspiration, and I greatly appreciate his help and guidance.

I would like to thank David Loker, for many fruitful discussions and his endless support. I would also like to thank Simina Branzei, a colleague and also my best friend, for our many fruitful discussions, for always being there for me, and for her ongoing encouragement.

I would like to thank our graduate secretary, Margaret Towell, who has gone out of her way to help me on numerous occasions.

A thanks goes out to my parents, Anna and Efim, for their support in every endeavour that I have ever undertaken. Their belief in me is at the core of everything I do.

Finally, I would like to thank my son, Alexander, born only four months after the beginning my PhD. His birth inspired me to become better in many ways, and in particular, made me a better researcher.

Studying at Waterloo was one of the best decisions I ever made. Thanks everyone who has made it so.

# CONTENTS

# CHAPTER 1

# INTRODUCTION

Clustering is a fundamental and immensely useful tool for exploratory data analysis. It is used in a wide range of applications. For instance, clustering is used in facility allocation to determine the placement of new services. In marketing, it is applied to identify groups of customers to which new products can be targeted. In phylogeny, a field whose aim is to reconstruct the tree of life, clustering techniques are used to construct phylogenetic trees.

The popularity of clustering is hardly surprising, as its goal is natural: to identity groups of similar items within data. Yet while the intuitive goal of clustering is simple, formalizing this task is much more challenging.

One of the main difficulties to formalizing clustering is that, unlike supervised learning, clustering is inherently ambiguous. Consider for example the data set displayed in Figure 1.1, in which there are two reasonable clusterings one with two and the other with three clusters. Ambiguity often occurs even when the number of clusters is fixed. Figure 1.2 illustrates a data set with two reasonable partitions into two clusters. Two radically different, yet reasonable, clusterings into three partitions are shown in Figure 1.3.

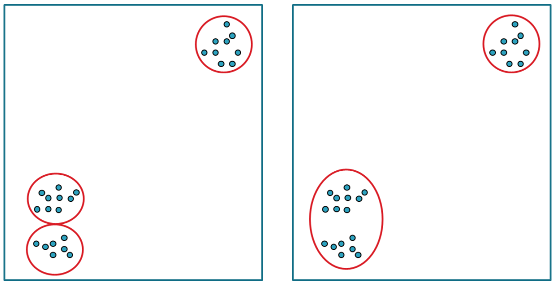


Figure 1.1: Two reasonable clusterings, with a different number of clusters, of the same data set.

The ambiguous nature of clustering led to a wide range of mathematical formalizations that define clustering. Perhaps the most common method for formalizing clustering is through clustering quality measures, which express the goal of clustering using concise mathematical formulae.

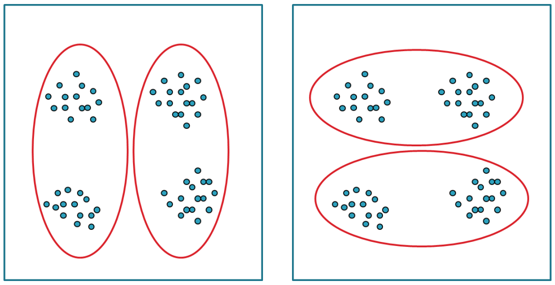


Figure 1.2: Two reasonable 2-clusterings of the same data set.

Clustering quality measures map pairs of the form (*dataset*, *clustering*) to real numbers. These measures define the goal of clustering by providing methods for comparing clusterings, suggesting that clusterings with better scores correspond to better clusterings. Some clustering quality measures are used to drive clustering algorithms, in that context quality measures are often referred to as "objective functions." Some common objective functions formalize the idea that a cluster should have strong internal cohesion. For example, one of the most popular clustering objective functions is $k$-means, which calculates the squared sum of distances from elements to the centers of mass of their clusters. That is, given a clustering (or, partition) $C = \{C_1, \cdots, C_k\}$, the $k$-means cost of $C$ is

$$\sum_{i=1}^{k} \sum_{x \in C_i} d(x, c_i)^2,$$

where $c_i$ is the center of mass of cluster $C_i$ and $d(x, c_i)$ denotes the distance between $x$ and $c_i$.

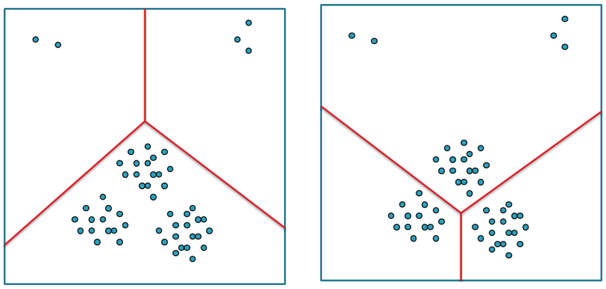


Figure 1.3: Two radically different 3-clusterings of the same data set.

Other common clustering objective functions, including ratio cut and normalized cut [45], focus on cluster separation instead of cluster cohesion. Formal definitions of these objectives are given in the preliminary section (Chapter 2). Unfortunately, for most popular clustering objectives, finding the clustering with optimal value of the objective function is $NP$-hard ([36], [38]). Therefore, in practice, heuristics are used. This further increases the set of available clustering tools.

Not only are there many clustering algorithms, but these algorithms also tend to have very different input-output behaviour. Unlike algorithms for supervised learning, clustering algorithms often output drastically different solutions over the same data. One such example is illustrated in Figure 1.4, where the clustering on the left hand side is obtained by the single-linkage algorithm

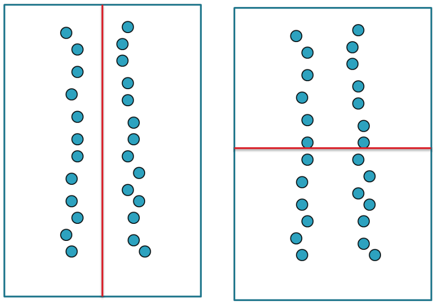


Figure 1.4: Two different 2-clusterings of the same data set. The clustering on the left hand side is found by single-linkage, while the clustering displayed on the right is obtained by $k$-means and related heuristics.

while center-based methods such as $k$-means obtain the partition on the right. Another example is found in Figure 1.3. On the left hand side is a partition with large separation between clusters, which is obtained by common linkage-based techniques (eg. average-linkage) as well as objective functions such as min-diameter and $k$-center. On the other hand, the $k$-means objective outputs the clustering on the right hand side[1].

The diversity of clustering techniques presents a real challenge for a user who needs to choose a technique for a specific application. Currently, such decisions are often made in a very *ad hoc*, if not completely random, manner. Given the crucial effect of the choice of a clustering algorithm on the resulting clustering, this state of affairs is truly regrettable. Cost related factors are often considered, such as running time and software purchasing costs. Yet these considerations do not go to the heart of the difference between these algorithms. To make an informed choice, it is first necessary to understand fundamental differences in the input-output behaviour of different clustering paradigms.

We propose an approach for providing guidance to clustering users centred on differences in the input-output behaviour of algorithms. Our approach is based on identifying significant properties of clustering functions that, on one hand distinguish between different clustering paradigms, and on the other hand are intended to be relevant to the domain knowledge that a user might have access to. Based on domain expertise, users could then choose which traits they want an algorithm to satisfy, and select an algorithm accordingly. The emphasis of the current thesis is to develop this approach. We identify properties that highlight fundamental differences in the input-output behaviour of clustering paradigms, and prove which algorithms satisfy these properties. This leads to improved understanding of clustering algorithms, which in turn helps make a more informed choice when selecting an algorithm for a specific application.

Before elaborating on our contributions, we discuss previous work on theoretical foundations of clustering.

[1] The data set illustrated in Figure 1.3 motivates our discussion in Chapter 6 where we study the underlying cause leading to such differences in the output of common clustering methods.

## 1.1 Previous Work

Mostly in recent years, a few different approaches towards developing a general theory of clustering have been investigated. Ben-David [14] considers a sample-based framework for clustering, where the input is an independent and identically distributed sample from a distribution, and the goal is to provide a partition of the full domain set. In subsequent work, Luxburg and Ben-David [46] propose other avenues for investigation towards a statistical theory of clustering.

In another direction of research, by Balcan, Blum, and colleagues ([12], [11], and [13]), the emphasis is on properties of clusterings that make clustering computationally easier. In particular, it is assumed that there is some correct, unique target clustering. If it is known that the target satisfies certain conditions, then this prior could be used to help uncover the target clustering. They propose examples of such priors, and show that there is an efficient algorithm that finds the right clustering for each prior.

The direction of research towards a general theory of clustering that is most relevant to our work is concerned with distilling natural, abstract properties of clustering. This line of research has been used to study different aspects of clustering. Meila [39] studies properties of criteria for comparing clusterings, functions that map pairs of clusterings to real numbers, and identifies properties that are sufficient to uniquely identify several such criteria. Puzicha et al. [43] explore properties of clustering objective functions. They propose a few natural axioms of clustering objective functions, and then focus on objective functions that arise by requiring functions to decompose into additive form.

Most work on abstract properties of clustering is concerned with clustering functions. Wright [47] proposes axioms of clustering functions in a weighted setting, where every domain element is assigned a positive real weight, and its weight may be distributed among multiple clusters. There have also been several property-based characterizations of the single-linkage algorithm. Jardine and Sibson [33] formulate a collection of properties that define single linkage within the class of hierarchical clustering functions. More recently, Ben-David and Bosagh Zadeh [48] characterize single linkage in the partitional setting (using the $k$-stopping criterion). In addition, Carlsson and Memoli [17] provide a characterization of the single-linkage algorithm in the hierarchical clustering setting.

One of the most influential papers in this line of work is Kleinberg's [35] impossibility result. Kleinberg proposes three axioms of clustering functions, each sounding natural, and proves that no clustering function can simultaneously satisfy these properties. This result has been interpreted as stating the impossibility of defining what clustering is, or even of developing a general theory of clustering.

We have recently found out that an approach for selecting clustering algorithms that is similar to ours has been proposed by Fisher and Van Ness [27]. However, the set of properties they discuss is very different from ours. Many of their properties require the assumption that the data lie in Euclidean space (and sometimes even restricted to the two-dimensional plane), while we focus on properties that make no assumptions on the underlying space. In a follow-up to that paper, Chen and Van Ness [19, 18, 20] investigated properties of linkage-functions. As such, these results apply to selecting clustering algorithms only when users know that they are interested in a linkage-based technique and also have some prior knowledge about the desired linkage function. In contrast, we rely exclusively on properties of the input-output behaviour of algorithms. This enables the use of our properties for comparing algorithms across different clustering paradigms. We emphasize that none of our results have appeared before our publications.

### 1.1.1 Our Contributions

We begin this thesis with a rebuttal to Kleinberg's impossibility result. We show that the impossibility result is, to a large extent, due to the specific formalism used by Kleinberg, rather than being an inherent feature of clustering. While Kleinberg's axioms are inconsistent in the setting of clustering functions, we show that consistency is retained in a closely related setting of clustering-quality measures [4]. In Chapter 3, we translate Kleinberg's axioms into the latter setting, and show that several clustering-quality measures satisfy these properties.

In the remainder of this thesis, we work towards a general theory of clustering by studying the input-output behaviour of clustering algorithms. While clustering axioms would identify what is common to all clustering functions, concisely formulated properties can be used to distinguish between different clustering paradigms. Identifying properties that bring to light fundamental differences between clustering algorithms and classifying them accordingly provides a disciplined approach for the selection of clustering techniques for specific applications.

In Chapter 4, we distil a set of abstract properties that distinguish between linkage-based clustering and all other clustering paradigms [5]. Linkage-based clustering is a family of clustering methods that include some of the most commonly-used and widely-studied clustering algorithms. We provide a simple set of properties that, on one hand is satisfied by all the algorithms in that family, while on the other hand, no algorithm outside that family satisfies all of the properties in that set. This characterization applies in the partitional setting by using the $k$-stopping criterion, and allows for a comparison of linkage-based algorithms to other common partitional methods.

The ultimate vision is that there would be a sufficiently rich set of properties that would provide a detailed, property-based, taxonomy of clustering methods. This taxonomy could then be used to guide algorithm selection for a wide variety of clustering applications. In Chapter 5, we take a step towards this goal by using natural properties to examine some popular clustering approaches, and present a property-based classification of these methods [6].

At the end of Chapter 5, we study relationships between the properties, independent of any particular algorithm. We illustrate some positive relationships between some of the properties that we study. Finally, we strengthen Kleinberg's impossibility result [35] by using a relaxation of one of the properties that he proposed. Our proof is also notably simpler than the proof of the original impossibility result.

In Chapter 6, we study differences in the input-output behaviour of clustering algorithms when a small number of points is added. We show that the output of some algorithms is highly sensitive to the addition of small sets. In such cases, we call such sets *oligarchies*. An oligarchy typically refers to a small group of individuals that have a lot of influence on a large group of people. Similarly, we use the term "oligarchy" to refer to a small number of points that greatly effect how the entire data set is clustered.

On the other hand, there are clustering methods that are robust to the addition of small sets, even when those are selected in an adversarial manner. As discussed in Chapter 6, robustness to small sets is an important consideration when selecting an algorithm, and the desired behaviour depends on the application.

While most of this thesis is concerned with partitional clustering, in Chapter 7, we turn to the hierarchical clustering setting. We provide a generalization of our characterization of linkage-based algorithms to the hierarchical setting [2]. While the characterization presented in Chapter 4 shows how linkage-based algorithms with the $k$-stopping criterion differ from other partitional clustering methods, this result shows how linkage-based algorithms are distinguished from other hierarchical techniques. We also show that linkage-based algorithms are distinct from a class of bisecting algorithms in the following strong sense: no linkage-based algorithm can be used to simulate the input-output behaviour of any algorithm in this class. We conclude with a discussion of our results in Chapter 8.

# CHAPTER 2

# PRELIMINARIES

In this chapter, we introduce our notation, definitions, and common clustering algorithms that will be referred to throughout the thesis.

## 2.1 Definitions and Notation

Clustering is a very wide and heterogenous domain. We choose to focus on a basic sub-domain where the input to the clustering function is a finite set of points endowed with a between-points distance (or similarity) function, and the output is a partition of that input. This sub-domain is rich enough to capture many of the fundamental issues of clustering, while keeping the underlying structure as succinct as possible.

**Definition 1** (Distance function)**.** *A* distance function *is a symmetric function*

$$d : X \times X \to R^{\geq 0},$$

*such that $d(x, x) = 0$ for all $x \in X$.*

The objects that we consider are pairs $(X, d)$, where $X$ is some finite domain set and $d$ is a distance function over $X$. These are the inputs for clustering functions. The *size* of a set $X$, denoted $|X|$, refers to the number of elements in $X$.

Given a distance function $d$ over $X$ and a positive real $c$, $c \cdot d$ is defined by setting, for every pair $x, y \in X$, $(c \cdot d)(x, y) = c \cdot d(x, y)$.

Two distance functions $d$ over $X$ and $d'$ over $X'$ *agree* on a domain set $Y$ if $Y \subseteq X$, $Y \subseteq X'$, and $d(x, y) = d'(x, y)$ for all $x, y \in Y$.

At times we consider a domain subset with the distance induced from the full domain set. We let $(X', d') \subseteq (X, d)$ denote $X' \subseteq X$ and $d' = d|X'$, which is defined by restricting the distance function $d$ to $X' \times X'$.

We say that a distance function $d$ over $X$ *extends* a distance function $d'$ over $X'$ if $X' \subseteq X$ and for all $x, y \in X'$, $d(x, y) = d'(x, y)$.

A *$k$-clustering* $C = \{C_1, C_2, \ldots, C_k\}$ of data set $X$ is a partition of $X$ into $k$ disjoint subsets of $X$ (so, $\bigcup_i C_i = X$). A *clustering* of $X$ is a $k$-clustering of $X$ for some $1 \leq k \leq |X|$. A clustering is *trivial* if either all data belongs to the same cluster, or each element is in a distinct cluster.

For a clustering $C$, let $|C|$ denote the number of clusters in $C$. For $x, y \in X$ and clustering $C$ of $X$, we write $x \sim_C y$ if $x$ and $y$ belong to the same cluster in $C$ and $x \not\sim_C y$, otherwise.

**Definition 2** (Clustering function)**.** *A* clustering function *is a function that takes a pairs* $(X, d)$*, and outputs a clustering of* $X$*.*

We also consider a clustering function that takes the number of clusters as a parameter. This parameter is often denoted "$k$", leading to the name "$k$-clustering function".

**Definition 3** ($k$-clustering function)**.** *A* $k$-clustering function *is a function that takes a pair* $(X, d)$ *and an integer* $1 \leq k \leq |X|$*, and outputs a $k$-clustering of* $X$*.*

We say that $(X, d)$ and $(X', d')$ are *isomorphic domains*, denoting it by $(X, d) \sim (X', d')$, if there exists a bijection $\phi : X \to X'$ so that $d(x, y) = d'(\phi(x), \phi(y))$ for all $x, y \in X$.

We say that two clusterings $C$ of some domain $(X, d)$ and $C'$ of some domain $(X', d')$ are *isomorphic clusterings*, denoted $(C, d) \cong_C (C', d')$, if there exists a domain isomorphism $\phi : X \to X'$ so that $x \sim_C y$ if and only if $\phi(x) \sim_{C'} \phi(y)$.

## 2.2 Common Clustering Methods

We define some common clustering functions referred to throughout the thesis.

### 2.2.1 Linkage-Based Clustering

Linkage-based clustering algorithms are iterative algorithms that begin by placing each point in a distinct cluster, and then repeatedly merge the closest clusters. When the *$k$-stopping criterion* is applied, the algorithm terminates when a specified number of clusters is formed.

The distance between clusters is determined by a linkage function. The linkage functions used by the most common linkage-based algorithms are as follows.

- *Single linkage:* $\min_{a \in A, b \in B} d(a, b)$.
- *Average linkage:* $\frac{\sum_{a \in A, b \in B} d(a,b)}{|A| \cdot |B|}$
- *Complete linkage:* $\max_{a \in A, b \in B} d(a, b)$.

We elaborate on linkage-based algorithm in Chapter 4. Linkage-based algorithms are also frequently applied in the hierarchical clustering setting, where linkage function are used to construct dendrograms, which simultaneously represent multiple clusterings. This is discussed in detail in Chapter 7.

### 2.2.2 Objective-Based Clustering

Many clustering algorithms aim to find clusterings with low loss with respect to a specific objective function. An example of such an objective function is Min-Sum, the sum of within-cluster distances,

$$\text{Min-sum}(C, (X, d)) = \sum_{x \sim_C y} d(x, y).$$

Every objective function $\mathcal{O}$ has a corresponding clustering function $F$ that outputs a clustering that optimizes $\mathcal{O}$, namely $F(X, d, k) = \text{argmin}_{C \text{ a } k\text{-clustering of } X}\ \mathcal{O}(C, (X, d))$ where argmax is used instead of argmin if higher values of $\mathcal{O}$ represent better clusterings.

Computing the optimal solution is often hard. But there are computationally efficient algorithms that aim to find solutions will low loss, even if they may not be optima. We discuss an example of such an algorithm at the end of this section.

We now present similarity-based clustering objective functions, centroid objective functions, and $k$-means.

#### Similarity-based

Similarity-based objective functions, typically estimated by spectral relaxations, focus on between-cluster edges. They are defined using similarities instead of distances. A *similarity function*, $s(x, y)$, is defined like a distance function, but the implied meaning of larger values represents greater similarity, instead of greater distance.

**Definition 4** (Similarity function)**.** *A* similarity function *is a symmetric function*

$$s : X \times X \to R^{\geq 0},$$

*for all $x \in X$.*

Given a cluster $C_i \subseteq X$, let $\bar{C_i} = X \backslash C_i$. Given $C_i, C_j \subseteq X$, we define $cut(C_i, C_j) = \sum_{x \in C_i, y \in C_j} s(x, y)$. Let the volume of a cluster $C_i$ be the sum of within-cluster similarities, $vol(C_i) = \sum_{x,y \in C_i} s(x, y)$. We consider two similarity-based objective functions. The first is Ratio Cut,

$$RatioCut(C, (X, s)) = \sum_{C_i \in C} \frac{cut(C_i, \bar{C_i})}{|C_i|}.$$

The next objective function is called Normalized Cut, as it normalizes by cluster volume.

$$NormalizedCut(C, (X, s)) = \sum_{C_i \in C} \frac{cut(C_i, \bar{C_i})}{vol(C_i)}.$$

Larger values of both ratio cut and normalized cut aim to represent better clusterings.

### Centroid

Following Kleinberg's [35] definition, $(k, g)$-*centroid* clustering functions find a set of $k$ "centroids" $\{c_1, \ldots, c_k\} \subseteq X$ so that $\sum_{x \in X} \min_i g(d(x, c_i))$ is minimized, where $g : R^+ \to R^+$ is a continuous, non-decreasing, and unbounded function. The *k-medoids* objective function is obtained by setting $g$ to the identity.

### $k$-means

The $k$-means objective is to find a set of $k$ elements $\{c_1, c_2, \ldots, c_k\}$ in the *underlying space*, so that $\sum_{x \in X} \min_i d(x, c_i)^2$ is minimized. A common variation on the $k$-means objective function is $k$-medians, which is obtained by omitting the square on $d(x, c_i)$.

This formalization of $k$-means assumes that the data lies in a normed vector space. This method is typically applied in Euclidean space, where the $k$-means objective is equivalent to $k\text{-means}(C, (X, d)) = \sum_{C_i \in C} \frac{1}{|C_i|} \sum_{x,y \in C_i} d(x, y)^2$. (See [42] for details).

The most common heuristic in Euclidean space for finding clusterings with low $k$-means loss is Lloyd's method.

**Definition 5** (Lloyd's method)**.** *Given a data set $(X, d)$, and a set $S$ of $k$ points in $R^n$, the Lloyd's method performs the following steps until two consecutive iterations return the same clustering.*

1. *Assign each point in $X$ to its closest element of $S$. That is, find the clustering $C$ of $X$ so that $x \sim_C y$ if and only if $argmin_{c \in S} \|c - x\| = argmin_{c \in S} \|c - y\|$.*
2. *Compute the centers of mass of the clusters. Set $S = \{c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x \mid C_i \in C\}$.*

A common initialization for Lloyd's method is to select $k$ random points from the input data set ([28]). We call this algorithm Randomized Lloyd. It is also commonly referred to as "the $k$-means algorithm." In order to find a solution with low $k$-means loss, it is common practice to run Randomized Lloyd multiple times and then select the minimal cost clustering.

Another well-known initialization method for Lloyd's method is furthest-centroid initialization [34]. Using this method, given a set $X$, the initial points $S = \{c_1, \ldots, c_k\}$ are chosen as follows: $c_1$ is the point with maximum norm (instead, an arbitrary point can be chosen). Then, for all $i$ between 2 and $k$, $c_i$ is set to be the point in $X$ that maximizes the distance from the other points that were already chosen. That is, $c_i = \text{argmax}_{x \in X} \min_{j \in [i-1]} d(x, c_j)$.

Both of these methods of initialization have significant shortcomings. In particular, such common initialization techniques can fail dramatically even when the data is very nice, such as i.i.d. samples of very well-separated spherical Gaussian in $R^d$ [22]. Additionally, random center initialization is unstable, in the sense that the final solution is sensitive to center initialization [16]. The good news is that there is an initialization method that does not suffer from these problems. The method is typically credited to Hochbaum and Shmoys [31]. The idea is to start by randomly selecting more centers than needed, then pruning some of the centers, and finding remaining centers that maximize the minimum distance between centers already selected. For a detailed description of this initialization method, see, for example, [16].

# CHAPTER 3

# AXIOMS OF CLUSTERING

In his highly influential paper, [35], Kleinberg advocates the development of a theory of clustering that will be "independent of any particular algorithm, objective function, or generative data model." As a step in that direction, Kleinberg sets up a set of axioms aimed to define what a clustering function is. Kleinberg suggests three axioms, each sounding plausible, and shows that these seemingly natural axioms lead to a contradiction - there exists no function that satisfies all three requirements.

Kleinberg's result is often interpreted as stating the impossibility of defining what clustering is, or even of developing a general theory of clustering. We disagree with this view. In this chapter, we show that the impossibility result is, to a large extent, due to the specific formalism used by Kleinberg rather than being an inherent feature of clustering.

Rather than attempting to define what a clustering function is, we turn our attention to the closely related issue of evaluating the quality of a given data clustering. In this chapter we develop a formalism and a consistent axiomatization of that latter notion.

As it turns out, the *clustering-quality* framework is more flexible than that of clustering *functions*. In particular, it allows the postulation of axioms that capture the features that Kleinberg's axioms aim to express, without leading to a contradiction.

A clustering-quality measure is a function that maps pairs of the form (*dataset*, *clustering*) to some ordered set (say, the set of non-negative real numbers), so that these values reflect how 'good' or 'cogent' that clustering is. Formally, a *clustering-quality measure* (CQM) is a function $m$ whose input is $(X, d)$ and a clustering $C$ over $(X, d)$, which returns a non-negative real number, and satisfies some additional requirements. In this chapter we explore the question of what these requirements should be.

Measures for the quality of a clustering are of interest not only as a vehicle for axiomatizing clustering. The need to measure the quality of a given data clustering arises naturally in many clustering contexts. The aim of clustering is to uncover meaningful groups in data. However, not any arbitrary partitioning of a given data set reflects such a structure. Upon obtaining a clustering, usually via some algorithm, a user needs to determine whether this clustering is sufficiently meaningful to rely upon for further analysis or application. Clustering-quality measures aim to answer this need by quantifying how good any specific clustering is.

Clustering-quality measures may also be used to help in clustering model-selection by comparing different clusterings over the same data set (e.g., comparing the results of a given clustering paradigm over different choices of clustering parameters, such as the number of clusters).

Clustering-quality measures corresponding to common objective functions, such as $k$-means or $k$-medoids have some shortcomings for the purpose at hand. Namely, these measures are usually not scale-invariant, and they cannot be used to compare the quality of clusterings obtained by different algorithms aiming to minimize different clustering costs (e.g., $k$-means with different values of $k$). See Section 3.4 for more details.

Clustering quality has been previously discussed in the applied statistics literature, where a variety of techniques for evaluating *'cluster validity'* were proposed. Many of these methods, such as the external criteria discussed in [41], are based on assuming some predetermined data generative model, and as such do not answer our quest for a general theory of clustering. In this work, we are concerned with quality measures regardless of any specific generative model. For examples, see the internal criteria surveyed in [41].

We formulate a theoretical basis for clustering-quality evaluations, proposing a set of requirements ('axioms') of clustering-quality measures. We demonstrate the relevance and consistency of these axioms by showing that the top-performing measures in Milligan's [41] extensive empirical study of internal validity criteria satisfy our axioms.

## 3.1 Kleinberg's Axioms

Kleinberg [35], proposes the following three axioms for clustering functions. These axioms are intended to capture the meaning of clustering by determining which functions are worthy of being considered *clustering functions* and which are not. Kleinberg shows that the set is inconsistent - there exist no functions that satisfy all three axioms.

Let $F$ be a proposed clustering function. The first two axioms require invariance of the clustering that $F$ defines under some changes of the input distance function.

**Function Scale Invariance**: Scale invariance requires that the output of a clustering function be invariant to uniform scaling of the input.

*A function $F$ is* scale-invariant *if for every $(X, d)$ and positive $c$, $F(X, d) = F(X, c \cdot d)$.*

**Function Consistency**: Consistency requires that if within-cluster distances are decreased, and between-cluster distances are increased, then the output of a clustering function does not change. Formally,

- Given a clustering $C$ over $(X, d)$, a distance function $d'$ is $(C, d)$*-consistent*, if $d'(x, y) \leq d(x, y)$ for all $x \sim_C y$, and $d'(x, y) \geq d(x, y)$ for all $x \not\sim_C y$.
- A function $F$ is *consistent* if $F(X, d) = F(X, d')$ whenever $d'$ is $(F(X, d), d)$-consistent.

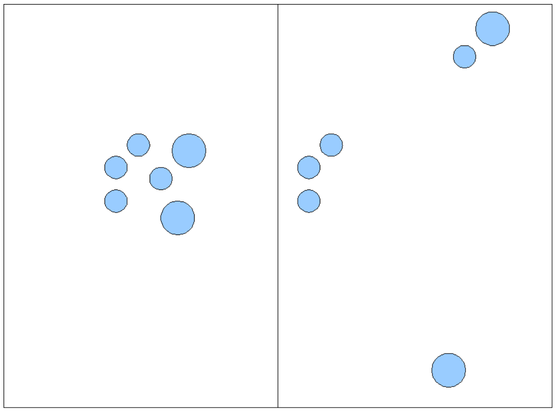


Figure 3.1: A consistent change of a 6-clustering that gives rise to an arguably better 3-clustering.

**Function Richness**: Richness requires that by choosing the right distance function, any partition of the underlying data set can be obtained.

*A function $F$ is* rich *if for each partitioning $C$ of $X$ there exists a distance function $d$ over $X$ so that $F(X, d) = C$.*

**Theorem 1** (Kleinberg, [35])**.** *There exists no clustering function that simultaneously satisfies scale invariance, consistency and richness.*

We strengthen the above result in Chapter 5, by considering a relaxation of the consistency axiom. Our result also provides a simpler proof for this Theorem.

The intuition behind these axioms is rather clear. Let us consider, for example, the Consistency requirement. It seems reasonable that by pulling closer points that are in the same cluster and pushing further apart points in different clusters, our confidence in the given clustering will only rise. However, while this intuition can be readily formulated in terms of clustering quality (namely, "changes such as these should not decrease the quality of a clustering"), the formulation through clustering functions says more. It actually requires that such changes to the underlying distance function should not create any new contenders for the best-clustering of the data.

For example, consider Figure 3.1, where we illustrate a good 6-clustering. On the right hand-side, we show a consistent change of this 6-clustering. Notice that the resulting data has a 3-clustering that may be better than the original 6-clustering. While one may argue that the quality of the original 6-clustering has not decreased as a result of the distance changes, the quality of the 3-clustering has improved beyond that of the 6-clustering. This is the case, for example, when Dunn's index is used (this index is discussed in more detail below). This illustrates a significant weakness of the consistency axiom for clustering *functions.*

The implicit requirement that the original clustering remain the best clustering following a consistent change is at the heart of Kleinberg's impossibility result. As we shall see below, once we relax that extra requirement, the axioms are no longer unsatisfiable.

## 3.2 Axioms of Clustering-Quality Measures

In this section we change the primitive that is being defined by the axioms from clustering functions to Clustering-Quality Measures (CQMs). We reformulate the above three axioms in terms of CQMs and show that this revised formulation is not only consistent, but is also satisfied by a number of natural and effective clustering-quality measures. In addition, we extend the set of axioms by adding another axiom (of clustering-quality measures) that is required to rule out some measures that should not be counted as CQMs.

### 3.2.1 Clustering-Quality Measure Analogues to Kleinberg's Axioms

The translation of the Scale Invariance axiom to the CQM terminology is straightforward:

**Definition 6** (Scale Invariance). *A quality measure $m$ satisfies* scale invariance *if for every clustering $C$ of $(X, d)$, and every positive $c$, $m(C, (X, d)) = m(C, (X, c \cdot d))$.*

The translation of the Consistency axiom is the place where the resulting CQM formulation is indeed weaker than the original axiom for functions. While it clearly captures the intuition that consistent changes to $d$ should not hurt the quality of a given partition, it allows the possibility that, as a result of such a change, some partitions will improve more than others[1].

**Definition 7** (Consistency). *A quality measure $m$ satisfies* consistency *if for every clustering $C$ over $(X, d)$, whenever $d'$ is $(C, d)$-consistent, then $m(C, (X, d')) \geq m(C, (X, d))$.*

**Definition 8** (Richness). *A quality measure $m$ satisfies* richness *if for each non-trivial clustering $C$ of $X$, there exists a distance function $d$ over $X$ such that*

$$C = \mathit{Argmax}_C\{m(C, (X, d)) \mid C \text{ is non-trivial}\}.$$

To demonstrate the consistency of the three axioms for clustering quality measures, we rely on a well-known quality measure, Dunn's index [24].

**Definition 9.** Dunn's index *of a clustering $C$ over $(X, d)$ is*

$$\mathit{Dunn}(C, (X, d)) = \frac{\min_{x \not\sim_C y} d(x, y)}{\max_{x \sim_C y} d(x, y)}.$$

**Theorem 2.** *Consistency, Scale Invariance, and Richness for clustering-quality measures form a consistent set of requirements.*

*Proof.* We show that the three requirements are satisfied by Dunn's index. Note that larger values of Dunn's index indicate better clustering quality. First, we show that Dunn's index satisfies consistency. Let $d'$ be a $(C, d)$-consistent distance function. Consistent changes can only increase between-cluster distances and decrease within-cluster distances. So, $\min_{x \not\sim_C y} d'(x, y) \geq \min_{x \not\sim_C y} d(x, y)$ and $\max_{x \sim_C y} d'(x, y) \leq \max_{x \sim_C y} d(x, y)$. This implies that $\mathit{Dunn}(C, (X, d')) \geq \mathit{Dunn}(C, (X, d))$.

[1] The following formalization assumes that larger values of $m$ indicate better clustering quality. For some quality measures, smaller values indicate better clustering quality, in which case we reverse the direction of inequalities for consistency and use Argmin instead of Argmax for richness.

For richness, given a non-trivial clustering $C$ of a data set $X$, we define a distance function $d$ as follows. For all pairs $x, y \in X$ set $d(x, y) = 1$ if $x \sim_C y$, and set $d(x, y) = 2$ otherwise. Then $Dunn(C, (X, d)) = 2$. Consider any non-trivial clustering $C'$ of $X$ different than $C$. Since $C'$ is both non-trivial and different from $C$, it must either have a within-cluster edge of length 2, or a between-cluster edge of length 1. So Dunn's index of $Dunn(C, (X, d)) \leq 1 < Dunn(C, (X, d))$.

Finally, for any $c > 0$, and any $(X, d)$ and clustering $C$ of $X$, $Dunn(C, (X, c{\cdot}d)) = Dunn(C, (X, d))$. It follows that scale-invariance, consistency, and richness are consistent axioms. □

### 3.2.2 Representation Independence

This axiom resembles the *permutation invariance* objective function axiom by Puzicha et al. [43], modeling the requirement that clustering should be indifferent to the individual identity of clustered elements. This axiom of clustering-quality measures does not have a corresponding Kleinberg axiom.

**Definition 10** (Representation Independence)**.** *A quality measure $m$ is* representation independent *if for all clusterings $C$, $C'$ over $(X, d)$ where $C \cong_C C'$, $m(C, (X, d)) = m(C', (X, d))$.*

**Theorem 3.** *The set of axioms consisting of Representation Independence, Scale Invariance, Consistency, and Richness, (all in their CQM formulation) is a consistent set of axioms.*

*Proof.* Dunn's index satisfies all four axioms. □

## 3.3 Examples of Clustering-Quality Measures

We demonstrate that our proposed axioms of CQMs are satisfied by some measures that have been shown to perform well in practice. In a survey of validity measures, Milligan [41] performs an extensive empirical study of internal validity indices. His study compared how clustering quality measures compare with external validity indices (namely Rand and Jaccard) on a large number of data sets where a correct clustering is known. We show here that the top performing internal criteria examined satisfy our axioms.

### 3.3.1 Gamma

The Gamma measure was proposed as a CQM by Baker and Hubert [10] and it is the top performing measure in Milligan's [41] study. Let $d(+)$ denote the number of times that a between-cluster edge is larger than a within-cluster edge, and let $d(-)$ denote the opposite result.

Formally,

$$d(+) = |\{\{x, y, x', y'\} \subseteq X \mid x \sim_C y, x' \not\sim_C y', d(x, y) < d(x', y')\}|,$$

and

$$d(-) = |\{\{x, y, x', y'\} \subseteq X \mid x \sim_C y, x' \not\sim_C y', d(x, y) \geq d(x', y')\}|.$$

The *Gamma* clustering-quality measure is defined as

$$\frac{d(+) - d(-)}{d(+) + d(-)}.$$

The range of values of gamma is [0,1] and larger values indicate greater clustering quality. We show that Gamma satisfies our four axioms of clustering-quality measures.

**Theorem 4.** *Gamma satisfies consistency, richness, scale-invariance, and representation independence.*

*Proof.* To see that Gamma is consistent, observe that a consistent change can only increase $d(+)$ and decrease $d(-)$, thus increasing the numerator of Gamma. However, since the number of within-cluster pairs and between-cluster pairs remains unchanged, the denominator is unaffected. As such, the value of Gamma cannot decrease following a consistent change.

To see that richness is satisfied, consider any non-trivial clustering $C$ of $X$. Construct a distance function $d$ as follows: if $x \sim_C y$, then set $d(x, y) = 1$, and otherwise, set $d(x, y) = 2$. Then Gamma of $C$ is 1, the largest possible value of Gamma, since all within-cluster distances are smaller than all between-cluster distances. Observe that $C$ is the only clustering of $(X, d)$ with $d(-) = 0$. It follows that the Gamma of any other non-trivial clustering of $(X, d)$ is strictly smaller than 1.

Since uniform scaling preserves the order of pairwise distances, the Gamma measure is unaffected by uniform scaling of the distance function, and so the measure is scale invariant. Gamma is also representation independent as it does not depend on the labellings of the data. □

### 3.3.2 C-Index

C-Index is the second-best performing quality measure in Milligan's study. The measure was introduced by Hubert and Levin [32]. Let $d_w(C, d)$ denote the sum of within-cluster distances,

$$d_w(C, d) = \sum_{x \sim_C y} d(x, y).$$

Let $n_w$ be the number of within-cluster pairs in $C$, that is $n_w = |\{\{x, y\} \subseteq X \mid x \sim_C y\}|$.

Let $S_{min}$ denote the set of $n_w$ minimal distances in $d$, and let $S_{max}$ denote the set of $n_w$ maximal distances in $d$. Further, let $min(n_w, d)$ be the sum of the distances in $S_{min}$, and let $max(n_w, d)$ be the sum of the distances in $S_{max}$. The c-index is defined as follows.

**Definition 11** (C-Index)**.** *The* c-index *of a clustering $C$ over $(X, d)$ is*

$$\frac{d_w(C, d) - min(n_w, d)}{max(n_w, d) - min(n_w, d)}.$$

The range is $[0, 1]$, and smaller values indicate better clustering quality.

**Lemma 1.** *C-index is consistent.*

*Proof.* A consistent change can be viewed as a series of changes each affecting a single edge. We consider consistent changes that modify a single edge, for all possible choices of that edge.

In the following, we first assume that the sets $S_{min}$ and $S_{max}$ are unmodified following the consistent change; that is, if $e$ was in one or both of these sets, then it remains such.

*Within-cluster edges:* Following a consistent change, a within-cluster edge $e$ either shrinks, or remains unchanged. Suppose that $e \in S_{max}$ and $e \notin S_{min}$. Then the numerator and denominator of the c-index decrease by the same amount. We show that such change can only decrease the c-index, improving the quality of the clustering.

Let $a = d_w(C, d) - min(n_w, d)$, the numerator of the c-index. Let $b = max(n_w, d) - min(n_w, d)$, the denominator of the c-index. Let $\alpha > 0$ be the amount by which the length of the edge $e$ decreases after the consistent change. Then the new value of the c-index, following the consistent change, is $\frac{a-\alpha}{b-\alpha}$. We show that $\frac{a-\alpha}{b-\alpha} < \frac{a}{b}$. Since $max(n_w, d) \geq d_w(C, d)$, it follows that $\frac{a-\alpha}{b-\alpha} \leq \frac{a}{b}$.

Suppose that $e \in S_{min}$ and $e \notin S_{max}$. Then $min(n_w, d)$ and $d_w(C, d)$ decrease by the same amount, and so the change is removed in the subtraction, not effecting the numerator. The denominator increases after such a consistent change, causing the c-index to decrease, improving the quality.

If $e \in S_{max} \cap S_{min}$, then shrinking that edges does not effect the value of the c-index, by leaving both the numerator and denominator unchanged. Finally, if $e \notin S_{max}$ and $e \notin S_{min}$, then only the numerator can decrease, and so the c-index can only improve.

*Between-cluster edges:* Such edges leave $d_w(C, d)$ unchanged, and can only increase following a consistent change. If $e \in S_{max}$ and $e \notin S_{min}$, then it improves the clustering quality by increasing the denominator.

If $e \in S_{min}$ and $e \notin S_{max}$, then the numerator and denominator decrease by the same amount, improving the quality by same argument as for the within-cluster edge case where $e \in S_{max}$ and $e \notin S_{min}$.

If $e \in S_{max} \cap S_{min}$, the numerator decreases, and the denominator does not change, improving the quality of the partition. Finally, if $e \notin S_{max}$ and $e \notin S_{min}$ then the c-index is unchanged.

Finally, consider what happens if the sets $S_{min}$ and $S_{min}$ can be modified following a consistent change. First, let's consider what happens if $S_{max}$ changes. Then $max(n_w, d)$ can only get larger, which can only decrease the c-index. If $S_{min}$ changes, then it must be a between-cluster edge as it has grown. This causes a new edge, previously larger than $e$, to be added to $S_{min}$, and so $min(n_w, d)$ becomes larger than it was before the consistent change, which decrease the c-index. □

**Theorem 5.** *C-index satisfies the four axioms of clustering-quality measures.*

*Proof.* Consistency follows by Lemma 3.3.2. To see that richness is satisfied, consider any non-trivial clustering $C$ of $X$. Construct a distance function $d$ as follows: if $x \sim_C y$, then set $d(x, y) = 1$, otherwise, set $d(x, y) = 2$. Then the c-index of clustering $C$ is 0, the minimal possible $C$-index value. Now consider any other non-trivial clustering $C' \neq C$ of $X$, any constant $c > 0$ and any data set $(X, d)$. Then the c-index of data set $(X, c \cdot d)$ = c-index$(X, d)$. Finally, the c-index is representation independent since it does not rely on data labellings. □

## 3.4 Dependence on Number of Clusters

The clustering-quality measures discussed here up to now are independent of the number of clusters, which enables the comparison of clusterings with a different number of clusters. In this section we discuss an alternative type of clustering quality evaluation, that depends on the number of clusters. Such quality measures arise naturally from common loss functions (or, objective functions) that drive clustering algorithms, such as $k$-means or $k$-medoids.

These common loss functions fail to satisfy two of our axioms, scale-invariance and richness. One can easily overcome the dependence on scaling by normalization. As we will show, the resulting normalized loss functions make a different type of clustering-quality measures from the measures we previously discussed, due to their dependence on the number of clusters.

A natural remedy to the failure of scale invariance is to normalize a loss function by dividing it by the variance of the data, or alternatively, by the loss of the 1-clustering of the data.

Common loss functions, even after normalization, usually have a bias towards more refined clusterings – they assign lower cost (that is, higher quality) to more refined clusterings. This prevents using them as a meaningful tool for comparing the quality of clusterings with different number of clusters. We formalize this feature of common clustering loss functions through the notion of *refinement preference*[2]:

**Definition 12** (Refinement)**.** *For a pair of clusterings $C, C'$ of the same domain, clustering $C'$ is a* refinement *of $C$ if every cluster in $C$ is a union of clusters of $C'$.*

**Definition 13** (Refinement Preference)**.** *A measure $m$ is* refinement preferring *if for every $(X, d)$ and every clustering $C$ of $(X, d)$ that has a non-trivial refinement different from $C$, there exists some non-trivial refinement $C'$ such that $m(C', (X, d)) < m(C, (X, d))$.*

We show that several well-known objective functions are refinement preferring. Recall the min-sum clustering functions defined in the Preliminaries.

**Theorem 6.** *Min-sum and k-medoids are refinement preferring.*

*Proof.* Consider any clustering $C$ that has a non-trivial refinement different from $C$. Given any refinement $C'$ of $C$, its set of within-cluster distances is a strict subset of that of $C$. Since min-sum is the sum of within-cluster distances, and all distances are positive, it follows that the min-sum cost of $C'$ is lower than that of $C$.

To see that k-medoids is refinement preferring, consider a non-trivial refinement $C'$ of $C$ that has only one within-cluster pair $(x, y)$ where $x$ is a cluster center in $C$. The $k$-medoids cost of $C'$ is $d(x, y)^2$. The clustering $C$ has a within-cluster pair additional to $(x, y)$, and so it follows that the $k$-medoids cost of $C$ is greater than $d(x, y)^2$. □

**Theorem 7.** *If the data lies in a normed vector space, then k-means is refinement preferring.*

[2] The following formalization assumes that lower scores indicate better clustering quality. If higher scores indicate better clustering quality, reverse the direction of the inequality.

*Proof.* Consider any clustering $C$ that has a non-trivial refinement different from $C$. Let $C'$ be any refinement of $C$ that has a single within-cluster pair $(x, y)$. Then $x$ and $y$ share some center $c$ in $C$, and so their contribution to $C$ is $\|x - c\|^2 + \|y - c\|^2$, implying that the cost of $C$ is greater than $\|x - c\|^2 + \|y - c\|^2$ as it has within-cluster pairs other than $(x, y)$. The cost of $C'$ is at most $\|x - c\|^2 + \|y - c\|^2$, since using the center of mass of $\{x, y\}$ instead of $c$ can only improve the cost of $C'$. □

We now show that refinement preferring measures fail to satisfy the richness axiom.

**Theorem 8.** *If a quality measure $m$ is refinement preferring, then it fails the richness axiom.*

*Proof.* Let $m$ be a refinement preferring measure. The richness axiom requires that for every domain set $X$, and every non-trivial clustering $C$ of $X$, there exists a distance function $d$ so that $C$ has optimal $m$ value over all clusterings of $(X, d)$.

Let $C$ be any clustering that has a non-trivial refinement different from $C$. Note that this property is independent of any distance function. Then since $m$ is refinement preferring, for any distance function $d$ of $X$, there exists some refinement $C'$ of $C$ that achieves a better score than $C$. It follows that $m$ is not rich. □

Many common clustering quality measures satisfy one of richness or refinement preference, but as shown above, no measure can satisfy both. To evaluate the quality of a clustering using a refinement preferring measure, the number of clusters should be fixed. Since the correct number of clusters is often unknown, measures that are independent of the number of clusters apply in a more general setting.

# CHAPTER 4

# A CHARACTERIZATION OF LINKAGE-BASED ALGORITHMS

In spite of the wide use of clustering in many practical applications, currently, there exists no principled method to guide the selection of a clustering algorithm. Of course, users are aware of the costs involved in employing different clustering algorithms (software purchasing costs, running times, memory requirements, needs for data preprocessing *etc.*) but there is very little understanding of the differences in the *outcomes* that these algorithms may produce. We focus on that aspect: The input-output properties of different clustering algorithms.

The choice of an appropriate clustering method should, of course, be task dependent. A clustering method that works well for one task may be unsuitable for another. Even more than for supervised learning, for clustering, the choice of an algorithm must incorporate domain knowledge. While some domain knowledge is embedded in the choice of similarity between domain elements (or the embedding of these elements into some Euclidean space), there is still a large variance in the behavior of difference clustering paradigms over a fixed similarity measure.

For some clustering tasks, there is a natural clustering objective function that one may wish to optimize, but very often the task does not readily translate into a corresponding objective function. Often users are merely searching for a meaningful clustering, without a prior preference for any specific objective function. Many common clustering paradigms do not optimize any clearly defined objective function, either because no such objective is defined (as in the case of, for example, single linkage clustering) or because optimizing the most relevant objective is computationally infeasible. To overcome computation infeasibility, the algorithms end up carrying out heuristics whose outcomes may be quite different than the actual objective-based optimum. What seems to be missing is a clear understanding of the differences in clustering outputs in terms of intuitive and usable properties.

Some heuristics have been proposed as a means of distinguishing between the output of clustering algorithms on specific data. These approaches require running the algorithms, and then selecting an algorithm based on the outputs that they produce. In particular, validity criteria can be used to evaluate the output of clustering algorithms. These measures can be used to select a clustering algorithm by choosing the one that yields the highest quality clustering [44]. However, the result only applies to single sets of data, and there are no guarantees on the quality of the output of these algorithms on any other data.

We propose a different approach to providing guidance to clustering users by identifying significant properties of the input-output behaviour of clustering functions that, on one hand distinguish between different clustering paradigms, and on the other hand are intended to be relevant to the domain knowledge that a user might have access to. Based on domain expertise, users could then choose which properties they want an algorithm to satisfy, and determine which algorithms meet their requirements.

In this chapter, we make a major step by distilling a set of abstract properties that distinguish between linkage based clustering and any other type of clustering paradigm. Linkage based clustering is a family of clustering methods that include some of the most commonly-used and widely-studied clustering paradigms. We provide a surprisingly simple set of properties that, on one hand is satisfied by all these algorithm in that family. On the other hand, no algorithm outside that family satisfies (all of) the properties in that set. Our characterization highlights the way in which the clusterings that are output by linkage based algorithms are different from the clusterings output by all other clustering algorithms.

## 4.1 Defining Linkage Based Clustering

A linkage based algorithm begins by placing every element of the input data set into its own cluster, and then repeatedly merging the "closest" clusters until some stopping criteria is met. We rely on the *k-stopping criteria*, which requires that exactly $k$ clusters have been formed. What distinguishes different linkage based algorithms from each other is the definition of between-cluster distance, which is used to determine the closest clusters. For example, *single linkage* defines cluster distance by the shortest edge between members of the clusters, while *complete linkage* uses the longest between-cluster edge to define the distance between clusters.

Between-cluster distance has been formalized in a variety of ways. It has been called a "linkage function," (see, for example, [23] and [29]). Everitt et al. [26] call it "inter-object distance." Common to all these formalisms is a function that maps pairs of clusters to real numbers. No further detailing of the concept has been previously explored. We zoom in on the concept of between-cluster distance and provide a rigorous, general definition.

**Definition 14** (Linkage function)**.** *A* linkage function *is a function*

$$\ell : \{(X_1, X_2, d) \mid d \textit{ is a distance function over } X_1 \cup X_2\} \rightarrow \mathbb{R}^+$$

*such that,*

1. *$\ell$ is* representation independent*: For all $(X_1, X_2)$ and $(X_1', X_2')$, if $(\{X_1, X_2\}, d) \cong_C (\{X_1', X_2'\}, d')$ then $\ell(X_1, X_2, d) = \ell(X_1', X_2', d')$.*

2. *$\ell$ is* monotonic*: For all $(X_1, X_2, d)$ if $d'$ is a distance function over $X_1 \cup X_2$ such that for all $x \sim_{\{X_1,X_2\}} y$, $d(x,y) = d'(x,y)$ and for all $x \not\sim_{\{X_1,X_2\}} y$, $d(x,y) \leq d'(x,y)$ then $\ell(X_1, X_2, d') \geq \ell(X_1, X_2, d)$.*

3. *Any pair of clusters can be made arbitrarily distant: For any pair of data sets $(X_1, d_1)$, $(X_2, d_2)$, and any $r$ in the range of $\ell$, there exists a distance function $d$ that extends $d_1$ and $d_2$ such that $\ell(X_1, X_2, d) > r$.*

*For technical reasons, we shall assume that a linkage function has a countable range. Say, the set of non-negative algebraic real numbers*[1].

Note that a linkage function is only given the data for two clusters, as such, the distance between two clusters does not depend on data that is outside these clusters. Condition (1) formalizes the requirement that the distance does not depend on the labels (or identities) of domain points. The between-cluster distance is fully determined by the matrix of between-point distances. Conditions (2) and (3) relate the linkage function to the input distance function, and capture the intuition that pulling the points of one cluster further apart from those of another cluster would not make the two clusters closer.

We now define linkage based $k$-clustering functions.

**Definition 15** (linkage based clustering function)**.** *A $k$-clustering function $F$ is* linkage based *if there exists a linkage function $\ell$ so that*

- $F(X, d, |X|) = \{\{x\} \mid x \in X\}$
- *For $1 \leq k < |X|$, $F(X, d, k)$ is constructed by merging the two clusters in $F(X, d, k+1)$ that minimize the value of $\ell$. Formally,*

  $$F(X, d, k) = \{C_i \mid C_i \in F(X, d, k+1), C_i \neq C_1, C \neq C_2\} \cup \{C_1 \cup C_2\},$$

  *such that $\{C_1, C_2\} = argmin_{\{C_1, C_2\} \subseteq F(X,d,k+1)} \ell(C_1, C_2, d)$.*

Here are examples of linkage functions used in the most common linkage based algorithms.

- *Single linkage:* $\ell_{SL}(A, B, d) = \min_{a \in A, b \in B} d(a, b)$.
- *Average linkage:* $\ell_{AL}(A, B, d) = \frac{\sum_{a \in A, b \in B} d(a,b)}{|A| \cdot |B|}$
- *Complete linkage:* $\ell_{CL}(A, B, d) = \max_{a \in A, b \in B} d(a, b)$.

Note that $\ell_{SL}$, $\ell_{AL}$, and $\ell_{CL}$ satisfy the conditions of Definition 14 and as such are linkage functions[2].

[1] Imposing this restriction simplifies our main proof, while not having any meaningful impact on the scope of clusterings considered.

[2] A tie breaking mechanism is often used to apply such linkage functions in practice. For simplicity, we assume in this discussion that no ties occur. In other words, we assume that the linkage function is one-to-one on the set of isomorphism-equivalence classes of pairs of clusters.

## 4.2 Properties of $k$-Clustering Functions

In this chapter, we require that $k$-clustering functions satisfy two natural requirements, presentation independence and scale-invariance. As will be shown in Chapter 5, all $k$-clustering functions that we consider satisfy these two properties.

**Definition 16** (Clustering functions)**.** *A* $k$-clustering function *is a function that takes as input a pair $(X, d)$ and a parameter $1 \leq k \leq |X|$ and outputs a $k$-clustering of the domain $X$. We require such a function, $F$, to satisfy the following:*

*1.* Representation Independence*: Whenever $(X, d) \sim (X', d')$, then, for every $k$, $F(X, d, k)$ and $F(X', d', k)$ are isomorphic clusterings.*

*2.* Scale Invariance*: For any domain set $X$ and any pair of distance functions $d, d'$ over $X$, if there exists $c \in \mathbb{R}^+$ such that $d(a, b) = c \cdot d'(a, b)$ for all $a, b \in X$, then $F(X, d, k) = F(X, d', k)$.*

We now introduce properties of $k$-clustering functions that we use to characterize linkage based clustering.

### 4.2.1 Locality

We now introduce a new property of clustering algorithms that we call "locality". Intuitively, a $k$-clustering function is local if its behavior on a union of a subset of the clusters (in a clustering it outputs) depends only on distances between elements of that union, and is independent of the rest of the domain set.

**Definition 17** (Locality)**.** *A $k$-clustering function $F$ is* local *if for any clustering $C$ output by $F$ and every subset of clusters, $C' \subseteq C$,*

$$F(\bigcup C', d, |C'|) = C'.$$

In other words, for every domain $(X, d)$ and number of clusters, $k$, if $X'$ is the union of $k'$ clusters in $F(X, d, k)$ for some $k' \leq k$, then, applying $F$ to $(X', d)$ and asking for a $k'$-clustering, will yield the same clusters that we started with.

To better understand locality, consider two runs of a clustering algorithm. In the first run, the algorithm is called on some data set $X$ and returns a $k$-clustering $C$. We then select some clusters $C_1, C_2, \ldots, C_{k'}$ of $C$, and run the clustering algorithm on the points that the selected clusters contains, namely, $C_1 \cup C_2 \cup \ldots \cup C_{k'}$, asking for $k'$ clusters. If the algorithm is local, then on the second run of the algorithm it will output $\{C_1, C_2, \ldots, C_{k'}\}$.

### 4.2.2 Consistency

Consistency, introduced by Kleinberg [35], requires that the output of a clustering function, be invariant to shrinking within-cluster distances, and stretching between-cluster distances. The following is a translation of consistency into the setting of $k$-clustering functions.

**Definition 18** (Consistency)**.** *Given a clustering $C$ of some domain $(X, d)$, we say that a distance function $d'$ over $X$, is $(C, d)$-*consistent *if*

1. *$d'(x, y) \leq d(x, y)$ whenever $x \sim_C y$, and*
2. *$d'(x, y) \geq d(x, y)$ whenever $x \not\sim_C y$.*

*A clustering function $F$ is* consistent *if for every $X, d, k$, if $d'$ is $(F(X, d, k), d)$-consistent then $F(X, d, k) = F(X, d', k)$.*

We introduce two relaxations of consistency for $k$-clustering functions.

**Definition 19** (Outer Consistency)**.** *Given a clustering $C$ of some domain $(X, d)$, we say that a distance function $d'$ over $X$, is $(C, d)$-*outer-consistent *if*

1. *$d'(x, y) = d(x, y)$ whenever $x \sim_C y$, and*
2. *$d'(x, y) \geq d(x, y)$ whenever $x \not\sim_C y$.*

*A $k$-clustering function $F$ is* outer consistent *if for every $X, d, k$, if $d'$ is $(F(X, d, k), d)$-outer-consistent then $F(X, d, k) = F(X, d', k)$.*

**Definition 20** (Inner Consistency)**.**

*Given a clustering $C$ of some domain $(X, d)$, we say that a distance function $d'$ over $X$, is $(C, d)$-*inner-consistent *if*

1. *$d'(x, y) \leq d(x, y)$ whenever $x \sim_C y$, and*
2. *$d'(x, y) = d(x, y)$ whenever $x \not\sim_C y$.*

*A $k$-clustering function $F$ is* inner consistent *if for every $X, d, k$, if $d'$ is $(F(X, d, k), d)$-inner consistent then $F(X, d, k) = F(X, d', k)$.*

Clearly, consistency implies both outer-consistency and inner-consistency.

As will be shown in chapter 5, outer-consistency is satisfied by many common $k$-clustering functions. We will also show that average-linkage and complete-linkage are not inner consistent, and therefore they are not consistent. In Lemma 12 of this chapter, we will show that any linkage based $k$-clustering function is outer-consistent.

### 4.2.3 Richness

We propose an extension on Kleinberg's richness axiom. A $k$-clustering function satisfies outer richness if for every finite collection of disjoint domain sets (each with its own distance function), by setting the distances between the data sets, we can get $F$ to output each of these data sets as a cluster. This corresponds to the intuition that if groups of points are moved sufficiently far apart, then they will be placed in separate clusters.

**Definition 21** (Outer Richness). *For every set of domains, $\{(X_1, d_1), \dots (X_n, d_n)\}$, there exists a distance function $\hat{d}$ over $\bigcup_{i=1}^{n} X_i$ that extends each of the $d_i$'s (for $i \leq n$), such that $F(\bigcup_{i=1}^{n} X_i, \hat{d}, n) = \{X_1, X_2, \dots, X_n\}$.*

The corresponding definition of this property using similarities instead of distances requires that there be no within-cluster pairs with 0 similarity, as this would correspond to infinite distance between these elements, which cannot be represented using distances in this framework.

### 4.2.4 Refinement Preserving

Recall the definition of a clustering refinement. A clustering $C'$ of $X$ is a *refinement* of a clustering $C$ of $X$ if every cluster of $C$ is a union of clusters of $C'$.

We now introduce our final property, requiring that as the number of clusters increases, the $k$-clustering function continues to refine the same clustering.

**Definition 22** (Refinement Preserving Functions). *A $k$-clustering function is* refinement preserving *if for every $1 \leq k \leq k' \leq |X|$, $F(X, d, k')$ is a refinement of $F(X, d, k)$.*

## 4.3 Main Result

Our main result specifies properties that uniquely identify linkage based $k$-clustering functions.

**Theorem 9.** *A $k$-clustering function is linkage based if and only if it is refinement-preserving and it satisfies: Outer Consistency, Locality and Outer Richness.*

We divide the proof into the following two sub-sections (one for each direction of the "if and only if").

### 4.3.1 The Properties Imply that the Function is Linkage Based

We show that if $F$ satisfies the prescribed properties, then there exists a linkage function that, plugged into the procedure in the definition of a linkage based function, will yield the same output as $F$ (for every input $(X, d)$ and $k$).

**Lemma 2.** *If a $k$-clustering function $F$ is refinement preserving and it satisfies Outer Consistency, Locality and Outer Richness, then $F$ is linkage based.*

The proof comprises the rest of this section.

*Proof.* Since $F$ is refinement-preserving, for every $1 \leq k < |X|$, $F(X, d, k)$ can be constructed from $F(X, d, k+1)$ by merging two clusters in $F(X, d, k+1)$. It remains to show that there exists a linkage function that determines which clusters to merge.

Due to the representation independence of $F$, one can assume w.l.o.g., that the domain sets over which $F$ is defined are (finite) subsets of the set of natural numbers, $\mathcal{N}$.

**Definition 23** (The (pseudo-) partial ordering $<_F$). *$<_F$ is a binary relation over equivalence classes, with respect to clustering-isomorphism. Two triples are equivalent $(A, B, d) \cong (A', B', d')$ if they are isomorphic as clusters, namely, if $(\{A, B\}, d) \cong_C (\{A', B'\}, d')$. We denote equivalence classes by square brackets. So, the domain of $<_F$ is*

$$\{[A, B, d] : A \subseteq \mathcal{N}, B \subseteq \mathcal{N}, A \cap B = \emptyset \textit{ and } d \textit{ is a distance function over } A \cup B\}.$$

*We define it by: $[(A, B, d)] <_F [(A', B', d')]$ if there exists a distance function $d^*$ over $X = A \cup B \cup A' \cup B'$ that extends both $d$ and $d'$, and there exists $k \in \{2, 3\}$ such that*

1. *$A, B, A', B' \in F(X, d^*, k+1)$*
2. *$A \cup B \in F(X, d^*, k)$*
3. *For all $D \in \{A, B, A', B'\}$, either $D \subseteq A \cup B$ or $D \in F(X, d^*, k)$.*

The definition consists of two cases, one for $k = 2$ and one for $k = 3$. If $k = 3$, then the sets $A, B, A', B'$ are all distinct, $F(X, d^*, 4) = \{A, B, A', B'\}$ and $F(X, d^*, 3) = \{A \cup B, A', B'\}$.

If $k = 2$, then either $A = A'$, $B = B'$, $A = B'$, or $B = A'$. Without loss of generality, assume that $A = A'$. Then $F(X, d^*, 3) = \{A, B, B'\}$ and $F(X, d^*, 2) = \{A \cup B, B'\}$.

Intuitively, $(A, B, d) <_F (A', B', d')$, if there is an input for which $F$ creates the clusters $A, B, A', B'$ as members of some clustering $F(X, d^*, k+1)$, then $F(X, d^*, k)$ merges $A$ with $B$ (before it merges $A'$ and $B'$).

The relation is well defined thanks to the assumption that $F$ is representation independent. For the sake of simplifying notation, we will omit the square brackets in the following discussion.

First, we show that for singleton sets $<_F$ respects the input distance function, $d$.

**Lemma 3.** *For every $x, y, x', y'$, such that $x \neq y$ and $x' \neq y'$, every value $d_1(x, y)$ and $d_2(x', y')$, and every refinement-preserving $k$-clustering function $F$ that satisfies outer-consistency, locality, and outer-richness,*

$$(\{x'\}, \{y'\}, d_2) <_F (\{x\}, \{y\}, d_1) \textit{ if and only if } \ d_2(x', y') < d_1(x, y).$$

*Proof.* Consider a data set on 4 points, $S = \{x, y, x', y'\}$. Let $b = d_1(x, y)$ and $a = d_2(x', y')$ and where $b > a$.

By outer richness, there exists a distance function $d$ that extends $d_1$ and $d_2$ so that $F(S, d, 2) = \{\{x, y\}, \{x', y'\}\}$. Since $F$ is outer-consistent, we can assume that $d(x, x') = d(x, y') = d(y, x') =$

$d(y,y') = D$ for some large $D$ greater than both $a$ and $b$. Since $F$ is refinement preserving it outputs either $\{\{x,y\},\{x'\},\{y'\}\}$ or $\{\{x\},\{y\},\{x',y'\}\}$ for $k=3$. It follows that either $(\{x\},\{y\},d_1) <_F (\{x'\},\{y'\},d_2)$ or $(\{x'\},\{y'\},d_2) <_F (\{x\},\{y\},d_1)$. By way of contradiction, assume that $F(S,d,3) = \{\{x,y\},\{x'\},\{y'\}\}$, which would imply that $(\{x\},\{y\},d_1) <_F (\{x'\},\{y'\},d_2)$ while $a = d_2(x',y') < d_1(x,y) = b$.

Set $c = b/a$. Note that $c > 1$. Let $d'$ be such that $d'(x,y) = b$, $d'(x',y') = cb$, $d'(p,q) = D$ for all other pairs of elements in $S$. Then $d'$ is $(F(S,d,3),d)$-outer-consistent. Since $F$ is outer-consistent, $F(S,d',3) = F(S,d,3)$. Next, consider the distance function $d''$ so that $d''(p,q) = (1/c)\cdot d'(p,q)$ for all $p,q \in S$. Since $F$ is scale invariant, by condition 2 of Definition 16, $F(S,d'',3) = F(S,d,3)$. Finally, let $d'''$ be such that $d'''(x',y') = b$, $d'''(x,y) = a$ and $d'''(p,q) = D$ for all $\{p,q\} \neq \{x',y'\}$. Note that $d'''$ is $(F(S,d'',3),d'')$-outer-consistent. Therefore, $F(S,d''',3) = F(S,d,3) = \{\{x,y\},\{x'\},\{y'\}\}$. Note that $d'''$ and $d$ are isomorphic up to relabelling, by switching $x$ with $x'$ and $y$ with $y'$. If follows that $F(S,d''',3)$ should be $\{\{x',y'\},\{x\},\{y\}\}$ - a contradiction.

□

To show that $<_F$ can be extended to a partial ordering, we first show that it is cycle-free.

**Lemma 4.** *Given a k-clustering function F that is outer-consistent, refinement-preserving, local and satisfies outer richness, there exists no finite sequence $(A_1,B_1,d_1)....(A_n,B_n,d_n)$, where $n > 2$, such that for all $1 \leq i < n$,*

1. *$A_i \cap B_i = \emptyset$,*
2. *$d_i$ is a distance function over $A_i \cup B_i$ and*
3. *$(A_i,B_i,d_i) <_F (A_{i+1},B_{i+1},d_{i+1})$*

*and $(A_1,B_1,d_1) = (A_n,B_n,d_n)$.*

*Proof.* Assume that such a sequence exists. Let $C_i = A_i \cup B_i$ and $X = \bigcup_{i=1}^n C_i$.

Using outer richness, we can construct $\hat{d}$ from the given set of domains $(C_i,d_i)$, for all $1 \leq i \leq n$, that extends all of the distances, such that $F(X,\hat{d},n) = \{C_1,C_2,\ldots,C_n\}$.

Let us consider what happens for $F(X,\hat{d},n+1)$. Since $F$ is refinement-preserving, the $(n+1)$-clustering must split one of the $C_i$'s. Given $1 \leq i < n$, we will show that one cannot split $C_i$ without causing a contradiction.

Recall that $(A_i,B_i,d_i) <_F (A_{i+1},B_{i+1},d_{i+1})$, and thus there exists a distance function $d'$ that extends $d_i$ and $d_{i+1}$ over $X' = A_i \cup B_i \cup A_{i+1} \cup B_{i+1}$, and $k \in \{2,3\}$, such that $A_i, B_i, A_{i+1}, B_{i+1} \in F(X',d',k+1)$, $A_i \cup B_i \in F(X',d',k)$ and for all $D \in \{A_i,B_i,A_{i+1},B_{i+1}\}$, either $D \subseteq A_i \cup B_i$ or $D \in F(X',d',k)$.

First, we will show that $C_i$ must be split into $A_i$ and $B_i$. Consider $F(C_i,d_i,2)$. Since $(A_i,B_i,d_i) <_F (A_{i+1},B_{i+1},d_{i+1})$, we know that $F(C_i,d_i,2) = \{A_i,B_i\}$, by locality.

Now we will show that splitting $C_i$ into $A_i$ and $B_i$ violates $(A_i,B_i,d_i) <_F (A_{i+1},B_{i+1},d_{i+1})$. Using locality, we focus on the data points in $C_i \cup C_{i+1}$. By locality, for some $k \in \{2,3\}$, $A_i, B_i \in$

$F(C_i \cup C_{i+1}, \hat{d}/C_i \cup C_{i+1}, k)$. At this point, the distances defined by $\hat{d}$ between $C_i$ and $C_{i+1}$ may be different from those defined in $d'$.

Using outer consistency, we define distance function $\tilde{d}$ over $X'$ that is both $(F(C_i \cup C_{i+1}, \hat{d}/C_i \cup C_{i+1}, k), \hat{d}/C_i \cup C_{i+1})$-outer-consistent and $(F(C_i \cup C_{i+1}, d', k), d')$-outer-consistent.

First, let $m_1 = \max\{\hat{d}(x,y) \mid x, y \in C_i \cup C_{i+1}\}$ and let

$m_2 = \max\{d'(x,y) \mid x, y \in C_i \cup C_{i+1}\}$. Finally, let $m^* = \max\{m_1, m_2\}$. Now, we defined $\tilde{d}$ as follows:

$$\tilde{d}(x,y) = \begin{cases} \hat{d}(x,y) & \text{if } x, y \in C_i \text{or } x, y \in C_{i+1} \\ m^* & \text{otherwise} \end{cases}$$

It is clear that $\tilde{d}$ meets our requirements. By outer consistency, $F(C_i \cup C_{i+1}, \tilde{d}, k) = F(C_i \cup C_{i+1}, \hat{d}/C_i \cup C_{i+1}, k)$, in which we showed that $A_i$ and $B_i$ are separate clusters. Also by outer consistency, $F(C_i \cup C_{i+1}, \tilde{d}, k) = F(C_i \cup C_{i+1}, d', k)$, in which $A_i$ and $B_i$ are part of the same cluster by the ordering $<_F$. Thus, we have a contradiction because $C_i \neq C_{i+1}$.

□

Note that for $n = 3$, the above Lemma shows antisymmetry of $<_F$.

We make use of the following general result.

**Lemma 5.** *For any cycle-free, anti-symmetric relation $P( \ , \ )$ over a finite or countable domain $D$ there exists an embedding $h$ into $\mathbb{R}^+$ so that for all $x, y \in D$, if $P(x,y)$ then $h(x) < h(y)$.*

*Proof.* First we convert the relation $P$ into a partial order by defining $a < b$ whenever there exists a sequence $x_1, \ldots, x_k$ so that $P(a, x_1), P(x_2, x_3), \ldots, P(x_k, b)$. This is a partial ordering because $P$ is antisymmetric and cycle-free. To map the partial order to the positive reals, we first enumerate the elements, which can be done because the domain is countable. The first element is then mapped to any value, $\phi(x_1)$. By induction, we assume that the first $n$ elements are mapped in an order preserving manner. Let $x_{i_1} \ldots x_{i_k}$ be all the members of $\{x_1, \ldots, x_n\}$ that are below $x_{n+1}$ in the partial order. Let $r_1 = \max\{\phi(x_{i_1}), \ldots, \phi(x_{i_k}\}$, and similarly let $r_2$ be the minimum among the images of all the members of $\{x_1, \ldots, x_k\}$ that are above $x_{n+1}$ in the partial order. Finally, let $\phi(x_{n+1})$ be any real number between $r_1$ and $r_2$. It is easy to see that now $\phi$ maps $\{x_1, \ldots, x_n, x_{n+1}\}$ in a way that respects the partial order. □

Finally, we define our linkage function by embedding the equivalence classes of triples into the positive real numbers in an order preserving way, as implied by applying Lemma 5 to $<_F$. Namely, $\ell_F : \{[(A, B, d)] : A \subseteq \mathcal{N}, B \subseteq \mathcal{N}, A \cap B = \emptyset \textit{ and } d \textit{ is a distance function over } A \cup B\} \to \mathbb{R}^+$ so that $[(A, B, d)] <_F [(A', B', d')]$ implies $\ell_F[(A, B, d)] < \ell_F[(A, B, d)]$.

**Lemma 6.** *The function $\ell_F$ is a linkage function for any refinement-preserving function $F$ that satisfies locality, outer-consistency, and outer richness.*

*Proof.* $\ell_F$ satisfies condition 1 of Definition 14 since it is defined on equivalence classes of isomorphic sets. The function $\ell_F$ satisfies condition 2 of Definition 14 by Lemma 7. By Lemma 8 $\ell_F$ satisfied condition 3 in Definition 14. □

**Lemma 7.** *Consider $d_1$ over $X_1 \cup X_2$ and $d_2$ an $(\{X_1, X_2\}, d_1)$-outer-consistent distance function, then $(X_1, X_2, d_2) \not<_F (X_1, X_2, d_1)$ whenever $F$ is refinement-preserving and satisfies locality, outer-consistency, and outer richness.*

*Proof.* Assume that there exist such $d_1$ and $d_2$ where $(X_1, X_2, d_2) <_F (X_1, X_2, d_1)$. Let $d_3$ over $X_1 \cup X_2$ be a distance function such that $d_3$ is $(\{X_1, X_2\}, d_1)$-outer-consistent and $d_2$ is $(\{X_1, X_2\}, d_3)$-outer-consistent.

By outer richness, there exists a distance function $d^*$ that extends both $d_1$ and $d_3$ over $X^* = X_1 \cup X_2 \cup X'_1 \cup X'_2$ where $(X'_1 \cup X'_2, d_3) \sim (X_1 \cup X_2, d_3)$ and $F(X^*, d^*, 2) = \{X_1 \cup X_2, X'_1 \cup X'_2\}$.

Since $F(X_1 \cup X_2, d_1, 2) = \{X_1, X_2\}$, by locality and outer-consistency, $F(X^*, d^*, 3) = \{X_1 \cup X_2, X'_1, X'_2\}$ or $F(X^*, d^*, 3) = \{X'_1 \cup X'_2, X_1, X_2\}$. If $F(X^*, d^*, 3) = \{X_1 \cup X_2, X'_1, X'_2\}$, then by applying outer-consistency, we get that $(X_1, X_2, d_1) <_F (X_1, X_2, d_2)$, contradicting the assumption.

So $F(X^*, d^*, 3) = \{X'_1 \cup X'_2, X_1, X_2\}$. By outer richness, there exists a distance function $d^{**}$ that extends both $d_2$ and $d_3$ over $X^*$ where $(X'_1 \cup X'_2, d_3) \sim (X_1 \cup X_2, d_3)$ and $F(X^*, d^{**}, 2) = \{X_1 \cup X_2, X'_1 \cup X'_2\}$. As before, $F(X^*, d^{**}, 3) = \{X_1 \cup X_2, X'_1, X'_2\}$ or $F(X^*, d^{**}, 3) = \{X'_1 \cup X'_2, X_1, X_2\}$. If $F(X^*, d^{**}, 3) = \{X_1 \cup X_2, X'_1, X'_2\}$, then by applying outer-consistency on $F(X^*, d^*, 3)$, this contradicts that $F(X^*, d^*, 3) = \{X'_1 \cup X'_2, X_1, X_2\}$.

So, $F(X^*, d^{**}, 3) = \{X'_1 \cup X'_2, X_1, X_2\}$. By outer richness, there exists a distance function $d^{***}$ over $X^*$ that extends both $d_1$ and $d_2$ where $(X'_1 \cup X'_2, d_2) \sim (X_1 \cup X_2, d_2)$ and $F(X^*, d^{***}, 2) = \{X_1 \cup X_2, X'_1 \cup X'_2\}$. Since $(X_1, X_2, d_2) <_F (X_1, X_2, d_1)$, $F(X^*, d^{***}, 4) = \{X_1, X_2, X'_1, X'_2\}$. To obtain $F(X^*, d^{**}, 3)$, either $X_1$ and $X_2$ or $X'_1$ and $X'_2$ must be merged. If $X_1$ and $X_2$ are merged, then we contradict $(X_1, X_2, d_2) <_F (X_1, X_2, d_1)$, but if $X'_1$ and $X'_2$ are merged, then by outer-consistency we contradict $F(X^*, d^{**}, 3) = \{X'_1 \cup X'_2, X_1, X_2\}$. □

**Lemma 8.** *The function $\ell_F$, for any refinement-preserving function $F$ that satisfies locality, outer-consistency, and outer richness, satisfies condition 3 of Definition 14.*

*Proof.* Let $r$ be in the range of $\ell_F$. Then there exist data sets $(X_3, d_3)$ and $(X_4, d_4)$, $X_3 \cap X_4 = \emptyset$, and distance $d'$ over $X_3 \cup X_4$, such that $\ell_F(X_3, X_4, d') \geq r$. Let $(X_1, d_1)$, $(X_2, d_2)$ be a pair of data sets as defined above. If $\{X_1, X_2\} = \{X_3, X_4\}$ then we are done, so assume that $\{X_1, X_2\} \neq \{X_3, X_4\}$.

By outer richness, there exists a distance function $\hat{d}$ over $X = \bigcup X_i$ that extends $d_1, d_2, d_3, d_4$ such that $F(X, \hat{d}, 4) = \{X_1, X_2, X_3, X_4\}$. We define $\tilde{d}$ to be $(F(X, \hat{d}, 4), \hat{d})$-outer-consistent defined as follows:

$\tilde{d}(x, y) = \max\{\hat{d}(x, y), d'(x, y)\}$ when $x \in X_3, y \in X_4$ or $x \in X_4, y \in X_3$ and $\tilde{d}(x, y) = \hat{d}(x, y)$ otherwise.

Notice that $\tilde{d}|X_3 \cup X_4$ is $(F(X_3 \cup X_4, d', 2), d')$-outer-consistent. Thus, $\ell_F(X_3, X_4, \tilde{d}|X_3 \cup X_4) \geq r$.

Also by outer richness, there exists a distance function $\hat{d'}$ over $X$ that extends $d_1, d_2, \tilde{d}|X_3 \cup X_4$ such that $F(X, \hat{d'}, 3) = \{X_1, X_2, X_3 \cup X_4\}$. Using outer consistency, we can find $\tilde{d'}$ that is $(F(X, \tilde{d}, 4), \tilde{d})$-outer-consistent and $F(X, \hat{d'}, 3), \hat{d'})$-outer-consistent by just increasing distances between $X_i$ and $X_j$, where $i \neq j$ and $\{i, j\} \neq \{3, 4\}$. Thus, $F(X, \tilde{d'}, 4) = \{X_1, X_2, X_3, X_4\}$ and $F(X, \tilde{d'}, 3) = \{X_1, X_2, X_3 \cup X_4\}$. Therefore,

$$\ell_F(X_1, X_2, \tilde{d'}) > \ell_F(X_3, X_4, \tilde{d'}) \geq r.$$

□

**Lemma 9.** *Given a clustering function $F$ that is refinement-preserving and satisfies locality, outer-consistency, and outer richness, the linkage based clustering that $\ell_F$ defines agrees with $F$ on any input data set.*

*Proof.* For every $(X,d)$, the linkage based clustering that $\ell_F$ defines starts with the clusters consisting of all singletons, and at each step merges two clusters. Thus, for all $2 \leq k \leq |X|$, we have a $k$-clustering $C$ and the $k-1$ clustering merges some $C_1, C_2 \in C$, where $C_1 \cup C_2 = C$ or $\ell_F(C_1, C_2) < \ell_F(C_3, C_4)$, for all $C_3, C_4 \in C$, $\{C_3, C_4\} \neq \{C_1, C_2\}$. Therefore, for all $2 \leq k \leq |X|$, $(C_1, C_2, d|(C_1 \cup C_2)) <_F (C_3, C_4, d|(C_3 \cup C_4))$, for all $C_3, C_4$ as described, by our construction of $\ell_F$. Therefore, $F$ would merge the same clusters to obtain the $k-1$ clustering, and so $\ell_F$ agrees with $F$ for any input $(X,d)$ on all $k$-clusterings, $2 \leq k \leq |X|$. Clearly they also agree when $k = 1$. □

□

This concludes the proof of Lemma 14.

### 4.3.2 Every Linkage Based $k$-Clustering Function Satisfied the Properties

If a $k$-clustering function is linkage based, then by construction it is refinement-preserving.

**Lemma 10.** *Every linkage based $k$-clustering function is refinement-preserving.*

*Proof.* For every $1 \leq k' \leq k \leq |X|$, by definition of linkage based, $F(X,d,k)$ can be constructed from $F(X,d,k')$ by continually merging clusters until $k$ clusters remain. □

**Lemma 11.** *Every linkage based $k$-clustering function $F$ is local.*

*Proof.* Let $k'$-clustering $C$ be a subset of $F(X,d,k)$. Let $X' = \bigcup_{C_i \in C} C_i$.

We will show that for all $k' \leq i \leq |X'|$, $F(X', d|X', i)$ is a subset of $F(X,d,j)$ for some $j$. After, we conclude our proof using the following argument: $F(X', d|X', k')$ has $k'$ clusters, $F(X', d|X', k')$ is a subset of $F(X,d,j)$ for some $j$, and since between $F(X,d,j)$ and $F(X,d,k)$ in the algorithm we cannot merge clusters in $C$ (as $C$ would no longer be a subset of $F(X,d,k)$), this gives us that $F(X', d|X', k')$ is a subset of $F(X,d,k)$ and it is equal to $C$.

We prove the result by induction on $i = |X'| \ldots k'$. The base case follows from the observation that $F(X', d|X', |X'|)$ and $F(X,d,|X|)$ both consist of singleton clusters.

For some $i > k'$, assume that there exists a $j$ such that $F(X', d|X', i)$ is a subset of $F(X,d,j)$. We need to show that there exists a $j'$ such that $F(X', d|X', i-1)$ is a subset of $F(X,d,j')$.

Since $F$ is linkage based, there exists a linkage function $\ell$ so that when $\ell$ is used in the algorithm in Definition 15, the algorithm yields the same output as $F$.

Since $F(X', d|X', i) \subseteq F(X,d,j)$, and $C \subseteq F(X,d,k)$, there exists a $j^*$ so that $F(X,d,j^*)$ can be obtained from $F(X,d,j^*+1)$ by merging two clusters in $\subseteq X'$. The pair of clusters $\subseteq X'$ with minimal $\ell$ is the same as the pair of clusters with minimal $\ell$ value in $F(X', d|X', i)$. Therefore, $j' = j^*$. □

**Lemma 12.** *Every linkage based $k$-clustering function $F$ is outer-consistent.*

*Proof.* By the monotonicity condition in Definition 14, whenever two clusters are pulled further apart from each other, the corresponding $\ell$ value does not decrease. Consider some data set $(X, d)$ and $d'$ an $(F(X, d, k), d)$-outer-consistent distance function. We will show that $F(X, d, k) = F(X, d', k)$ by induction on $k$. Clearly, $F(X, d, |X|) = F(X, d', |X|)$. Assume that $F(X, d, j) = F(X, d', j)$ for some $j > k$. In order to obtain $F(X, d', j - 1)$, $F$ merges the pair of clusters $C'_1, C'_2 \in F(X, d', j)$ with minimal $\ell$ value. Similarly, to obtain $F(X, d, j - 1)$, $F$ merges the pair $C_1, C_2 \in F(X, d, j)$.

Suppose that $\{C_1, C_2\} \neq \{C'_1, C'_2\}$. Then $\ell(C'_1, C'_2, d) \leq \ell(C'_1, C'_2, d') < \ell(C_1, C_2, d') = \ell(C_1, C_2, d)$, where the first equality follows by monotonicity and the second inequality follows by the minimality of $\ell(C'_1, C'_2, d')$. Note that $C_1, C_2 \subseteq C_k$, where $C_k \in F(X, d, k)$. That is, $C_1$ and $C_2$ are part of the same cluster in $F(X, d, k)$, and since $d'$ is $(F(X, d, k), d)$-outer-consistent, the equality follows by representation-independence. But $\ell(C'_1, C'_2, d) < \ell(C_1, C_2, d)$ contradicts the minimality of $\ell(C_1, C_2, d)$, so $\{C_1, C_2\} = \{C'_1, C'_2\}$.

□

**Lemma 13.** *Every linkage based function is outer rich.*

*Proof.* Let $(X_1, d_1), (X_2, d_2), \ldots, (X_n, d_n)$ be some data sets. We will show that there exists an extension $d$ of $d_1, d_2, \ldots, d_n$ so that $F(\bigcup_{i=1}^{n} X_i, d, n) = \{X_1, X_2, \ldots, X_n\}$.

To make $F$ give this output, we design $d$ in such a way that for any $i$, and $A, B \subseteq X_i$, and any $C \subseteq X_i$, and $D \subseteq X_j$ where $i \neq j$, $\ell(A, B, d) < \ell(C, D, d)$.

Let $r = \max_{X_i, i\in\{1,2\}, A,B\subseteq X_i} \ell(A, B)$. Since $\ell$ satisfies property 4 of Definition 14, for any $C \subseteq X_i$, $D \subseteq X_j$, for $i \neq j$, there exists a distance function $d_{CD}$ that extends $d_i|C$ and $d_j|D$ so that $\ell(C, D) > r$. Consider constructing such distance function $d_{CD}$ for every pair $C \subseteq X_i$ and $D \subseteq X_j$, where $i \neq j$. Then, let $m = \max_{i\neq j, C\subseteq X_i, D\subseteq X_j} \max_{x\in C, y\in D} d_{CD}(x, y)$.

We define $d$ as follows: $d(x, y) = d_i(x, y)$ if $x, y \in X_i$ for some $i$ and $d(x, y) = m$ otherwise. Since $\ell$ satisfies property 2 of Definition 14, $\ell(C, D) > r$ for all $C \in X_i$, $D \in X_j$ where $i \neq j$. On the other hand, $\ell(A, B) \leq r$ for any $A, B \subseteq X_i$ for some $i$. Therefore, the algorithm will not merge any $C \subseteq X_i$ with $D \subseteq X_j$ where $i \neq j$, while there are any clusters $A, B \subseteq X_i$ for some $i$ remaining. This gives that $F(\bigcup_{i=1}^{n} X_i, d, n) = \{X_1, X_2, \ldots, X_n\}$. □

Finally, we put our results together to conclude the main theorem.

**Theorem 9 restated** *A $k$-clustering function is linkage based if and only if it is refinement-preserving and it satisfies: Outer Consistency, Locality and Outer Richness.*

*Proof.* By Lemma 2, if a $k$-clustering function is outer-consistent, refinement-preserving, and local, then it is linkage based. By Lemma 10, every linkage based $k$-clustering function is refinement-preserving. By and Lemma 11 every linkage based $k$-clustering function is local. By Lemma 12, every linkage based $k$-clustering function is outer-consistent. Finally, by Lemma 13, every linkage based function satisfies outer richness. □

## 4.4 Relaxations of a Linkage Function and Corresponding Characterizations

### 4.4.1 Simplified Linkage Function

Our proof also yields some insights about clustering that are defined by looser notions of linkage functions. We describe the characterization of the class of $k$-clustering functions that are based of linkage functions that are not required to obey the conditions of Definition 14.

**Definition 24** (Simplified linkage function)**.** *A* simplified linkage function $\ell$ *takes a data set* $(X, d)$ *and a partition* $(X_1, X_2)$ *of the domain* $X$ *and outputs a real number.*

We then define a *simplified linkage based function* as in Definition 15, but with a simplified linkage function instead of the linkage function in Definition 14. This leads to an interesting characterization of simplified linkage based functions that satisfy outer-consistency and outer richness.

**Theorem 10.** *A $k$-clustering function that satisfies outer-consistency and outer richness is simplified linkage based if and only if it is refinement-preserving and local.*

*Proof.* Since a linkage function is a simplified linkage function with additional constraints, by Lemma 2 we get that an outer-consistent, refinement-preserving and local $k$-clustering function is simplified linkage based. The results and proofs of Lemma 10 and Lemma 11 also apply for simplified linkage functions, thus showing that simplified linkage based functions are refinement-preserving and local. □

### 4.4.2 General Linkage Function

Unlike linkage based $k$-clustering functions defined in Definition 15 or simplified linkage based functions, a *general* linkage based $k$-clustering function might use a different linkage procedure on every data set.

This results from a modification on the definition of a linkage function, allowing the function to have access to the entire data set, outside the two clusters under comparison.

**Definition 25** (General linkage function)**.** *A* general linkage function*, given a data set* $(X, d)$ *and* $A, B \subseteq X$ *where* $A \cap B = \emptyset$*, outputs a real number.*

Note that in the above definition, $A$ and $B$ need not partition $X$. As such, the function may use information outside of both $A$ and $B$ to determine what value to assign to this pair of clusters. We define a *general linkage based $k$-clustering function* as in Definition 15, except using a general linkage function instead of the linkage function in definition 14.

**Definition 26** (general linkage based $k$-clustering function)**.** *A $k$-clustering function $F$ is* general linkage based *if there exists a general linkage function $\ell$ so that*

- $F(X, d, |X|) = \{\{x\} \mid x \in X\}$

- *For $1 \leq k < |X|$, $F(X,d,k)$ is constructed by merging the two clusters in $F(X,d,k+1)$ that minimize the value of $\ell$. Formally,*

$$F(X,d,k) = \{C_i \mid C_i \in F(X,d,k+1), C_i \neq C_1, C_i \neq C_2\} \cup \{C_1 \cup C_2\},$$

  *such that $\{C_1, C_2\} = argmin_{\{C_1,C_2\} \subseteq F(X,d,(k+1))} \ell((X,d), C_1, C_2)$.*

For example, a $k$-clustering function that uses single-linkage on data sets with an even number of points, and maximal linkage on data sets with an odd number of points, is not linkage based, but it is a general linkage based $k$-clustering function. Many other examples of general linkage based functions are artificial, and do not correspond to what is commonly thought of as linkage based clustering. Yet general linkage based functions include linkage based functions, and are actually easier to characterize. In addition, the Neighbour Joining algorithm, commonly applied in Phylogeny, is another example of a general linkage based $k$-clustering function.

**Theorem 11.** *A $k$-clustering function is refinement-preserving if and only if it is a general linkage based $k$-clustering function.*

*Proof.* For every $1 \leq k \leq k' \leq |X|$, by definition of a general linkage based $k$-clustering function, $F(X,d,k)$ can be constructed from $F(X,d,k')$ by continually merging clusters until $k$ clusters remain. Therefore, general linkage based functions are refinement-preserving.

Assume that $F$ is refinement-preserving. Then whenever $k' > k$, $F(X,d,k)$ can be obtained from $F(X,d,k')$ by merging clusters in $F(X,d,k')$. In particular, $F(X,d,k)$ can be obtained from $F(X,d,k+1)$ by merging a pair of clusters in $F(X,d,k+1)$. It remains to show that there exists a general linkage function $\ell$ that defines which clusters are merged.

We now show how to construct the general linkage function. For every $(X,d)$, and for every $k$, if $F(X,d,k)$ can be obtained from $F(X,d,k+1)$ by merging clusters $A$ and $B$, then set $\ell((X,d),(A,B)) = |X| - k$. For the remaining $A, B \subseteq X$, set $\ell((X,d)(A,B)) = |X|$.

Consider the function $F'$ resulting from using the general linkage function $\ell$ to determine which pair of clusters to merge, until $k$ clusters remain. Clearly, $F'(X,d,|X|) = F(X,d,|X|)$. Assume that $F'(X,d,k+1) = F(X,d,k+1)$. We show that $F'(X,d,k) = F(X,d,k)$. Since $F'$ is a general linkage based $k$-clustering function, it merges some clusters $C_1, C_2 \in F'(X,d,k+1)$ to obtain $F'(X,d,k)$. Since $F$ is refinement-preserving, it merges some clusters $C_3, C_4 \in F(X,d,k+1)$ to obtain $F(X,d,k)$, therefore $\ell((X,d)(C_3,C_4)) = |X| - k$. For any $\{C_5, C_6\} \in F(X,d,k)$ so that $\{C_5, C_6\} \neq \{C_3, C_4\}$, either $C_5$ and $C_6$ are merged to obtain $F(X,d,k')$ for some $k' < k$ and so $\ell((X,d)(C_5,C_6)) = |X| - k'$, or $C_5$ and $C_6$ are never merged directly (they are first merged with other clusters), and so $\ell((X,d)(C_5,C_6)) = |X|$. In either case, $\ell((X,d)(C_3,C_4)) < \ell((X,d)(C_5,C_6))$. Since $\ell$ defines $F'$, $F'$ merges $C_1, C_2 \in F'(X,d,k+1) = F(X,d,k+1)$ to obtain $F'(X,d,k)$. Therefore, $\{C_1, C_2\} = \{C_3, C_4\}$ and so $F'(X,d,k) = F(X,d,k)$. □

CHAPTER 5

# A CLASSIFICATION OF PARTITIONAL CLUSTERING METHODS

Our vision is that ultimately, there would be a sufficiently rich set of properties that would provide a detailed, property-based, taxonomy of clustering methods, that could, in turn, be used as guidelines for a wide variety of clustering applications. This chapter takes a step towards this goal by using natural properties to examine some popular clustering techniques.

In this chapter, we present a taxonomy for common deterministic $k$-clustering functions with respect to the properties that we propose. We also study relationships between properties, independent of any particular algorithm. We show positive relationships between some of the properties. In addition, we strengthen Kleinberg's impossibility result [35] using a relaxation of one of the properties that he proposed.

## 5.1 Properties of Clustering Functions

A key component in our approach are properties of $k$-clustering functions that address the input-output behavior of these functions. Several of the properties that we use in this chapter were defined in Chapter 4, namely, locality, consistency, outer-consistency, inner-consistency, refinement preserving, outer richness, and representation independence.

**Order invariance**: Order invariance, proposed by Jardine and Sibson [33], describes clustering functions that are based on the ordering of pairwise distances. That is, it matters when a distance between a pair of elements is smaller than or larger than another pairwise distance, but the precise values are not important. Formally, a distance function $d'$ of $X$ is an *order invariant modification* of $d$ over $X$ if for all $x_1, x_2, x_3, x_4 \in X$, $d(x_1, x_2) < d(x_3, x_4)$ if and only if $d'(x_1, x_2) < d'(x_3, x_4)$.

**Definition 27** (Order invariance)**.** *A $k$-clustering function $F$ is* order invariant *if whenever a distance function $d'$ over $X$ is an order invariant modification of $d$, $F(X, d, k) = F(X, d', k)$ for all $k$.*

**k-Richness**: The k-richness property requires that we be able to obtain any partition of the domain by modifying the distances between elements. This property is based on Kleinberg's [35] richness axiom, requiring that for any sets $X_1, X_2, \ldots, X_k$, there exists a distance function $d$ over $X' = \bigcup_{i=1}^{k} X_i$ so that $F(X', d) = \{X_1, X_2, \ldots, X_k\}$.

**Definition 28** (K-richness)**.** *A $k$-clustering function $F$ satisfies* k-richness *if for any disjoint sets $X_1, X_2, \ldots, X_k$, there exists a distance function $d$ over $X' = \bigcup_{i=1}^{k} X_i$ so that $F(X', d, k) = \{X_1, X_2, \ldots, X_k\}$.*

**Threshold-richness**: Fundamentally, the goal of clustering is to group points that are close to each other, and to separate points that are far apart. Axioms of clustering need to represent these objectives and no set of axioms of clustering can be complete without integrating such requirements. Consistency is the only previous property that aims to formalize these requirements. However, consistency is not satisfied by many common $k$-clustering functions.

**Definition 29** (Threshold richness)**.** *A $k$-clustering function $F$ is* threshold-rich *if for every clustering $C$ of $X$, there exist real numbers $a < b$ so that for every distance function $d$ over $X$ where $d(x, y) \leq a$ for all $x \sim_C y$, and $d(x, y) \geq b$ for all $x \not\sim_C y$, we have that $F(X, d, |C|) = C$.*

This property is based on Kleinberg's [35] $\Gamma$-forcing property, and is equivalent to the requirement that for every partition $\Gamma$, there exist $a < b$ so that $(a, b)$ is $\Gamma$-forcing.

**Inner richness**: Complementary to outer richness defined in Chapter 4, inner richness requires that there be a way of setting distances within sets, without modifying distances between the sets, so that $F$ outputs each set as a cluster. This corresponds to the intuition that between-cluster distances cannot eliminate any partition of $X$.

**Definition 30** (Inner richness)**.** *A $k$-clustering function $F$ satisfies* inner richness *if for every data set $(X, d)$ and clustering $C$ of $X$, there exists a $\hat{d}$ where for all $x \not\sim_C y$, $\hat{d}(x, y) = d(x, y)$, and $F(X, \hat{d}, k) = C$.*

## 5.2 Property-Based Classification of $k$-Clustering Functions

In this section we present a taxonomy of common $k$-clustering functions. The taxonomy is presented in Figure 5.1.

The taxonomy in Figure 5.1 illustrates how clustering algorithms differ from one another. For example, order-invariance and inner-consistency can be used to distinguish among the three common linkage-based algorithms. Min-sum differs from $k$-means and $k$-medoids in that it satisfies inner-consistency. Unlike all the other algorithms discussed, the similarity-based clustering functions are not local. Note also the same results hold for all distance-based measures if the triangle inequality is required.

### 5.2.1 Properties that could be used as axioms

In Figure 5.1, we show which properties are satisfied by some common clustering methods. But could any of these properties be clustering axioms?

| *Function* | outer consistent | inner consistent | local | refinement-preferring | order invariant | k-rich | outer rich | inner rich | threshold rich | scale invariant | rep. independence |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Single Linkage | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Average Linkage | ✓ | $X$ | ✓ | ✓ | $X$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Complete Linkage | ✓ | $X$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $k$-medoids | ✓ | $X$ | ✓ | $X$ | $X$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $k$-means | ✓ | $X$ | ✓ | $X$ | $X$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Min sum | ✓ | ✓ | ✓ | $X$ | $X$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Ratio cut | $X$ | ✓ | $X$ | $X$ | $X$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Normalized cut | $X$ | $X$ | $X$ | $X$ | $X$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Figure 5.1: A taxonomy of $k$-clustering functions, illustrating what properties are satisfied by some common $k$-clustering functions.

First, let's consider what a set of axioms for clustering should satisfy. Usually, when a set of axioms is proposed for some semantic notion (or a class of objects, say clustering functions), the aim is to have both soundness and completeness. Soundness means that every element of the described class satisfies all axioms (so, in particular, soundness implies consistency of the axioms), and completeness means that every property shared by all objects of the class is implied by the axioms. Intuitively, ignoring logic subtleties, a set of axioms is complete for a class of objects if any element outside that class fails at least one of these axioms.

In our context, there is a major difficulty - there exist no semantic definition of what clustering is. We wish to use the axioms as a definition of clustering functions, but then what is the meaning of soundness and completeness? We have to settle for less. While we do not have a clear definition of what is clustering and what is not, we do have some examples of functions that should be considered clustering functions, and we can come up with some examples of partitionings that are clearly not worthy of being called "clusterings". We replace soundness by the requirement that all of our axioms are satisfied by all these examples of common clustering functions (relaxed soundness), and we want that partitioning functions that are clearly not reasonable clustering functions fail at least one of our axioms (relaxed completeness).

Our taxonomy reveals that some intuitive properties, which may have been expected of all $k$-clustering functions, are not satisfied by some common $k$-clustering functions, and so soundness is failed. For example, locality is not satisfied by the similarity-based clustering functions ratio-cut and normalized-cut. Also, most functions fail inner consistency, and therefore do not satisfy consistency, even though the latter was previously proposed as an axiom of clustering functions [35].

On the other hand, representation independence, scale invariance, and all richness properties (in the setting where the number of clusters, $k$, is a part of the input), are satisfied by all the

$k$-clustering functions considered. It seems that representation independence and scale-invariance make for natural axioms. Threshold richness is the only one that is both satisfied by all $k$-clustering functions considered, and reflects the main objective of clustering: to group points that are close together and to separate points that are far apart.

Threshold richness directly implies k-richness. In Section 5.3, we show that when threshold richness is combined with scale invariance, it also implies outer-richness and inner-richness. Therefore, scale-invariance, representation independence, and threshold richness are sound and as such, are candidate clustering axioms.

However, we emphasize that the set of axioms consisting of these three properties fails relaxed completeness. These three properties do not make a complete set of axioms for clustering, since some functions that satisfy all three properties do not make reasonable $k$-clustering functions; a function that satisfies representation independence and scale invariance can also satisfy threshold richness by behaving reasonably only when there are clusters that are very well separated, while producing poor partitions of other data.

Therefore, we are not proposing here a complete set of axioms of clustering. Instead, we identified three properties that are both natural, and are satisfied by all clustering functions that we analysed. It is therefore possible that these properties combined with some other properties may yield a complete set of axioms of clustering.

### 5.2.2 Taxonomy Proofs

We now prove the results presented in Table 5.1

We say that a clustering function $F$ *depends only on within-cluster distances* if there exists a function $g : (X, d) \to \mathbb{R}^+$ so that for any data set $(X, d)$ and $1 \leq k \leq |X|$, $F(X, d, k) = \min_{C \text{ of } X} \sum_{C_i \in C} g(C_i, d|C_i)$. Note that $k$-means, $k$-median, min-sum, and all centroid-based clustering functions depend only on within-clusters distances.

**Theorem 12.** *If a clustering function $F$ depends only on within-cluster distances, then it is local.*

*Proof.* Consider any data set $X$ and $1 \leq k' \leq k \leq |X|$. Let $C = F(X, d, k)$. Let $k'$-clustering $C'$ be a subset of $C$. Let $S \subseteq X$ be the union of all the clusters in $C'$. Assume by way of contradiction that there exists a clustering $C''$ of $S$ with lower loss than $C'$.

Since $F$ depends only on within-cluster distances, we can obtain a $k$-clustering of $X$ with lower loss than $C$ by clustering $X \cap S$ using $C''$ instead of $C'$. Since $F(X, d, k)$ has minimal loss over all $k$-clusterings of $X$, this is a contradiction. □

In the above definition of the function $g$, we could require the following natural monotonicity property: Given any pair of distance functions $d$ and $d'$ over a domain set $X$ where $d'(x, y) \geq d(x, y)$ for all $x, y \in X$, we have that $g(X, d') \geq g(X, d)$. This means that we cannot decrease the cost of a cluster by increasing some pairwise distances within it. If $F(X, d, k) = \min_{C \text{ of } X} \sum_{C_i \in C} g(C_i, d|C_i)$ for a monotone function $g$, then we say that $F$ *depends only on within-cluster distances monotonically.* Note that $k$-means, $k$-median, min-sum, and all centroid-based clustering functions depend only on within-clusters distances monotonically.

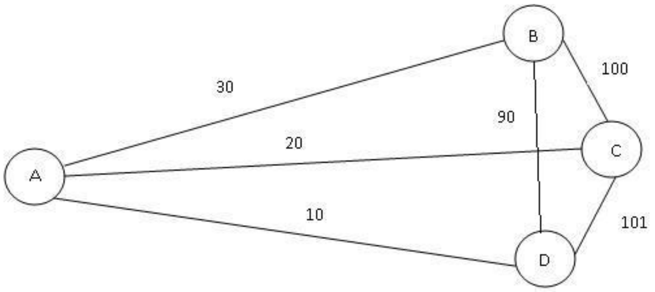


Figure 5.2: A data set used to illustrate that Ratio-Cut does not satisfy locality.

**Theorem 13.** *If a clustering function $F$ depends only on within-cluster distances monotonically, then it is outer-consistent.*

*Proof.* By way of contradiction, assume that there exists a data set $(X, d)$, $k \in Z^+$ and $d'$ a $(F(X, d, k), d)$-outer-consistent distance function, so that $F(X, d, k) \neq F(X, d', k)$. Let $C = F(X, d, k)$ and $C' = F(X, d', k)$. Since $d'$ is $(C, d)$-outer-consistent and $F$ depends only on within-cluster distances, $C$ has the same loss on $(X, d)$ and $(X, d')$. As $F(X, d', k) \neq C$, $C'$ has lower loss than $C$ on $(X, d')$. Now consider clustering $(X, d)$ with $C'$. Since $d'$ is $(C, d)$-outer-consistent, for all $x, y \in X$, $d(x, y) \leq d'(x, y)$. Since $F$ depend only on within-cluster distances monotonically, this implies that the cost of every cluster in $C'$ on $(X, d)$ is no larger than the cost of that cluster in $(X, d')$. It follows that the loss of $C'$ on $(X, d)$ is at most the loss of $C'$ on $(X, d')$. However, since $C = F(X, d)$, the minimal loss clustering on $(X, d)$ is $C$, giving a contradiction. □

Kleinberg showed that centroid-based clustering functions are not consistent (Theorem 4.1, [35]). Indeed, his proof shows that centroid-based clustering functions are not inner-consistent. The same argument also shows that $k$-means is not inner-consistent.

**Theorem 14.** *Ratio-Cut is not local.*

*Proof.* Figure 5.2 illustrates a data set (with the similarity indicated on the edges) where the optimal ratio-cut 3-clustering is $\{\{A\}, \{B, C\}, \{D\}\}$. However, on data set $\{B, C, D\}$ (with the same pairwise similarities as in Figure 5.2), the clustering $\{\{B\}, \{C, D\}\}$ has lower ratio-cut than $\{\{B, C\}, \{D\}\}$. □

**Theorem 15.** *Normalized-Cut is not local.*

*Proof.* Figure 5.3 illustrates a data set with the similarities indicated on the arrows - a missing arrow indicates a similarity of 0. The optimal normalized-cut 3-clustering is $\{\{A, A'\}, \{B, B', C, C'\}, \{D, D'\}\}$. However, on data set $\{B, B', C, C', D, D'\}$ (with the same pairwise similarity as in Figure 5.3), the clustering $\{\{B, B'\}, \{C, C', D, D'\}\}$ has lower normalized cut than $\{\{B, B', C, C'\}, \{D, D'\}\}$. □

We now prove that inner consistency distinguishes between ratio cut and normalized cut.

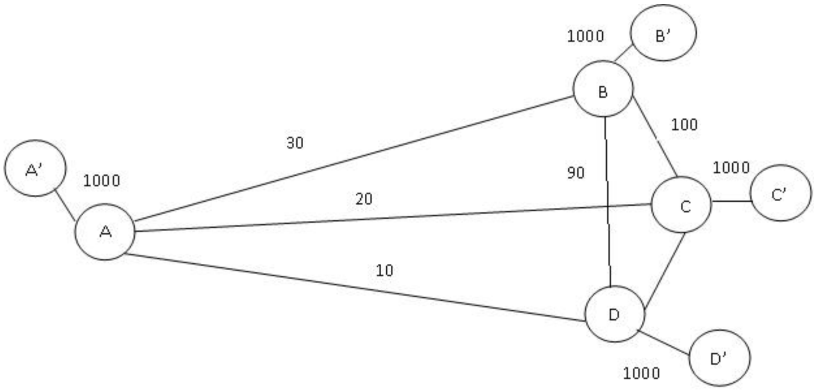


Figure 5.3: A data set used to illustrate that Normalized-Cut does not satisfy locality.

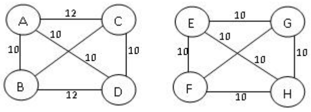


Figure 5.4: A data set used to illustrate that normalized cut does not satisfy inner-consistency. The similarities not marked are set to 0.

**Theorem 16.** *Ratio-cut is inner-consistent.*

*Proof.* Let $F$ denote the Ratio-cut clustering function. Assume by way of contradiction that ratio-cut is not inner-consistent. Then there exist some $(X, s)$, $k$, and $s'$ an $(F(X, s, k), s)$-inner-consistent distance function so that $F(X, s', k) \neq F(X, s, k)$. Let $C = F(X, s, k)$ and $C' = F(X, s, k)$.

Then $RatioCut(C', (X, s')) < RatioCut(C, (X, s))$.

Now consider clustering $C'$ on $(X, s)$. The ratio-cut of $C'$ on $(X, s)$ is at most the ratio-cut of $C'$ on $(X, s')$ since going from $s'$ to $s$ can only decrease similarities, which can only decrease the ratio-cut. That is, $RatioCut(C', (X, s)) \leq RatioCut(C', (X, s'))$. Therefore, $RatioCut(C', (X, s)) \leq RatioCut(C', (X, s')) < RatioCut(C, (X, s))$, which contradicts that $F(X, s, k) = C \neq C'$. □

**Theorem 17.** *Normalized-cut is not inner-consistent.*

*Proof.* Let $F$ denote the Normalized-cut clustering function. Consider the data set $(X, d)$ in Figure 5.4. For $k = 3$, $F(X, d, 3) = \{\{A, C\}, \{B, D\}, \{E, F, G, H\}\}$. Define a distance function $d'$ over $X$ so that $d'(E, F) = d'(G, H) = 100$, and $d'(x, y) = d(x, y)$ for all $\{x, y\} \neq \{E, F\}$, $\{x, y\} \neq \{G, H\}$. Then $d'$ is a $(F(X, d, 3), d), d)$-inner consistent change, however, $F(X, d', 3) = \{\{A, B, C, D\}\}$. But then $F(X, d', 3) \neq F(X, d, 3)$, violating inner-consistency. □

**Lemma 14.** *Ratio cut satisfies inner-richness.*

*Proof.* Consider any data set $(X, s)$ and partitioning $\{C_1, C_2, \ldots, C_n\}$ of $s$. Let $m = \max_{i \neq j, a \in C_i, b \in C_j} s(a, b)$. Construct $s'$ as follows: for all $i \neq j, a \in C_i, b \in C_j$, set $s'(a, b) = s(a, b)$. Otherwise, set $s(a, b) = m|X|^3 + 1$. The ratio cut loss of $\{C_1, C_2, \ldots, C_n\}$ on $(X, s')$ is less than $m|X|^2$, and any other $n$-clustering of $(X, s')$ has loss greater than $m|X|^2$. □

**Lemma 15.** *Normalized cut satisfies inner-richness.*

*Proof.* We can modify the within edges to make the normalized cut of the clustering $\{C_1, C_2, \ldots, C_k\}$ arbitrarily close to 0, making all within-cluster edges equal. The cost of any other clustering would have an edge $(x, y)$ so that $x, y \in C_i$ for some $i$, and so the cost of any such clustering is arbitrarily greater than the cost of $\{C_1, C_2, \ldots, C_k\}$ (in particular, great than $1/m$ where $m$ is the number of edges). □

**Lemma 16.** *Average linkage and complete linkage are not inner consistent.*

*Proof.* We present here a counter example for both. Let $X = \{A, B, C, D\}$ and define distance $d$ over $X$ as follows: $d(A, B) = 1+\epsilon, d(A, C) = 1-3.5\epsilon, d(A, D) = 1, d(B, C) = 1-4\epsilon, d(B, D) = 1-\epsilon$ and $d(C, D) = 1-2\epsilon$.

For sufficiently small epsilon, all individual lengths are approximately 1, but the sum of any path between two points in $X$ is approximately 2 or more. For both average and complete linkage, $B$ and $C$ are merged first, followed by $(B, C)$ and $D$. If we make an inner consistent change, and set $d(B, D) = 1-5\epsilon$, then $B$ and $D$ are merged first, followed by $A$ and $C$. □

**Lemma 17.** *Min-sum is inner consistent.*

*Proof.* Given a data set $(X, d)$, minsum yields a clustering $C^*$ of $X$. Assume, by means of contradiction, that shrinking some within cluster edges yields a different clustering as the output to minsum, and denote this clustering by $C'$. Let the sum of all differences over the edges we shrunk be denoted by $\alpha$, and the new distance function be denoted by $d'$. Define $cost(C, d) = \sum_{x \sim_C y} d(x, y)$. The difference between $cost(C', d')$ and $cost(C', d)$ is at most $\alpha$. So, $cost(C', d') \geq cost(C', d) - \alpha > cost(C^*, d) - \alpha = cost(C^*, d')$, since $C^*$ had the minimum cost with distance function $d$. □

**Lemma 18.** *Normalized cut and ratio cut are not outer consistent.*

*Proof.* We present a simple counter example. Let $X = \{a, b, c, d\}$ and define similarity function $d$ over $X$ as follows: $d(a, b) = 1, d(a, d) = 0.999, d(b, c) = 1.0015, d(c, d) = 1.001, d(a, c) = 0$ and $d(b, d) = 0$. With this arrangement, using ratio cut we arrive at the 3-clustering $a, d, \{b, c\}$. If we change the similarity between $a$ and $b$ to 0.997, which is an outer consistent change because we are dealing with similarities, then we arrive at the 3-clustering $a, b, \{c, d\}$. Therefore, ratio cut is not outer consistent. The same example works for normalized cut, except that we create points $x_a, x_b, x_c, x_d$ such that $d(x_i, i) = 100$ and the similarity between $x_i$ and every other point is 0. □

The linkage-based algorithms single-linkage, average-linkage, and complete linkage are local, outer-consistent, outer-rich, and refinement-preserving as show in Chapter 4. Single linkage is inner-consistent since by Kleinberg's Theorem 2.2(a) single-linkage is consistent. Refinement-preserving is a property specific to linkage-based algorithms, and it is easy to see that the remaining methods do not satisfy it.

Single linkage and complete linkage are order invariant since the algorithms make use only of relative distances according to the less-than relation. All other clustering functions that we classify make use of the exact values in the distance function, and it can be shown that those functions are not order invariant by demonstrating data sets with order invariant modifications of those data sets on which the output of the clustering functions differ.

Threshold richness for all clustering functions is achieved by making the ratio of the maximum between edges and minimum within edges sufficiently large. It follows immediately that these methods also satisfy $k$-richness. By Theorem 18 and Theorem 19 it also follows that the clustering functions satisfy inner-richness and outer-richness.

## 5.3 Relationships Between Properties

We now present several relationships between the properties discussed above. These relationships help in the analysis of clustering algorithms, in addition to providing a better general understanding of the properties themselves. Many properties are independent, as shown in our Taxonomy.

### 5.3.1 Relationships Between Richness Properties

K-richness is the weakest of the richness properties, implied by all the other richness variants. For functions that satisfy scale-invariance, threshold-richness implies outer-richness.

**Theorem 18.** *If a clustering function $F$ is scale-invariant and threshold-rich then it is also outer-rich.*

*Proof.* Consider any data sets $(X_1, d_1), \ldots, (X_k, d_k)$. Let $X = \cup_{i=1}^{k} X_i$ and $C = \{X_1, \ldots, X_k\}$. Scale $(X_i, d_i)$, for every $1 \leq i \leq k$ by the same positive scalar $c$ so that the longest edge over all the $d_i$'s is less than $a$ (from the definition of threshold richness). Formally, let $m = \max_{i \neq j, x \in X_i, y \in X_j} d(x, y)$. If $m = 0$, then $C$ consists of $k$ singletons and so $F(X, d, k) = C$ for any $d$. Otherwise, for every $1 \leq i \leq k$, let $d_i'$ be such that $d_i'(x, y) = \frac{a}{m} d(x, y)$. Then, let $d^*$ over $X$ be a distance function that extends $X_i'$ for every $1 \leq i \leq k$, and for every $x \in X_i$, $y \in X_j$ where $i \neq j$, $d^*(x, y) \geq b$ (for concreteness, we can set $d^*(x, y) = b$). Since $F$ is threshold-rich, $F(X, d^*, k) = C$. Now, let $d^{**}$ be a distance function over $X$ so that, for all $x, y \in X$, $d^{**}(x, y) = \frac{m}{a} d^*(x, y)$. Then $d^{**}$ extends $X_i$ for every $1 \leq i \leq k$. Since $F$ is scale-invariant, $F(X, d^{**}, k) = F(X, d^*, k) = C$. □

Similarly, we show that scale invariant functions that are threshold-rich also satisfy inner-richness.

**Theorem 19.** *If a clustering function $F$ is scale-invariant and threshold-rich then it is also inner-rich.*

*Proof.* Given any $(X, d)$, any partition $C$ of $X$, and any $d$ over $X$, we need to show that there exists a $\hat{d}$ where for all $x \not\sim_C y$, $\hat{d}(x, y) = d(x, y)$, and $F(X, \hat{d}, k) = C$. To do so, we create $d'$ by setting all within-cluster distances in $C$ to $a$ from the definition from threshold richness, and for $x \not\sim_C y$, set $d'(x,y) = c \cdot d(x,y) > b$, where $b$ is from the definition of threshold richness and $c$ is a large enough constant so that $cd(x,y) > b$ for all $x \not\sim_C y$. Since $F$ satisfies threshold richness, it follows that $F(X,d') = C$. Next, scale $d'$ by $\frac{1}{c}$ to obtain $\hat{d}$. Since $F$ is scale invariant, $F(X,\hat{d}) = C$, and for all $x \not\sim_C y$, $\hat{d}(x,y) = d(x,y)$ by construction. □

On the other hand, outer-richness (even with scale-invariance) does not imply threshold-richness. However, for consistent clustering functions, K-richness implies threshold-richness, and therefore outer-richness implies threshold-richness.

**Theorem 20.** *If a clustering function $F$ is rich and consistent, then it is also threshold-rich.*

*Proof.* Let $C = \{X_1, \ldots, X_k\}$ be some clustering of data set $X = \cup_{i=1}^{k} X_i$. Since $F$ is outer-rich, there exists a distance function $d$ over $X$ that extends $X_i$ for every $1 \leq i \leq k$, where $F(X,d,k) = C$. Let $d'$ be a $(C,d)$-outer consistent change so that $\max_{x \sim_C y} d(x,y) = a < \min_{x \not\sim_C} y = b$. Since $F$ is consistent, $F(X,d') = C$. Let $d^*$ be any distance function over $X$ where for all $x \sim_C y$, $d^*(x,y) \leq a$ for all $x \not\sim_C y$, $d^*(x,y) \geq b$. Then $d^*$ is a consistent change of $d'$, and since $F$ is consistent, $F(d^*, X, k) = C$. □

### 5.3.2 Relationships Between Consistency and Richness Properties

**Lemma 19.** *If a clustering function $F$ is scale-invariant, outer-rich and outer-consistent then it is also inner-rich.*

*Proof.* Consider any data set $(X,d)$ and partition $\{X_1, X_2, \ldots, X_n\}$ of $X$. Let $d_i = d|X_i$. Since $F$ is satisfies outer-richness, there exists a distance function $d'$ that extends $d_1, d_2, \ldots, d_n$ and $F(X,d',n) = \{X_1, X_2, \ldots, X_n\}$. Let $m = \max_{i \neq j, a \in X_i, b \in X_j} \frac{d'(a,b)}{d(a,b)}$.

We now construct a distance function $\hat{d}$ that is $(F(X,d',n), d')$-outer-consistent such that $\hat{d}(a,b) = m \cdot d(a,b) \geq d'(a,b)$, for all $a \in X_i, b \in X_j, i \neq j$. This is possible because $d'(a,b) \leq m \cdot d(a,b)$ by our definition of $m$. Therefore, $F(X,\hat{d},n) = F(X,d',n)$. Now, by applying scale invariance we construct $\tilde{d}$ such that $\tilde{d}(a,b) = \frac{\hat{d}(a,b)}{m}$ and we have $F(X,\tilde{d},n) = F(X,d',n)$ and for all $a \in X_i, b \in X_j, i \neq j$ we have $\tilde{d}(a,b) = d(a,b)$.

□

**Lemma 20.** *If a function $F$ is scale-invariant, inner-rich, and inner-consistent then it is also outer-rich.*

*Proof.* Consider any $(X_1, d_1), \ldots, (X_n, d_k)$. By inner richness, there exists some distance function $d$ so that $F(X,d) = \{X_1, \ldots, X_k\}$, although $d$ does not necessarily extend any of the $d_i$s.

Let $m$ be the length of the minimum within-cluster distance in $F(X,d)$. Construct $d'$ by shrinking all within-cluster distances to be smaller than $m$, so that for all $1 \leq i \leq k$, $d'|X_i = c \cdot d_i$ for some constant $c$. Then $F(d, \frac{1}{c} \cdot d') = \{X_1, \ldots, X_k\}$ by inner consistency and scale-invariance, and for all $1 \leq i \leq k$, $d'$ extends $d_i$. □

**Corollary 1.** *A consistent and scale-invariant clustering function $F$ is outer-rich if and only if it is inner-rich.*

## 5.4 Impossibility Results

We strengthen Kleinberg's famous impossibility result [35] for clustering functions (which do not take the number of clusters as part of its input), yielding a substantially simpler proof of the original result.

Kleinberg impossibility theorem (Theorem 2.1, [35]) was that no clustering function can simultaneously satisfy scale-invariance, richness, and consistency. As shown in Chapter 3, consistency has some counter-intuitive consequences. In Section 5.1, we showed that many natural clustering functions fail inner consistency[1], which implies that there are many general clustering functions that fail consistency.

On the other hand, many natural algorithms satisfy outer consistency. We strengthen Kleinberg's impossibility result by relaxing consistency to outer-consistency.

**Theorem 21.** *No clustering function can simultaneously satisfy outer-consistency, scale-invariance, and richness.*

*Proof.* Let $F$ be any clustering function that satisfies outer-consistency, scale-invariance and richness.

Let $X$ be some domain set with two or more elements. By richness, there exist distance functions $d_1$ and $d_2$ such that $F(X, d_1)$ is the clustering where every domain point is a cluster on its own and $F(X, d_2)$ is some different clustering, $C = \{C_1, \ldots, C_k\}$ of $X$.

Let $r = \max\{d_1(x, y) : x, y \in X\}$ and let $c$ be such that for every $x \neq y$, $cd_2(x, y) \geq r$. Define $\hat{d}(x, y) = c \cdot d_2(x, y)$, for every $x, y \in X$. Note that $\hat{d}(x, y) \geq d_1(x, y)$ for all $x, y \in X$. By outer-consistency, $F(X, \hat{d}) = F(X, d_1)$. However, by scale-invariance $F(X, \hat{d}) = F(X, d_2)$. This is a contradiction since $F(X, d_1)$ and $F(X, d_2)$ are different clusterings. □

A similar result is obtained with inner-consistency replacing outer consistency. Namely,

**Lemma 21.** *No clustering function can simultaneously satisfy inner-consistency, scale-invariance, and richness.*

*Proof.* Let $F$ be any clustering function that satisfies inner-consistency, scale-invariance and richness.

Let $X$ be some domain set with two or more elements. By richness, there exist distance functions $d_1$ and $d_2$ such that $F(X, d_1)$ is the clustering that puts all elements in the same cluster and $F(X, d_2)$ is some different clustering of $X$.

Let $r = \min\{d_1(x, y) : x, y \in X\}$ and let $c$ be such that for every $x \neq y$, $c \cdot d_2(x, y) \leq r$. Define $\hat{d}(x, y) = c \cdot d_2(x, y)$, for every $x, y \in X$. Then by scale-invariance, $F(X, \hat{d}) = F(X, d_2)$. But by inner-consistency, $F(X, \hat{d}) = F(X, d_1) \neq F(X, d_2)$. □

[1] Note that a $k$-clustering function and it's corresponding clustering function satisfy the same set of consistency properties.

Since consistency implies both outer-consistency and inner-consistency, Kleinberg's original result follows from Theorem 21.

Kleinberg's impossibility result illustrates property trade-offs for general clustering functions. The good news is that these results do not apply when the number of clusters is part of the input, as is illustrated in our taxonomy; single linkage satisfies scale-invariance, consistency and richness.

CHAPTER 6

# CLUSTERING OLIGARCHIES

Can the output of an algorithm be radically altered by the addition of a small, possibly adversarial, set of points? We use the term *oligarchies* to describe such sets of "influential" points. At first glance, it appears that all clustering methods are susceptible to oligarchies. Even $k$-means can substantially change its output upon the addition of a small set; if a data set has multiple structurally distinct solutions with near-optimal loss, then even a single point can radically alter the resulting partition. However, a more interesting picture emerges when considering how algorithms behave on well-clusterable data[1].

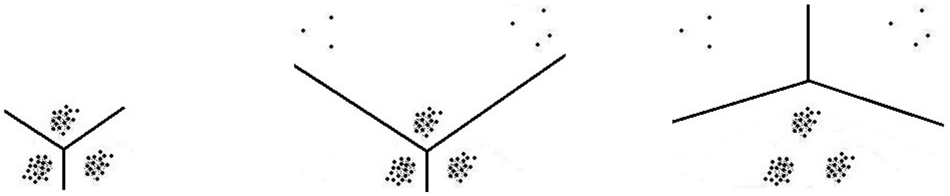
Examining their behavior on data that is well-clusterable, we find that some clustering methods exhibit a high degree of robustness to oligarchies; even small sets chosen in an adversarial manner have very limited influence on the output of these algorithms. These methods include $k$-means, $k$-medians, and $k$-medoids, as well the popular Lloyd's method with random center initialization. We perform a quantitative analysis of these techniques, showing precisely how clusterability affects their robustness to small sets additions. Our results demonstrate that the more clusterable a data set, the greater its robustness to the influence of potential oligarchies.

Other well-known methods admit oligarchies even on data that is highly clusterable. We prove that common linkage-based algorithms, including the popular average-linkage, exhibit this behavior. Several well-known objective-function-based methods, as well Lloyd's method initialized with pairwise distant centers, also fall within this category. More generality, we prove that all methods that detect clusterings satisfying a natural separability criteria, admit oligarchies even when the original data is well-clusterable.

Given the same well-clusterable input, algorithms that admit oligarchies can produce very different outputs from algorithms that prohibit them. For example, consider the data set displayed in Figure 6.1(a) and set the number of clusters, $k$, to 3. All algorithms that we considered, both those that admit and those that prohibit oligarchies, cluster this data as shown in Figure 6.1(a).

[1] Notice that the behavior of a clustering algorithm is often less important to the user when data is inherently un-clusterable.

As illustrated in Figure 6.1(b), when a small number of points is added, algorithms that prohibit oligarchies (eg. $k$-means) partition the original data in the same way as they did before the small set was introduced. In contrast, algorithms that admit oligarchies (eg. average-linkage) yield a radically different partition of the original data after the small set is added, as shown in Figure 6.1(c).



(a) A clustering produced by all clustering methods considered here

(b) A clustering produced by methods that prohibit oligarchies after a small number of points is added.

(c) A clustering produced by methods that admit oligarchies after the same small set is added.

Figure 6.1: An illustration of the contrasting input-output behaviour of algorithms that prohibit oligarchies with those that admit them.

For some clustering applications, algorithms that prohibit oligarchies are preferred. This occurs, for example, when some of the data may be faulty. This may be the case in fields such as cognitive science and psychology, when analyzing subject-reported data. In such cases, an algorithm that is heavily influenced by a small number of elements is inappropriate since the resulting clustering may be an artifact of faulty data. Algorithms that prohibit oligarchies may also be preferable when the data is entirely reliable, but clusters are expected to be roughly balanced (in terms of the number of points). Consider, for example, the use of clustering for identifying marketing target groups. Since target groups are typically large, no small set of individuals should have radical influence on how the data is partitioned.

However, there are applications that call for algorithms that admit oligarchies. Consider the task of positioning a predetermined number of fire stations within a new district. To ensure that the stations can quickly reach all households in the district, we may require that the maximum distance of any household to a station be minimized. If follows that a small number of houses can have a significant effect on the resulting clustering.

The chapter is organized as follows. We begin with a summary of related previous work followed by an introduction of our formal framework. In Section 6.2, we present a summary of our main results, contrasting the manner in which different algorithms treat oligarchies. In Section 6.3 and Section 6.4 we provide a quantitative analysis of the extent to which some popular clustering methods are robust to potential oligarchies.

## 6.1 Definitions

The *diameter* of a set $(X, d)$ is $\max_{x,y \in X} d(x, y)$. Throughout this chapter, we assume the diameter of a set is at most 1. The diameter of a clustering $C$ is the maximal diameter of a cluster in $C$.

The *Hamming distance* between clusterings $C$ and $C'$ of the same set $X$ is defined by

$$\Delta(C_1, C_2) = |\{\{x, y\} \subset X \mid (x \sim_C y) \oplus (x \sim_{C'} y)\}| / \binom{|X|}{2},$$

where $\oplus$ denotes the logical XOR operation. For sets $X, Z$ such that $X \subseteq Z$ and a clustering $C$ of $Z$, $C|X$ denotes the restriction of $C$ to $X$, thus if $C = \{C_1, \dots, C_k\}$, then $C|X = \{C_1 \cap X, \dots, C_k \cap X\}$.

Algorithms typically accept the number of desired clusters as a parameter. In that case we denote the output clustering by $F(X, k)$. $k$ is sometimes omitted when it is clear from context.

In this chapter we consider the robustness of sets to a small number of points. This is quantified by the following definition. Consider a data set $X$ and a (typically large) subset $Y$, where the set $O = X \setminus Y$ is a potential oligarchy. The set $Y$ is robust to the potential oligarchy $O$ relative to a clustering function, if $Y$ is clustered similarly with and without the points in $O$.

**Definition 31** ($\delta$-Robust)**.** *Given data sets $X$, $Y$ and $O$, where $X = Y \cup O$ and $Y \cap O = \emptyset$ $Y$, is* $\delta$-robust *to $O$ with respect to a clustering function $F$, if*

$$\Delta(F(Y), F(X)|Y) \leq \delta.$$

When the algorithm requires the number of clusters $k$ as part of the input, we say that $Y$ is $\delta$-robust to $O$ with respect to a clustering function $F$ and $k$, if $\Delta(F(Y, k), F(X, k)|Y) \leq \delta$. When $k$ is clear from context, we write that $Y$ is $\delta$-robust to $O$ with respect to a clustering function $F$.

A small $\delta$ indicates a robust subset, meaning that the data within that subset determines how it is clustered (to a large extent). For example, if $\delta = 0$, then how the subset is clustered is entirely determined by the data within that subset. On the other hand, large values of $\delta$ represent a subset that is volatile to oligarchy $O$, where data outside of this subset have substantial influence on how data within this subset are partitioned. Note that $\delta$ ranges between 0 and 1.

For a randomized algorithm $F$ we define probabilistic robustness as follows:

**Definition 32** (Probabilistically $\delta$-Robust)**.** *Let $F$ be a randomized clustering function. Given data sets $X$, $Y$, and $O$ where $X = Y \cup O$ and $Y \cap O = \emptyset$ $Y$, $Y$ is* $\delta$-robust *to $O$ with respect to $F$ with probability $1 - \epsilon$, if with probability $1 - \epsilon$ over the random choices of $F$,*

$$\Delta(F(Y), F(X)|Y) \leq \delta.$$

As our results will show, the robustness of a dataset is affected by whether it is well-clusterable, as captured in the following definition, based on a notion by Epter et al. [25].

**Definition 33** ($\alpha$-Separable)**.** *A clustering $C$ of $X$ is* $\alpha$-separable *for $\alpha \geq 0$ if for any $x_1, x_2, x_3, x_4 \in X$ such that $x_1 \sim_C x_2$ and $x_3 \nsim_C x_4$, $\alpha d(x_1, x_2) < d(x_3, x_4)$.*

If a data set contains an $\alpha$-separable clustering for some large $\alpha$ (such as $\alpha \geq 1$), then it is well-clusterable.

We define a balanced clustering based on the balance of cluster cardinalities.

**Definition 34** ($\beta$-Balanced)**.** *A clustering $C = \{C_1, \dots, C_k\}$ of $X$ is* $\beta$-balanced *if $|C_i| \leq \beta |X|$ for all $1 \leq i \leq k$.*

Note that $\frac{1}{k} \leq \beta \leq 1$ and that $\beta = \frac{1}{k}$ for a perfectly balanced clustering.

## 6.2 Main Results

We demonstrate radical differences in the behaviour of clustering algorithms under the addition of a small number of elements. Using some clustering methods, clusterable subsets are robust to the influence of small sets. That is, small sets have little effect on how clusterable data is partitioned. In contrast, there are common clustering techniques in which arbitrarily well-clusterable sets admit oligarchies. That is, a small proportion of the data can have a crucial effect on the resulting clustering.

The $k$-means, $k$-medians and $k$-medoids objective functions fall in the former category. Our first main result shows that the robustness of a set to potential oligarchies with respect to these objective functions is proportional to its size and degree of clusterability.

In the following theorem, we consider a data set $X$, a typically large subset $Y \subset X$, and $O = X \setminus Y$ representing a potential oligarchy. The $\alpha$-separability and $\beta$-balance of clusterings in $Y$ quantifies its degree of clusterability. Theorem 22 bounds the robustness of $Y$ in terms of its degree of clusterability and diameter, and the relationship between its size and the size of the potential oligarchy. The theorem shows that the larger and more clusterable a subset, the more robust it is to the influence of small sets.

**Theorem 22.** *Let $F$ be one of $k$-means, $k$-medians or $k$-medoids. Let $p = 2$ if $F$ is $k$-means and $p = 1$ otherwise. Consider data sets $X$, $Y$, and $O$ where $X = Y \cup O$ and the set $Y$ has an $\alpha$-separable, $\beta$-balanced $k$-clustering of diameter $s$, for some $\alpha > 0$, $\beta \in [\frac{1}{k}, 1]$ and $s \in (0, 1]$. Then $Y$ is $\delta$-robust to $O$ with respect to $F$ for*

$$\delta \leq \tfrac{4p}{\alpha^p}(1 + \tfrac{|O|}{|Y|s^p}) + 2k \cdot \beta^2.$$

The proof appears in Section 6.3.

To see the implications of this theorem, suppose $\beta = c/k$ where $c \geq 1$ is a small constant, so that the cluster sizes are fairly balanced in $C$. Fix $s, d$ and $\alpha$, and assume $\alpha \gg 4p$. In that case, if the size of the potential oligarchy is small, $|O| \ll |Y|$, then the robustness of $Y$ is bounded by approximately $2c^2/k$.

Note that Theorem 22 applies when some of the data in $O$ is located within the convex hull of $Y$, which can be thought of as noise within $Y$. This effectively relaxes the clusterability condition on the region containing $Y$, allowing some data to lie between the well-separated clusters. Finally, note also that even if $Y$ has a very small diameter, if it is sufficiently large and clusterable, then it is robust to the influence of small sets.

In contrast to $k$-means and similar objective functions, we show that many clustering techniques do not have a property such as Theorem 22 in a strong sense. We show that algorithms that detect $\alpha$-separable clusterings, for a large enough $\alpha$, admit oligarchies. Formally, we define this property of being *$\alpha$-separability detecting* as follows.[2]

**Definition 35** ($\alpha$-Separability Detecting)**.** *A clustering function $F$ is* $\alpha$-separability-detecting *for $\alpha \geq 1$, if for all $X$ and all $2 \leq k \leq |X|$, if there exists an $\alpha$-separable $k$-clustering $C$ of $X$, then $F(X, k) = C$.*

[2] Note that for $\alpha \geq 1$, the $\alpha$-separable $k$-clustering of any given data set is unique, if it exists.

In other words, whenever there is a clustering of the full data that consists of well-separated clusters, then this clustering is produced by the algorithm.

The above property is satisfied by many well-known clustering methods. In Section 6.4, we show that the linkage-based algorithms single-linkage, average-linkage, and complete-linkage, and the min-diameter objective functions, are all 1-separability detecting, and the $k$-center objective function is 2-separability-detecting.

The following Theorem demonstrates a sharp contrast between the behaviour of $k$-means (and similar objectives) as captured in Theorem 22 and algorithms that are $\alpha$-separability detecting. It shows that for any desired level of clusterability, there exists a data set $X$ with a subset $Y \subset X$ and $O = X \setminus Y$, such that $Y$ is highly clusterable, the set $O$ representing an oligarchy contains as little as $k-1$ points, and yet $Y$ is poorly robust to $O$ with respect to these algorithms – thus $Y$ is volatile to the influence of the oligarchy $O$.

**Theorem 23.** *Let $F$ be a clustering function that is $\alpha$-separability-detecting for some $\alpha \geq 1$. Then for any $\beta \in [1/k, 1]$, $s \in [0, \frac{1}{\alpha+1})$ and any integer $m \geq k-1$, there exist data sets $X$, $Y$, and $O$ where $X = Y \cup O$, the set $O$ contains at most $m$ elements, $Y$ has an $\alpha$-separable, $\beta$-balanced $k$-clustering with diameter $s$, and yet $Y$ is not even $\beta(k-1)$-robust to $O$ with respect to $F$.*

The proof appears in Section 6.4.

For example, if $\beta = \frac{1}{k}$, then the robustness of $Y$ to $O$ is at least $\frac{k-1}{k}$, which approaches 1 as $k$ grows. Recall that 1 is the worst possible robustness score. We emphasize that the oligarchy $O$ can contain as few as $k-1$ elements, showing that $\alpha$-separability detecting algorithms are highly volatile to the influence of constant size sets.

Lastly, the behaviour of Lloyd's method depends on the method of initialization. The furthest-centroid initialization method deterministically selects a set of pairwise distant centers. We show that this algorithm is 1-separability detecting, implying that it admits oligarchies (see Section 6.4). In contrast, in Section 6.3 we discuss a result by Sivan Sabato in a co-authored paper, where it was shown that Lloyd's method with random initialization behaves similarly to the $k$-means objective function, whereby well-clusterable sets are robust to the influence of a small number of elements.

## 6.3 Methods that Prohibit Oligarchies

In this section, we study clustering methods that are robust to the influence of a small number of elements when the data is well-clusterable. We distinguish between clustering objective functions and practical clustering algorithms, providing bounds for both popular objective functions, such as $k$-means, $k$-medians and $k$-medoids, and for Lloyd's method with random center initialization, a popular heuristic for finding clusterings with low $k$-means loss.

For this section we assume that the data lays in a normed space $E$, with $d(x, y) = \|x - y\|$ for any $x, y \in E$.

### 6.3.1 $k$-means, $k$-medians and $k$-medoids

Recall that $k$-means and $k$-medians find the clustering $C = \{C_1, \ldots, C_k\}$ that minimizes the relevant cost denoted by $\text{COST}_p(C) = \sum_{i \in [k]} \min_{c_i \in E} \{\sum_{x \in C_i} \|x - c_i\|^p\}$, where the $k$-means cost is $\text{COST}_2$

and the $k$-medians cost is $\textsc{cost}_1$. The $k$-medoids cost relies on cluster centers selected from the input set, $\textsc{cost}_m(C) = \sum_{i \in [k]} \min_{c_i \in C_i}\{\sum_{x \in C_i} \|x - c_i\|\}$.

We work towards proving Theorem 22 by first showing that if the optimal clustering of a subset is relatively stable in terms of cost, then the subset is robust. Some stability assumption is necessary, since if there are two very different clusterings for the data set which have very similar costs, then even a single additional point might flip the balance between the two clusterings. We use the following notion of a cost-optimal clustering (which bears similarity to a notion by Balcan et al. [12]).

**Definition 36** (($\delta, c$)-cost-optimal)**.** *A clustering $C$ of $X$ is* ($\delta, c$)-cost-optimal *with respect to a cost function* $\textsc{cost}$ *if for all clusterings $C'$ of $X$ for which* $\textsc{cost}(C') \leq \textsc{cost}(C) + c$, $\Delta(C, C') \leq \delta$.

In Lemma 24, we demonstrate the existence of $(\delta, c)$-cost-optimal clustering, see the discussion below Lemma 24 for details. In addition, Meila [40] shows that clusterings that are good in terms of their $k$-means cost are also structurally similar to the optimal solution, using misclassification error for distance between clusterings.

**Lemma 22.** *Let $F$ be one of $k$-means, $k$-medians or $k$-medoids. Consider data sets $X$ and $Y \subseteq X$. If there exists a $(\delta, |X \setminus Y|)$-cost-optimal clustering of $Y$ relative to the cost associated with $F$, then $Y$ is $2\delta$-robust in $X$ with respect to $F$.*

*Proof.* Let $C = \{C_1, \ldots, C_k\}$ be the assumed cost-optimal clustering of $Y$. Let $\textsc{cost}$ be the cost associated with $F$. Let $p = 2$ if $F$ is $k$-means and $p = 1$ otherwise. For $i \in [k]$, let $T_i = E$ if $F$ is $k$-means or $k$-medians, and let $T_i = C_i$ if $F$ is $k$-medoids. Let $\bar{c}_i = \mathrm{argmin}_{c_i \in T_i}\{\sum_{x \in C_i} \|x - c_i\|^p\}$. Then, the cost of the clustering $F(X)$ is at most the cost of the clustering $C_1, \ldots, C_{k-1}, C_k \cup X \setminus Y$, since this is a possible clustering of $X$. Thus

$$\textsc{cost}(F(X)) \leq \sum_{i \in [k]} \sum_{x \in C_i} \|x - \bar{c}_i\|^p + \sum_{z \in X \setminus Y} \|z - \bar{c}_k\|^p.$$

In all the possibilities for $F$, $\bar{c}_i$ is in the convex hull of $X$ which has a diameter at most 1. Thus for all $z \in X \setminus Y$, $\|z - \bar{c}_k\|^p \leq 1$. Since $\textsc{cost}(F(X)|Y) \leq \textsc{cost}(F(X))$, it follows that

$$\textsc{cost}(F(X)|Y) \leq \sum_{i \in [k]} \sum_{x \in C_i} \|x - \bar{c}_i\|^p + |X \setminus Y| = \textsc{cost}(C) + |X \setminus Y|.$$

Thus, by the cost-optimality property of $C$, if $c \geq |X \setminus Y|$ then $\Delta(F(X)|Y, C) \leq \delta$. In addition, $\textsc{cost}(F(Y)) \leq \textsc{cost}(C)$, thus for any $c \geq 0$, $\Delta(F(Y), C) \leq \delta$. It follows that $\Delta(F(X)|Y, F(Y)) \leq 2\delta$, thus the robustness of $Y$ in $X$ with respect to $F$ is at most $2\delta$. □

The next lemma provides a useful connection between the Hamming distance of two clusterings, and the number of disjoint pairs that belong to the same cluster in one clustering, but to different clusters in the other.

**Lemma 23.** *Let $C_1$ and $C_2$ be two clusterings of $Y$, where $C_1$ is $\beta$-balanced and has $k$ clusters. If $\Delta(C_1, C_2) \geq \delta$, then the number of disjoint pairs $\{x, y\} \subseteq Y$ such that $x \nsim_{C_1} y$ and $x \sim_{C_2} y$ is at least $\frac{1}{2}(\delta - k \cdot \beta^2)|Y|$.*

*Proof.* Let $A = \{\{x, y\} \mid x \not\sim_{C_1} y, x \sim_{C_2} y\}$, and let $B = \{\{x, y\} \mid x \sim_{C_1} y, x \not\sim_{C_2} y\}$. If $\Delta(C_1, C_2) \geq \delta$ then $|A \cup B| \geq \frac{1}{2}\delta|Y|(|Y| - 1)$. Since every cluster in $C_1$ is of size at most $\beta|Y|$,

$$|B| \leq |\{\{x, y\} \mid x \sim_{C_1} y\}| \leq \tfrac{1}{2}k \cdot \beta|Y|(\beta|Y| - 1).$$

Thus

$$|A| \geq \tfrac{1}{2}\delta|Y|(|Y| - 1) - \tfrac{1}{2}k \cdot \beta|Y|(\beta|Y| - 1) \geq \tfrac{1}{2}(\delta - k \cdot \beta^2)|Y|(|Y| - 1).$$

Now, for every $x$ such that $\{x, y\} \in A$, there are at most $|Y| - 1$ pairs in $A$ that include $x$. Thus the number of disjoint pairs in $A$ is at least $|A|/(|Y| - 1)$. Therefore that are at least $\frac{1}{2}(\delta - k \cdot \beta^2)|Y|$ disjoint pairs in $A$. □

We now show that clusterings that are balanced and well-separable in a geometrical sense are also cost-optimal.

**Lemma 24.** *Suppose a $k$-clustering $C$ of $Y$ is $\alpha$-separable, $\beta$-balanced and has diameter $s$. Let* COST *be one of* $\text{COST}_1$, $\text{COST}_2$ *or* $\text{COST}_m$. *Let $p = 2$ if* COST *is* $\text{COST}_2$ *and $p = 1$ otherwise. Then for any $\delta \in (0, 1)$, $C$ is $(\delta, |Y|s^p(\frac{\alpha^p(\delta - k\cdot\beta^2)}{2p} - 1))$-cost-optimal with respect to* COST.

*Proof.* Let $C'$ be a clustering of $Y$ such that $\Delta(C, C') \geq \delta$. For $i \in [k]$, let $T_i = E$ if $F$ is $k$-means or $k$-medians, and let $T_i = C_i$ if $F$ is $k$-medoids. Let $c_i = \text{argmin}_{c_i \in T_i}\{\sum_{x \in C_i} \|x - c_i\|^p\}$, and $c'_i = \text{argmin}_{c'_i \in T_i}\{\sum_{x \in C'_i} \|x - c'_i\|^p\}$. For every cluster $C_i$ in $C$, and every $x \in C_i$, $\|x - c_i\|^p \leq s^p$. Thus $\text{COST}(C) \leq |Y|s^p$. On the other hand, for every pair $\{x, y\} \subseteq Y$, if $x \not\sim_C y$ and $x \sim_{C'} y$, then for $p = \{1, 2\}$

$$\|x - c'_i\|^p + \|y - c'_i\|^p \geq \|x - y\|^p/p \geq (\alpha s)^p/p.$$

The first inequality is the triangle inequality for $p = 1$. For $p = 2$ the inequality can be derived by observing that the left hand side is minimized for $c'_i = (x + y)/2$. The last inequality follows from the properties of $C$ and the fact that $x \not\sim_C y$. By Lemma 23, there are at least $|Y|\frac{1}{2}(\delta - k \cdot \beta^2)$ such $\{x, y\}$ pairs. Thus $\text{COST}(C') \geq |Y|\frac{1}{2p}(\alpha s)^p(\delta - k \cdot \beta^2)$. It follows that $\text{COST}(C') - \text{COST}(C) \geq |Y|(\frac{1}{2p}(\alpha s)^p(\delta - k \cdot \beta^2) - s^p)$. The lemma follows from the definition of cost-optimality. □

Consider the parameters $\delta$ and $c$ from the notion of a $(\delta, c)$-cost-optimal clustering. The above lemma requires that $\delta > k\beta^2$, and so $\delta > \frac{1}{k}$. Therefore, this lemma holds for small $\delta$ when the number of clusters, $k$, is large. A small value of $c$ can be obtained by setting the separability parameter $\alpha$ to the appropriate value. For example, let $\beta = 1/k$ and let $\epsilon = \delta - k \cdot \beta^2$. Then set $\alpha$ so that $\frac{\alpha^p \cdot \epsilon}{2p}$ is larger than, but close to 1; The closer is this value to 1, the smaller the resulting value of $c$.

We now combine the above lemmas to bound the robustness of a clusterable set $Y$ to a potential oligarchy, thereby proving Theorem 22.

**Theorem 22** (restated): *Let $F$ be one of $k$-means, $k$-medians or $k$-medoids. Let $p = 2$ if $F$ is $k$-means and $p = 1$ otherwise. Consider data sets $X$, $Y$, and $O$ where $X = Y \cup O$ and the set $Y$ has an $\alpha$-separable, $\beta$-balanced $k$-clustering of diameter $s$, for some $\alpha > 0$, $\beta \in [\frac{1}{k}, 1]$ and $s \in (0, 1]$. Then $Y$ is $\delta$-robust to $O$ with respect to $F$ for*

$$\delta \leq \tfrac{4p}{\alpha^p}(1 + \tfrac{|O|}{|Y|s^p}) + 2k \cdot \beta^2.$$

The proof of this theorem follows by letting $\delta' = \frac{2p}{\alpha^p}(1 + \frac{|O|}{|Y|s^p}) + k \cdot \beta^2$. Then, by Lemma 24, $C$ is $(\delta', |O|)$-cost-optimal. Thus by Lemma 22, the robustness of $Y$ to $O$ is at most $2\delta'$.

### 6.3.2 Lloyd's Method with Random Initial Centers

The results above pertain to algorithms that find the minimal-cost clustering. In practice, this task is often not tractable, and algorithms that search for a locally optimal clustering are used instead. For $k$-means, a popular algorithm is Lloyd's method. A common initialization for Lloyd's method is to select $k$ random points from the input data set [28]. We call this algorithm Randomized Lloyd. It is also commonly referred to as "the k-means algorithm." In order to find a solution with low $k$-means loss, it is common practice to run Randomized Lloyd multiple times and then select the minimal cost clustering. We show that clusterable data sets are immune to the influence of oligarchies when Randomizes Lloyd is repeated enough times. Specifically, we show that large clusterable subsets are robust with respect to this technique.

The following result is by Sivan Sabato in a co-authored paper and is included here for completeness.

**Theorem 24.** *Consider data sets $X$, $Y$ and $O$ where $X = Y \cup X$ such that there exists an $\alpha$-separable, $\beta$-balanced $k$-clustering $C$ of $Y$ with diameter $s > 0$, for some $\alpha \geq 3$. Let $m$ be the size of the smallest cluster in $C$, and assume $m \geq \frac{2|O|}{(\alpha-1)s}$. Then with probability at least $1 - \epsilon$, $Y$ is $\delta$-robust to $O$ with respect to $n$ runs of Randomized Lloyd, for*

$$n \geq \left(\frac{e|X|}{km}\right)^k \log(2/\epsilon),$$

*and*

$$\delta \leq \frac{8}{\alpha^2}\left(1 + \frac{|O|}{|Y|s^2}\right) + 2\beta^2 k.$$

## 6.4 Methods that Admit Oligarchies

We now turn to algorithms that admit oligarchies. We prove that all algorithms that detect $\alpha$-separable clusterings admit oligarchies even on data that is highly clusterable. In this section, we prove Theorem 23 from Section 6.2, demonstrating a sharp contrast between the behaviour of $\alpha$-separability detecting algorithms and the behaviour captured in Theorem 22 for $k$-means and similar objective functions. Next, we will show that many well-known clustering methods are $\alpha$-separability-detecting.

**Theorem 23** (restated): *Let $F$ be a clustering function that is $\alpha$-separability detecting for some $\alpha \geq 1$. Then for any $\beta \in [1/k, 1]$, $s \in [0, \frac{1}{\alpha+1})$ and any integer $m \geq k - 1$, there exist data sets $X$, $Y$, and $O$ where $X = Y \cup O$, the set $O$ consists of at most $m$ elements, $Y$ has an $\alpha$-separable, $\beta$-balanced $k$-clustering with diameter $s$, and yet $Y$ is not even $\beta(k-1)$-robust to $O$ with respect to $F$.*

*Proof.* Let $Y$ be a set of points with diameter $s$ that contains most of the elements in $X$, and make it so that $Y$ has an $\alpha$-separable, $\beta$-balanced $k$-clustering. The data set $O$ contains $k-1$ points at distance $\alpha s + \epsilon$ from each other and from any point in $Y$. Then $F(X,k)$ places all elements in $Y$ within the same cluster, while $F(Y,k)$ produces a $\beta$-balanced clustering of $Y$. □

Theorem 23 shows that even when $Y$ is very large ($\frac{|Y|}{|X|}$ can be arbitrarily close to 1) and has an arbitrarily well-separable ($\alpha$ can be arbitrary large) and balanced partition ($\beta = \frac{1}{k}$), the robustness score of $Y$ to the oligarchy $O$ can be bounded from below by $\beta(k-1)$, which approaches the worst possible score of robustness 1 as $k$ grows. This shows that $\alpha$-separability detecting algorithms admit oligarchies of constant size (in particular, size $k-1$), even on data that is highly clusterable.

We now continue to show that several well-known algorithms are separability-detecting, resulting in the immediate conclusion that Theorem 23 holds for them.

### 6.4.1 Separability-Detecting Algorithms

In this section, we show that several common algorithms are $\alpha$-separability-detecting. First, we consider linkage-based clustering, one of the most commonly-used clustering paradigms. Linkage-based algorithms use a greedy approach; at first every element is in its own cluster. Then the algorithm repeatedly merges the "closest" pair of clusters until some stopping criterion is met. To identify the closest clusters, these algorithms use a linkage function, which maps each pair of clusters to a real number representing their proximity. See Chapter 4 for more detail.

Consider the following condition: For all choices of $A, B$ and distance function $d$,

$$\min_{a \in A, b \in B} d(a,b) \leq \ell(A,B,d) \leq \max_{a \in A, b \in B} d(a,b). \tag{6.1}$$

Observe that the linkage functions of the most common linkage-based algorithms, single-linkage, average-linkage, and complete-linkage, all satisfy the above condition.

We consider linkage-based algorithms with the $k$-stopping criterion, which terminates a linkage-based algorithm when $k$ clusters remain, and returns the resulting clustering.

**Theorem 25.** *Let $F$ be a clustering function that uses a linkage-based function $\ell$ to merge clusters, and stops when there are $k$ clusters. If Equation 6.1 holds for $\ell$, then $F$ is 1-separability-detecting.*

*Proof.* By way of contradiction, assume that there exists a data set $(X,d)$ with a 1-separable $k$-clustering $C$, but $F(X,k) \neq C$. Consider the first iteration of the algorithm in which the clustering stops being a refinement of $C$. Let $C'$ be the clustering before this iteration. There are clusters $C'_1, C'_2, C'_3 \in C'$ such that $C'_1, C'_2 \in C_i$ for some $i$, $C'_3 \in C_j$ for $j \neq i$, and the algorithm merges $C'_1$ and $C'_3$. Thus $\ell(C'_1, C'_2, d) \geq \ell(C'_1, C'_3, d)$. By Eq. 6.1, $\ell(C'_1, C'_2, d) \leq \max_{a \in C'_1, b \in C'_2} d(a,b)$, and $\min_{a \in C'_1, b \in C'_3} d(a,b) \leq \ell(C'_1, C'_3, d)$. Since $C$ is 1-separable, $\max_{a \in C'_1, b \in C'_2} d(a,b) < \min_{a \in C'_1, b \in C'_3} d(a,b)$, so $\ell(C'_1, C'_2, d) < \ell(C'_1, C'_3, d)$, contradicting the assumption. □

There are also clustering objective functions that are $\alpha$-separability-detecting. Thus clustering algorithms that minimize them satisfy Theorem 23. The min-diameter objective function [9] is simply the diameter of the clustering. We show that it is 1-separability-detecting.

**Theorem 26.** *Min-diameter is* 1*-separability-detecting.*

*Proof.* For a set $X$, assume that there exists a 1-separable $k$-clustering $C$ with diameter $s$. For any $k$-clustering $C' \neq C$ there are points $x, y$ such that $x \sim_{C'} y$ while $x \nsim_C y$. $d(x, y) > s$, thus the diameter of $C'$ is larger than $s$. Thus $C'$ is not the optimal clustering for $X$. □

The $k$-center [8] objective functions finds a clustering that minimizes the maximum radius of any cluster in the clustering. In $k$-center the centers are arbitrary points in the underlying space, and in *discrete k-center* they are a subset of the input points. We show that if $d$ satisfies the triangle inequality then $k$-center and discrete $k$-center are 2-separability detecting.

**Theorem 27.** *If $d$ satisfies the triangle inequality then $k$-center and discrete $k$-center are* 2*-separability detecting.*

*Proof.* Assume that there exists a 2-separable $k$-clustering $C$ of a set $X$. Then the $k$-center cost is at most the diameter of $C$. For any $k$-clustering $C' \neq C$ there are points $x, y$ such that $x \nsim_C y$ while $x \sim_{C'} y$. Hence the radius of $C'$ is at least $\frac{1}{2} \cdot \min_{x \not\sim_C y} d(x, y) > \max_{x \sim_C y} d(x, y)$, and thus it is larger than the cost of $C$. The proof for discrete $k$-center is similar. □

### 6.4.2 Lloyd's Method with Furthest Centroids Initialization

Large clusterable sets are robust with respect to Randomized Lloyd. This does not hold for the furthest-centroid initialization method [34], which admits oligarchies. The method is described in detail in Section 2.2.2.

**Lemma 25.** *Lloyd's method with furthest centroid initialization is* 1*-separability detecting.*

*Proof.* If $Z$ has a 1-separable $k$-clustering $C$, then between-cluster distances are larger than within-cluster distances. Thus, for every $i \geq 2$, the cluster of $C$ that includes $c_i$ is different from the clusters that include $c_1, \ldots, c_{i-1}$. Thus the clustering induced by the initial points is $C$. In the next iteration the centers remain unchanged, thus the clustering remains $C$. □

## 6.5 Related Work

Hennig [30] performed a similar analysis of how algorithms respond to the addition of small sets, with one important difference: the diameter of data sets was not bounded. As a result, all algorithms considered, including $k$-means, were sensitive to oligarchies. That is, outliers that are placed sufficiently far are assigned their own clusters, even when $k$-means is used. If the number of clusters is fixed, and the diameter of the data is not fixed, then even $k$-means is not robust to oligrachies. By restricting the diameter of data sets, we are able to differentiate between the behaviour of k-means and that of common linkage-based algorithms based on their robustness to small sets.

There is also a related line of work in the *planted partition model*. In this model, given a clustering $C$ of $X$, a random graph $G = (X, E)$ is constructed by placing an edge between nodes $x$ and $y$ with probability $p$ whenever $x \sim_C y$, and probability $q < p$ whenever $x \not\sim_C y$. The objective is then to recover the partition $C$ given the random graph $G$, with high probability. Several algorithms for this problem have been proposed ([15], [21],[37]). These algorithms uncover $C$ when a large number of outliers are added. Therefore, they are both able to detect well separable clusters and are robust to oligarchies. There are several important differences from our setting that make this possible. Primarily, the number of clusters that algorithms should output is not restricted in the planted partition model. Note that our proof showing that $\alpha$-separable algorithms are susceptible to oligarchies relies on there being a fixed number of clusters. In addition, the planted partition model is concerned with the case where data has only two similarity values.

# CHAPTER 7

# A CHARACTERIZATION OF HIERARCHICAL LINKAGE-BASED ALGORITHMS

In this chapter, we extend our characterization of linkage-based algorithms into the hierarchical setting. Hierarchical algorithms output dendrograms, which users can then traverse to obtain a desired clustering. Dendrograms provide a convenient method for exploring multiple clusterings of the data. Indeed, for some applications the dendrogram itself, not any clustering found in it, is the desired final outcome. One such application is found in the field of phylogenetics, which aims to reconstruct the tree of life.

We provide a property-based characterization of hierarchical linkage-based algorithms, identifying two properties of hierarchical algorithms that are satisfied by all linkage-based algorithms, and prove that at the same time no algorithm that is not linkage-based can satisfy both of these properties.

The popularity of linkage-based algorithms lead to a common misconception that linkage-based algorithms are synonymous with hierarchical algorithms. We show that even when the internal workings of algorithms are ignored, and the focus is placed solely on their input-output behaviour, there are natural hierarchical algorithms that are not linkage-based. We define a large class of divisive algorithms that includes the popular bisecting $k$-means algorithm, and show that no linkage-based algorithm can simulate the input-output behaviour of any algorithm in this class.

## 7.1 Definitions

We introduce several definitions specific to the hierarchical clustering setting.

Given a rooted directed tree $T$ where the edges are oriented away from the root, let $V(T)$ denote the set of vertices in $T$, and $E(T)$ denote the set of edges in $T$. We use the standard interpretation of the terms leaf, descendant, parent, and child.

A dendrogram over a data set $X$ is a binary rooted tree where the leaves correspond to elements of $X$. In addition, every node is assigned a level, using a level function ($\eta$); leaves are placed at

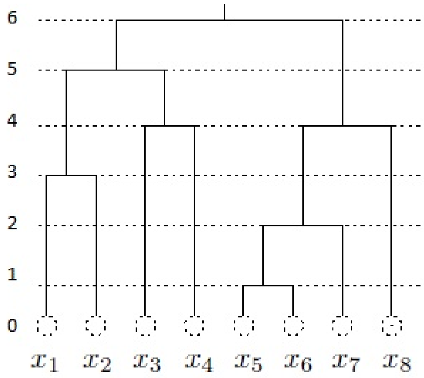


Figure 7.1: A dendrogram of domain set $\{x_1, \ldots, x_8\}$. The horizontal lines represent levels and every leaf is associated with an element of the domain.

level 0, parents have higher levels than their children, and no level is empty. See Figure 7.1 for an illustration. Formally,

**Definition 37** (dendrogram)**.** *A* dendrogram *over $(X, d)$ is a triple $(T, M, \eta)$ where $T$ is a binary rooted tree, $M : \text{leaves}(T) \to X$ is a bijection, and $\eta : V(T) \to \{0, \ldots, h\}$ is surjective (for some $h \in \mathcal{Z}^+ \cup \{0\}$) such that*

1. *For every leaf node $x \in V(T)$, $\eta(x) = 0$.*
2. *If $(x, y) \in E(T)$, then $\eta(x) > \eta(y)$.*

Given a dendrogram $\mathcal{D} = (T, M, \eta)$ of $X$, we define a mapping from nodes to clusters $\mathcal{C} : V(T) \to 2^X$ by $\mathcal{C}(x) = \{M(y) \mid y \text{ is a leaf and a descendent of } x\}$. If $\mathcal{C}(x) = A$, then we write $v(A) = x$. We think of $v(A)$ as the vertex (or node) in the tree that represents cluster $A$.

We say that $A \subseteq X$ is a cluster in $\mathcal{D}$ if there exists a node $x \in V(T)$ so that $\mathcal{C}(x) = A$. We say that a clustering $C = \{C_1, \ldots, C_k\}$ of $X' \subseteq X$ is in $\mathcal{D}$ if $C_i$ is in $\mathcal{D}$ for all $1 \leq i \leq k$. Note that a dendrogram may contain clusterings that do not partition the entire domain, and $\forall i \neq j$, $v(C_i)$ is not a descendent of $v(C_j)$, since $C_i \cap C_j = \emptyset$.

**Definition 38** (sub-dendrogram)**.** *A* sub-dendrogram *of $(T, M, \eta)$ rooted at $x \in V(T)$ is a dendrogram $(T', M', \eta')$ where*

1. *$T'$ is the subtree of $T$ rooted at $x$,*
2. *For every $y \in \text{leaves}(T')$, $M'(y) = M(y)$, and*
3. *For all $y, z \in V(T')$, $\eta'(y) < \eta'(z)$ if and only if $\eta(y) < \eta(z)$.*

**Definition 39** (Isomorphisms)**.** *A few notions of isomorphisms of structures are relevant to our discussion.*

1. *We say that* $(T_1, \eta_1)$ *and* $(T_2, \eta_2)$ *are* isomorphic trees*, denoted* $(T_1, \eta_1) \cong_T (T_1, \eta_1)$*, if there exists a bijection* $H : V(T_1) \to V(T_2)$ *so that*

   (a) *for all* $x, y \in V(T_1)$*,* $(x, y) \in E(T_1)$ *if and only if* $(H(x), H(y)) \in E(T_2)$*, and*

   (b) *for all* $x \in V(T_1)$*,* $\eta_1(x) = \eta_2(H(x))$*.*

2. *We say that* $\mathcal{D}_1 = (T_1, M_1, \eta_1)$ *of* $(X, d)$ *and* $\mathcal{D}_2 = (T_2, M_2, \eta_2)$ *of* $(X', d')$ *are* isomorphic dendrograms*, denoted* $\mathcal{D}_1 \cong_D \mathcal{D}_2$*, if there exists a domain isomorphism* $\phi : X \to X'$ *and a tree isomorphism* $H : (T_1, \eta_1) \to (T_2, \eta_2)$ *so that for all* $x \in \text{leaves}(T_1)$*,* $\phi(M_1(x)) = M_2(H(x))$*.*

## 7.2 Hierarchical and Linkage-Based Algorithms

In the hierarchical clustering setting, linkage-based algorithms are hierarchical algorithms that can be simulated by repeatedly merging close clusters. In this section, we formally define hierarchical algorithms and linkage-based hierarchical algorithms.

### 7.2.1 Hierarchical Algorithms

In addition to outputting a dendrogram, we require that hierarchical clustering functions satisfy a few natural properties.

**Definition 40** (Hierarchical clustering function)**.** *A* hierarchical clustering function *$F$ is a function that takes as input a pair $(X, d)$ and outputs a dendrogram $(T, M, \eta)$. We require such a function, $F$, to satisfy the following:*

1. Representation Independence*: Whenever* $(X, d) \cong_X (X', d')$*, then* $F(X, d) \cong_D F(X', d')$*.*

2. Scale Invariance*: For any domain set $X$ and any pair of distance functions $d, d'$ over $X$, if there exists $c \in \mathbb{R}^+$ such that $d(a, b) = c \cdot d'(a, b)$ for all $a, b \in X$, then $F(X, d) = F(X, d')$.*

3. Richness*: For all data sets $\{(X_1, d_1), \ldots, (X_k, d_k)\}$ where $X_i \cap X_j = \emptyset$ for all $i \neq j$, there exists a distance function $\hat{d}$ over $\bigcup_{i=1}^k X_i$ that extends each of the $d_i$'s (for $i \leq k$), so that the clustering $\{X_1, \ldots, X_k\}$ is in $F(\bigcup_{i=1}^k X_i, \hat{d})$.*

The last condition, richness, requires that by manipulating between-cluster distances every clustering can be produced by the algorithm. Intuitively, if we place the clusters sufficiently far apart, then the resulting clustering should be in the dendrogram.

In this work we focus on distinguishing linkage-based algorithms from other hierarchical algorithms.

### 7.2.2 Linkage-Based Algorithms

We defined linkage functions in Chapter 4. For the current characterization, it suffices to use a relaxation of that definition, by omitting the last condition.

**Definition 41** (Linkage Function)**.** *A* linkage function *is a function*

$$\ell : \{(X_1, X_2, d) \mid d \text{ over } X_1 \cup X_2\} \to \mathbb{R}^+$$

*such that,*

1. $\ell$ *is* representation independent*: For all* $(X_1, X_2)$ *and* $(X'_1, X'_2)$*, if* $(\{X_1, X_2\}, d) \cong_C (\{X'_1, X'_2\}, d')$ *then* $\ell(X_1, X_2, d) = \ell(X'_1, X'_2, d')$*.*
2. $\ell$ *is* monotonic*: For all* $(X_1, X_2, d)$ *if* $d'$ *is a distance function over* $X_1 \cup X_2$ *such that for all* $x \sim_{\{X_1, X_2\}} y$*,* $d(x, y) = d'(x, y)$ *and for all* $x \not\sim_{\{X_1, X_2\}} y$*,* $d(x, y) \leq d'(x, y)$ *then* $\ell(X_1, X_2, d') \geq \ell(X_1, X_2, d)$*.*

As in our characterization of partitional linkage-based algorithms, we assume that a linkage function has a countable range. Say, the set of non-negative algebraic real numbers.

For a dendrogram $\mathcal{D}$ and clusters $A$ and $B$ in $\mathcal{D}$, if there exists $x$ so that $\text{parent}(v(A)) = \text{parent}(v(B)) = x$, then let $\text{parent}(A, B) = x$, otherwise $\text{parent}(A, B) = \emptyset$.

We now define hierarchical linkage-based functions.

**Definition 42** (Linkage-Based Function)**.** *A hierarchical clustering function* $F$ *is* linkage-based *if there exists a linkage function* $\ell$ *so that for all* $(X, d)$*,* $F(X, d) = (T, M, \eta)$ *where* $\eta(\text{parent}(A, B)) = m$ *if and only if* $\ell(A, B)$ *is minimal in* $\{\ell(S, T) : S \cap T = \emptyset, \eta(S) < m, \eta(T) < m, \eta(\text{parent}(S)) \geq m, \eta(\text{parent}(T)) \geq m\}$*.*

Note that the above definition implies that there exists a linkage function that can be used to simulate the output of $F$. We start by assigning every element of the domain to a leaf node. We then use the linkage function to identify the closest pair of nodes (with respect to the clusters that they represent), and repeatedly merge the closest pairs of nodes that do yet have parents, until only one such node remains.

### 7.2.3 Locality

We formulate the locality property from Chapter 4 in the hierarchical setting. Locality states that if we select a clustering from a dendrogram (a union of disjoint clusters that appear in the dendrogram), and run the hierarchical algorithm on the data underlying this clustering, we obtain a result that is consistent with the original dendrogram.

**Definition 43** (Locality)**.** *A hierarchical function* $F$ *is* local *if for all* $X$*,* $d$*, and* $X' \subseteq X$*, whenever clustering* $C = \{C_1, C_2, \ldots, C_k\}$ *of* $X'$ *is in* $F(X, d) = (T, M, \eta)$*, then for all* $1 \leq i \leq k$

1. *Cluster* $C_i$ *is in* $F(X', d|X') = (T', M', \eta')$*, and the sub-dendrogram of* $F(X, d)$ *rooted at* $v(C_i)$ *is also a sub-dendrogram of* $F(X', d|X')$ *rooted at* $v(C_i)$*.*
2. *For all* $x, y \in X'$*,* $\eta'(x) < \eta'(y)$ *if and only if* $\eta(x) < \eta(y)$*.*

Locality is often a desirable property. Consider for example the field of phylogenetics, which aims to reconstruct the tree of life. If an algorithm clusters phylogenetic data correctly, then if we cluster any subset of the data, we should get results that are consistent with the original dendrogram.

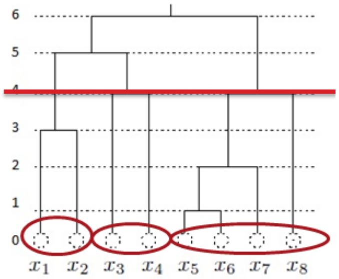


Figure 7.2: An example of an $A$-cut.

### 7.2.4 Outer Consistency

A basic requirement from a good clustering is that it separate dissimilar elements. Given successfully clustered data, if points that are already assigned to different clusters are drawn even further apart, then it is natural to expect that, when clustering the resulting new data set, such points will not share the same cluster. We now formulate the outer-consistency property from Chapter 4 in the hierarchical setting.

Given a dendrogram produced by a hierarchical algorithm, we select a clustering $C$ from a dendrogram and pull apart the clusters in $C$ (thus making the clustering $C$ more pronounced). If we then run the algorithm on the resulting data, we can expect that the clustering $C$ will occur in the new dendrogram. Outer consistency is a relaxation of the above property, making this requirement only on a subset of clusterings.

For a cluster $A$ in a dendrogram $\mathcal{D}$, the $A$-cut of $\mathcal{D}$ is a clustering in $\mathcal{D}$ represented by nodes on the same level as $v(A)$ or directly below $v(A)$. Formally,

**Definition 44** ($A$-cut)**.** *Given a cluster $A$ in a dendrogram $\mathcal{D} = (T, M, \eta)$, the* $A$-cut *of $\mathcal{D}$ is $cut_A(\mathcal{D}) = \{\mathcal{C}(u) \mid u \in V(T), \eta(\text{parent}(u)) > \eta(v(A))$ and $\eta(u) \leq \eta(v(A)).\}$.*

Note that for any cluster $A$ in $\mathcal{D}$ of $(X, d)$, the $A$-cut is a clustering of $X$, and $A$ is one of the clusters in that clustering.

For example, consider the diagram in Figure 7.2. Let $A = \{x_3, x_4\}$. The horizontal line on level 4 of the dendrogram represents the intuitive notion of a cut. To obtain the corresponding clustering, we select all clusters represented by nodes on the line, and for the remaining clusters, we choose clusters represented by nodes that lay directly below the horizontal cut. In this example, clusters $\{x_3, x_4\}$ and $\{x_5, x_6, x_7, x_8\}$ are represented by nodes directly on the line, and $\{x_1, x_2\}$ is a cluster represented by a node directly below the marked horizontal line.

Recall that a distance function $d'$ over $X$ is $(C, d)$-*outer-consistent* if $d'(x, y) = d(x, y)$ whenever $x \sim_C y$, and $d'(x, y) \geq d(x, y)$ whenever $x \not\sim_C y$.

**Definition 45** (Outer-Consistency)**.** *A hierarchical function $F$ is* outer consistent *if for all $(X, d)$ and any cluster $A$ in $F(X, d)$, if $d'$ is $(cut_A(F(X, d)), d)$-outer-consistent then $cut_A(F(X, d)) = cut_A(F(X, d'))$.*

## 7.3 Main Result

The following is our characterization of linkage-based hierarchical algorithms.

**Theorem 28.** *A hierarchical function $F$ is linkage-based if and only if $F$ is outer consistent and local.*

We prove the result in the following subsections (one for each direction of the iff). In the last part of this section, we demonstrate the necessity of both properties.

### 7.3.1 All Local, Outer-Consistent Hierarchical Functions are Linkage-Based

**Lemma 26.** *If a hierarchical function $F$ is outer-consistent and local, then $F$ is linkage-based.*

*Proof.* We show that there exists a linkage function $\ell$ so that when $\ell$ is used in Definition 42 then for all $(X, d)$ the output is $F(X, d)$. Due to the representation independence of $F$, one can assume w.l.o.g., that the domain sets over which $F$ is defined are (finite) subsets of the set of natural numbers, $\mathcal{N}$.

**Definition 46** (The (pseudo-) partial ordering $<_F$). *We consider triples of the form $(A, B, d)$, where $A \cap B = \emptyset$ and $d$ is a distance function over $A \cup B$. Two triples, $(A, B, d)$ and $(A', B', d')$ are equivalent, denoted $(A, B, d) \cong (A', B', d')$ if they are isomorphic as clusterings, namely, if $(\{A, B\}, d) \cong_C (\{A', B'\}, d')$.*

*$<_F$ is a binary relation over equivalence classes of such triples, indicating that $F$ merges a pair of clusters earlier than another pair of clusters. Formally, denoting $\cong$-equivalence classes by square brackets, we define it by: $[(A, B, d)] <_F [(A', B', d')]$ if*

1. *At most two sets in $\{A, B, A', B'\}$ are equal and no set is a strict subset of another.*
2. *The distance functions $d$ and $d'$ agree on $(A \cup B) \cap (A' \cup B')$.*
3. *There exists a distance function $d^*$ over $X = A \cup B \cup A' \cup B'$ so that $F(X, d^*) = (T, M, \eta)$ such that*
   - *(a) $d^*$ extends both $d$ and $d'$,*
   - *(b) There exist $(x, y), (x, z) \in E(T)$ such that $\mathcal{C}(x) = A \cup B$, $\mathcal{C}(y) = A$, and $\mathcal{C}(z) = B$*
   - *(c) For all $D \in \{A', B'\}$, either $D \subseteq A \cup B$, or $D \in cut_{A\cup B}F(X, d^*)$.*
   - *(d) $\eta(v(A')) < \eta(v(A \cup B))$ and $\eta(v(B')) < \eta(v(A \cup B))$.*

Since we define hierarchical algorithms to be representation independent, we can just discuss triples, instead of their equivalence classes. For the sake of simplifying notation, we will omit the square brackets in the following discussion.

In the following lemma we show that if $(A, B, d) <_F (A', B', d')$, then $A' \cup B'$ cannot have a lower level than $A \cup B$.

**Lemma 27.** *Given a local and outer-consistent hierarchical function $F$, whenever*

*$(A_1, B_1, d_1) <_F (A_2, B_2, d_2)$, there is no data set $(X, d)$ such that $A_1, B_1, A_2, B_2 \subseteq X$ and $\eta(v(A_2 \cup B_2)) \leq \eta(v(A_1 \cup B_1))$, where $F(X, d) = (T, M, \eta)$.*

*Proof.* By way of contradiction, assume that such $(X, d)$ exists. Let $X' = A_1 \cup B_1 \cup A_2 \cup B_2$. Since $(A_1, B_1, d_1) <_F (A_2, B_2, d_2)$, there exists $d'$ that satisfies the conditions of Definition 46.

Consider $F(X', d|X')$. By locality, the sub-dendrogram rooted at $v(A_1 \cup B_1)$ contains the same nodes in both $F(X', d|X')$ and $F(X, d)$, and similarly for the sub-dendrogram rooted at $v(A_2 \cup B_2)$. In addition, the relative level of nodes in these subtrees is the same.

Construct a distance function $d^*$ over $X'$ that is both $(\{A_1 \cup B_1, A_2 \cup B_2\}, d|X')$-outer consistent and $(\{A_1 \cup B_2, A_2, B_2\}, d')$-outer consistent as follows:

- $d^*(x, y) = \max(d(x, y), d'(x, y))$ whenever $x \in A_1 \cup B_1$ and $y \in A_2 \cup B_2$
- $d^*(x, y) = d_1(x, y)$ whenever $x, y \in A \cup B$
- $d^*(x, y) = d_2(x, y)$ whenever $x, y \in A' \cup B'$

Note that $\{A_1 \cup B_1, A_2 \cup B_2\}$ is an $(A_1 \cup B_1)$-cut of $F(X', d|X')$. Therefore, by outer-consistency, $cut_{A_1 \cup B_1}(F(X', d^*)) = \{A_2 \cup B_2, A_1 \cup B_1\}$.

Since $d'$ satisfies the conditions in Definition 46, $cut_{A_1 \cup B_1} F(X, d') = \{A_1 \cup B_1, A_2, B_2\}$. By outer-consistency we get that $cut_{A_1 \cup B_1}(F(X', d^*)) = \{A_2 \cup B_2, A_1, B_1\}$. Since these sets are all non-empty, this is a contradiction. □

We now define equivalence with respect to $<_F$.

**Definition 47** ($\cong_F$)**.** *$[(A, B, d)]$ and $[(A', B', d')]$ are $F$-equivalent, denoted $[(A, B, d)] \cong_F [(A', B', d')]$, if*

1. *At most two sets in $\{A, B, A', B'\}$ are equal and no set is a strict subset of another.*
2. *The distance function $d$ and $d'$ agree on $(A \cup B) \cap (A' \cup B')$.*
3. *There exists a distance function $d^*$ over $X = A \cup B \cup A' \cup B'$ so that $F(A \cup B \cup A' \cup B', d^*) = (T, \eta)$ where*
   (a) *$d^*$ extends both $d$ and $d'$,*
   (b) *There exist $(x, y), (x, z) \in E(T)$ such that $\mathcal{C}(x) = A \cup B$, and $\mathcal{C}(y) = A$, and $\mathcal{C}(z) = B$,*
   (c) *There exist $(x', y'), (x', z') \in E(T)$ such that $\mathcal{C}(x') = A' \cup B'$, and $\mathcal{C}(y') = A'$, and $\mathcal{C}(z') = B'$, and*
   (d) *$\eta(x) = \eta(x')$*

$(A, B, d)$ is *comparable* with $(C, D, d')$ if they are $<_F$ comparable or $(A, B, d) \cong_F (C, D, d')$.

Whenever two triples are $F$-equivalent, then they have the same $<_F$ or $\cong_F$ relationship with all other triples.

**Lemma 28.** *Given a local, outer-consistent hierarchical function $F$, if $(A, B, d_1) \cong_F (C, D, d_2)$, then for any $(E, F, d_3)$, if $(E, F, d_3)$ is comparable with both $(A, B, d_1)$ and $(C, D, d_2)$ then*

- *if $(A, B, d_1) \cong_F (E, F, d_3)$ then $(C, D, d_2) \cong_F (E, F, d_3)$*
- *if $(A, B, d_1) <_F (E, F, d_3)$ then $(C, D, d_2) <_F (E, F, d_3)$*

*Proof.* Let $X = A \cup B \cup C \cup D \cup E \cup F$. By richness (condition 3 of Definition 40), there exists a distance function $d$ that extends $d_i$ for $i \in \{1, 2, 3\}$ so that $\{A \cup B, C \cup D, E \cup F\}$ is a clustering in $F(X, d)$. Assume that $(E, F, d_3)$ is comparable with both $(A, B, d_1)$ and $(C, D, d_2)$. By way of contradiction, assume that $(A, B, d_1) \cong_F (E, F, d_3)$ and $(C, D, 2_1) <_F (E, F, d_3)$. Then by locality, in $F(X, d)$, $\eta(v(A \cup B)) = \eta(v(E \cup F))$.

Observe that by locality, since $(C, D, d_1) <_F (E, F, d_3)$, then $\eta(v(C \cup D)) < \eta(v(E \cup F))$ in $F(X, d)$. Therefore (again by locality) $\eta(v(A \cup B)) \neq \eta(v(C \cup D))$ in any data set that extends $d_1$ and $d_2$, contradicting that $(A, B, d_1) \cong_F (C, D, d_2)$. □

Note that $<_F$ is not transitive. In particular, if $(A, B, d_1) <_F (C, D, d_2)$ and $(C, D, d_2) <_F (E, F, d_3)$, it may be that $(A, B, d_1)$ and $(E, F, d_3)$ are incomparable. To show that $<_F$ can be extended to a partial ordering, we first prove the following "anti-cycle" property.

**Lemma 29.** *Given a hierarchical function $F$ that is local and outer-consistent, there exists no finite sequence $(A_1, B_1, d_1) <_F \cdots <_F (A_n, B_n, d_n) <_F (A_1, B_1, d_1)$.*

*Proof.* Without loss of generality, assume that such a sequence exists. By richness, there exists a distance function $d$ that extends each of the $d_i$ where $\{A_1 \cup B_1, A_1 \cup B_2, \ldots, A_n \cup B_n\}$ is a clustering in $F(\bigcup_i A_i \cup B_i, d) = (T, M, \eta)$.

Let $i_0$ be so that $\eta(v(A_{i_0} \cup B_{i_0}) \leq \eta(v(A_j \cup B_j))$ for all $j \neq i_0$. By the circular structure with respect to $<_F$, there exists $j_0$ so that $(A_{j_0}, B_{j_0}, d_{j_0}) <_F (A_{i_0}, B_{i_0}, d_{i_0})$. This contradicts Lemma 27. □

We make use of the following general result, proved in Chapter 4.

**Lemma 30.** *For any cycle-free, anti-symmetric relation $P(\ ,\ )$ over a finite or countable domain $D$ there exists an embedding $h$ into $\mathbb{R}^+$ so that for all $x, y \in D$, if $P(x, y)$ then $h(x) < h(y)$.*

Finally, we define our linkage function by embedding the $\cong_F$-equivalence classes into the positive real numbers in an order preserving way, as implied by applying Lemma 30 to $<_F$. Namely, $\ell_F : \{[(A, B, d)] : A \subseteq \mathcal{N}, B \subseteq \mathcal{N}, A \cap B = \emptyset \textit{ and } d \textit{ is a distance function over } A \cup B\} \to \mathbb{R}^+$ so that $[(A, B, d)] <_F [(A', B', d')]$ implies $\ell_F[(A, B, d)] < \ell_F[(A, B, d)]$.

**Lemma 31.** *The function $\ell_F$ is a linkage function for any hierarchical function $F$ that satisfies locality and outer-consistency.*

*Proof.* Since $\ell_F$ is defined on $\cong_F$-equivalence classes, representation independence of hierarchical functions implies that $\ell_F$ satisfies condition 1 of Definition 41. The function $\ell_F$ satisfies condition 2 of Definition 41 by Lemma 32, whose proof follows. □

**Lemma 32.** *Consider $d_1$ over $X_1 \cup X_2$ and $d_2$ that is $(\{X_1, X_2\}, d_1)$-outer-consistent, then $(X_1, X_2, d_2) \not<_F (X_1, X_2, d_1)$, whenever $F$ is local and outer-consistent.*

*Proof.* Assume that there exist such $d_1$ and $d_2$ where $(X_1, X_2, d_2) <_F (X_1, X_2, d_1)$. Let $d_3$ over $X_1 \cup X_2$ be a distance function such that $d_3$ is $(\{X_1, X_2\}, d_1)$-outer-consistent and $d_2$ is $(\{X_1, X_2\}, d_3)$-outer-consistent. In particular, $d_3$ can be constructed as follows:

- $d_3(x, y) = \frac{d_1(x,y)+d_2(x,y)}{2}$ whenever $x \in X_1$ and $y \in X_2$
- $d_3(x, y) = d_1(x, y)$ whenever $x, y \in X_1$ or $x, y \in X_2$

Set $(X_1', X_2', d_2) = (X_1, X_2, d_2)$ and $(X_1'', X_2'', d_3) = (X_1, X_2, d_3)$.

Let $X = X_1 \cup X_2 \cup X_1' \cup X_2' \cup X_1'' \cup X_2''$. By richness, there exists a distance function $d^*$ that extends $d_i$ for all $1 \leq i \leq 3$ so that $\{X_1 \cup X_2, X_1' \cup X_2', X_1'' \cup X_2''\}$ is a clustering in $F(X, d^*)$.

Let $F(X, d^*) = (T, M, \eta)$. Since $(X_1', X_2', d_2) <_F (X_1, X_2, d_1)$, by locality and outer-consistency, we get that $\eta(v(X_1' \cup X_2')) < \eta(v(X_1 \cup X_2))$. We consider the level ($\eta$ value) of $v(X_1'' \cup X_2'')$ with respect to the levels of $v(X_1' \cup X_2')$ and $v(X_1 \cup X_2)$ in $F(X, d^*)$.

We now consider a few cases.

Case 1: $\eta(v(X_1'' \cup X_2'')) \leq \eta(v(X_1' \cup X_2'))$. Then there exists an outer-consistent change moving $X_1$ and $X_2$ further away from each other until $(X_1, X_2, d_1) = (X_1'', X_2'', d_3)$. Let $\hat{d}$ be the distance function that extends $d_1$ and $d_2$ which shows that $(X_1', X_2', d_2) <_F (X_1, X_2, d_1)$. $cut_{X_1' \cup X_2'} F(X_1 \cup X_2 \cup X_1' \cup X_2', \hat{d}) = \{X_1' \cup X_2', X_1, X_2\}$. We can apply outer consistency on $\{X_1' \cup X_2', X_1, X_2\}$ and move $X_1$ and $X_2$ away from each other until $\{X_1, X_2\}$ is isomorphic to $\{X_1'', X_2''\}$. By outer consistency, this modification should not effect the $(X_1 \cup X_2)$-cut. Applying locality, we have two isomorphic data sets that produce different dendrogram, one in which the further pair ($d_2$) not below the medium pair ($d_3$), and the other in which the medium pair (turning $d_3$ into $d_2$) is above the furthest pair.

Case 2: $\eta(v(X_1'' \cup X_2'')) \geq \eta(v(X_1 \cup X_2))$. Since $X_i''$ is isomorphic to $X_i$ for all $i \in \{1, 2\}$, $\eta(v(X_i)) = \eta(v(X_i''))$ for all $i \in \{1, 2\}$. This gives us that in this case, $cut_{X_1 \cup X_2} F(X_1 \cup X_2 \cup X_1'' \cup X_2'', d^*) = \{X_1 \cup X_2, X_1'', X_2''\}$. We can therefore apply outer consistency and separate $X_1''$ and $X_2''$ until $\{X_1'', X_2''\}$ is isomorphic to $\{X_1' \cup X_2'\}$. So this gives us two isomorphic data sets, one which the further pair is not below the closest pair, and the other in which the further pair is below the closest pair.

Case 3: $\eta(X_1 \cup X_2) < \eta(X_1'' \cup X_2'') < \eta(X_1' \cup X_2')$. Notice that $cut_{X_1'' \cup X_2''} F(X_1 \cup X_2 \cup X_1'' \cup X_2'', d^*) = \{X_1'' \cup X_2'', X_1, X_2\}$. So outer-consistency applies when we increase the distance between $X_1$ and $X_2$ until $\{X_1, X_2\}$ is isomorphic to $\{X_1' \cup X_2'\}$. This gives us two isomorphic sets, one in which the medium pair is below the further pair, and another in which the medium pair is above the furthest pair. □

The following Lemma concludes the proof that every local, outer-consistent hierarchical algorithm is linkage-based.

**Lemma 33.** *Given any hierarchical function $F$ that satisfies locality and outer-consistency, let $\ell_F$ be the linkage function defined above. Let $L_{\ell_F}$ denote the linkage-based algorithm that $\ell_F$ defines. Then $L_{\ell_F}$ agrees with $F$ on every input data set.*

*Proof.* Let $(X, d)$ be any data set. We prove that at every level $s$, the nodes at level $s$ in $F(X, d)$ represent the same clusters as the nodes at level $s$ in $L_{\ell_F}(X, d)$. In both $F(X, d) = (T, M, \eta)$ and $L_{\ell_F}(X, d) = (T', M', \eta')$, level 0 consists of $|X|$ nodes each representing a unique elements of $X$.

Assume the result holds below level $k$. We show that pairs of nodes that do not have parents below level $k$ have minimal $\ell_F$ value only if they are merged at level $k$ in $F(X, d)$.

Consider $F(X, d)$ at level $k$. Since the dendrogram has no empty levels, let $x \in V(T)$ where $\eta(x) = k$. Let $x_1$ and $x_2$ be the children of $x$ in $F(X, d)$. Since $\eta(x_1), \eta(x_2) < k$, these nodes also appear in $L_{\ell_F}(X, d)$ below level $k$, and neither node has a parent below level $k$.

If $x$ is the only node in $F(X, d)$ above level $k-1$, then it must also occur in $L_{\ell_F}(X, d)$. Otherwise, there exists a node $y_1 \in V(T)$, $y_1 \notin \{x_1, x_2\}$ so that $\eta(y_1) < k$ and $\eta(\text{parent}(y_1)) \geq k$. Let $X' = \mathcal{C}(x) \cup \mathcal{C}(y_1)$. By locality, $cut_{\mathcal{C}(x)} F(X', d|X') = \{\mathcal{C}(x), \mathcal{C}(y_1)\}$, $y_1$ is below $x$, and $x_1$ and $x_2$ are the children of $x$. Therefore, $(\mathcal{C}(x_1), \mathcal{C}(x_2), d) <_F (\mathcal{C}(x_1), \mathcal{C}(y_1), d)$ and $\ell_F(\mathcal{C}(x_1), \mathcal{C}(x_2), d) < \ell_F(\mathcal{C}(x_1), \mathcal{C}(y_1), d)$.

Assume that there exists $y_2 \in V(T)$, $y_2 \notin \{x_1, x_2, y_1\}$ so that $\eta(y_2) < k$ and $\eta(\text{parent}(y_2)) \geq k$. If $\text{parent}(y_1) = \text{parent}(y_2)$ and $\eta(\text{parent}(y_1)) = k$, then $(\mathcal{C}(x_1), \mathcal{C}(x_2), d) \cong_F (\mathcal{C}(y_1), \mathcal{C}(y_2), d)$ and so $\ell_F(\mathcal{C}(x_1), \mathcal{C}(x_2), d) = \ell_F(\mathcal{C}(y_1), \mathcal{C}(y_2), d)$.

Otherwise, let $X' = \mathcal{C}(x) \cup \mathcal{C}(y_1) \cup \mathcal{C}(y_2)$. By richness, there exists a distance function $d^*$ that extends $d|\mathcal{C}(x)$ and $d|(\mathcal{C}(y_1) \cup \mathcal{C}(y_1))$, so that $\{\mathcal{C}(x), \mathcal{C}(y_1) \cup \mathcal{C}(y_2)\}$ is in $F(X', d^*)$. Note that by locality, the node $v(\mathcal{C}(y_1) \cup \mathcal{C}(y_2))$ has children $v(\mathcal{C}(y_1))$ and $v(\mathcal{C}(y_2))$ in $F(X', d^*)$. We can separate $\mathcal{C}(x)$ from $\mathcal{C}(y_1) \cup \mathcal{C}(y_2)$ in both $F(X', d^*)$ and $F(X', d|X')$ until both are equal. Then by outer-consistency, $cut_{\mathcal{C}(x)} F(X', d|X') = \{\mathcal{C}(x), \mathcal{C}(y_1), \mathcal{C}(y_2)\}$ and by locality $y_1$ and $y_2$ are below $x$. Therefore, $(\mathcal{C}(x_1), \mathcal{C}(x_2), d) <_F (\mathcal{C}(y_1), \mathcal{C}(y_2), d)$ and so $\ell_F(\mathcal{C}(x_1), \mathcal{C}(x_2), d) < \ell_F(\mathcal{C}(y_1), \mathcal{C}(y_2), d)$. □

□

### 7.3.2 All Linkage-Based Functions are Local and Outer-Consistent

**Lemma 34.** *Every linkage-based hierarchical clustering function is local.*

*Proof.* Let $C = \{C_1, C_2, \ldots, C_k\}$ be a clustering in $F(X, d) = (T, M, \eta)$. Let $X' = \cup_i C_i$. For all $X_1, X_2 \in X'$, $\ell(X_1, X_2, d) = \ell(X_1, X_2, d|X')$. Therefore, for all $1 \leq i \leq k$, the sub-dendrogram rooted at $v(C_i)$ in $F(X, d)$ also appears in $F(X, d')$, with the same relative levels. □

**Lemma 35.** *Every linkage-based hierarchical clustering function is outer-consistent.*

*Proof.* Let $C = \{C_1, C_2, \ldots, C_k\}$ be a $C_i$-cut in $F(X, d)$ for some $1 \leq i \leq k$. Let $d'$ be $(C, d)$-outer-consistent. Then for all $1 \leq i \leq k$, and all $X_1, X_2 \subseteq C_i$, $\ell(X_1, X_2, d) = \ell(X_1, X_2, d')$, while for all $X_1 \subseteq C_i$, $X_2 \subseteq C_j$, for any $i \neq j$, $\ell(X_1, X_2, d) \leq \ell(X_1, X_2, d')$ by monotonicity. Therefore, for all $1 \leq j \leq k$, the sub-dendrogram rooted at $v(C_j)$ in $F(X, d)$ also appears in $F(X, d')$. All nodes added after these sub-dendrograms are at a higher level than the level of $v(C_i)$. And since the $C_i$-cut is represented by nodes that occur on levels no higher than the level of $v(C_i)$, the $C_i$-cut in $F(X, d')$ is the same as the $C_i$-cut in $F(X, d)$. □

### 7.3.3 Necessity of Both Properties

We now show that both the locality and outer-consistency properties are necessary for defining linkage-based algorithms. Neither property individually is sufficient for defining this family of algorithms. Our results above showing that all linkage-based algorithms are both local and outer-consistent already imply that a clustering function that satisfies one, but not both, of these requirements is not linkage-based. It remains to show that neither of these two properties implies the other. We do so by demonstrating the existence of a hierarchical function that satisfies locality but not outer-consistency, and one that satisfy outer-consistency but not locality.

Consider a hierarchical clustering function $F$ that applies average-linkage on data sets with an even number of elements, and single-linkage on data sets consisting of an odd number of elements. Since both average-linkage and single-linkage are linkage-based algorithms, they are both outer-consistent. It follows that $F$ is outer-consistent. However, this hierarchical clustering function fails locality, as it is easy to construct a data set with an even number of elements where average-linkage detects an odd-sized cluster, for which single-linkage would produce a different dendrogram.

Now, consider the following function

$$\ell(X_1, X_2, d) = \frac{1}{\max_{x \in X_1, y \in X_2} d(x, y)}.$$

The function $\ell$ is not a linkage-function since it fails the monotonicity condition. The function $\ell$ also does not conform with the intended meaning of a linkage-function. For instance, $\ell(X_1, X_2, d)$ is smaller than $\ell(X_1', X_2', d')$ when *all* the distances between $X_1$ and $X_2$ are (arbitrarily) larger than any distance between $X_1'$ and $X_2'$. If we then consider the hierarchical clustering function $F$ that results by utilizing $\ell$ in a greedy fashion to construct a dendrogram (by repeatedly merging the closest clusters according to $\ell$), then the function $F$ is local by the same argument as the proof of Lemma 34. We now demonstrate that $F$ is not outer-consistent. Consider a data set $(X, d)$ such that for some $A \subset X$, the $A$-cut of $F(X, d)$ is a clustering with a least 3 clusters where every cluster consists of a least 2 elements. Then if we move two clusters sufficiently far away from each other and all other data, they will be merged by the algorithm before any of the other clusters are formed, and so the $A$-cut on the resulting data changes following an outer-consistent change. As such, $F$ is not outer-consistent.

## 7.4 Divisive Algorithms

Our formalism provides a precise sense in which linkage-based algorithms make only local considerations, while many divisive algorithms inevitably take more global considerations into account. This fundamental distinction between these paradigms can be used to help select a suitable hierarchical algorithm for specific applications.

This distinction also implies that many divisive algorithms cannot be simulated by any linkage-based algorithm, showing that the class of hierarchical algorithms is strictly richer than the class of linkage-based algorithm (even when focusing only on the input-output behaviour of algorithms).

A 2-clustering function $\mathcal{F}$ maps a data set $(X, d)$ to a 2-partition of $X$. An $\mathcal{F}$-Divisive algorithm is a divisive algorithm that uses a 2-clustering function $\mathcal{F}$ to decide how to split nodes. Formally,

**Definition 48** ($\mathcal{F}$-Divisive)**.** *A hierarchical clustering function is $\mathcal{F}$-Divisive with respect to a 2-clustering function $\mathcal{F}$, if for all $(X, d)$, $\mathcal{F}(X, d) = (T, M, \eta)$ such that for all $x \in V(T)/\text{leaves}(T)$ with children $x_1$ and $x_2$, $\mathcal{F}(\mathcal{C}(x)) = \{\mathcal{C}(x_1), \mathcal{C}(x_2)\}$.*

Note that Definition 48 does not place restrictions on the level function. This allows for some flexibility in the levels. Intuitively, it doesn't force an order on splitting nodes.

The following property represents clustering functions that utilize contextual information found in the remainder of the data set when partitioning a subset of the domain.

**Definition 49** (Context sensitive)**.** *$\mathcal{F}$ is* context-sensitive *if there exist distance functions $d$ and $d'$, where $d'$ extends $d$, such that $\mathcal{F}(\{x, y, z\}, d) = \{\{x\}, \{y, z\}\}$ and $\mathcal{F}(\{x, y, z, w\}, d') = \{\{x, y\}, \{z, w\}\}$.*

Many 2-clustering functions, including $k$-means, min-sum, and min-diameter are context-sensitive (see Corollary 3, below). Natural divisive algorithms, such as bisecting $k$-means ($k$-means-Divisive), rely on context-sensitive 2-clustering functions.

Whenever a 2-clustering algorithm is context-sensitive, then the $\mathcal{F}$-divisive function is not local.

**Theorem 29.** *If $\mathcal{F}$ is context-sensitive then the $\mathcal{F}$-divisive function is not local.*

*Proof.* Since $\mathcal{F}$ is context-sensitive, there exists a distance functions $d \subset d'$ so that $\{x\}$ and $\{y, z\}$ are the children of the root in $\mathcal{F}(\{x, y, z\}, d)$, while in $\mathcal{F}(\{x, y, z, w\}, d')$, $\{x, y\}$ and $\{z, w\}$ are the children of the root and $z$ and $w$ are the children of $\{z, w\}$. Therefore, $\{\{x, y\}, \{z\}\}$ is clustering in $\mathcal{F}(\{x, y, z, w\}, d')$. But cluster $\{x, y\}$ is not in $\mathcal{F}(\{x, y, z\}, d)$, so the clustering $\{\{x, y\}, \{z\}\}$ is not in $\mathcal{F}(\{x, y, z\}, d)$, and so $\mathcal{F}$-divisive is not local. □

Applying Theorem 28, we get:

**Corollary 2.** *If $\mathcal{F}$ is context-sensitive, then the $\mathcal{F}$-divisive function is not linkage-based.*

We say that two hierarchical algorithms *strongly disagree* if they may output dendrograms with different clusterings. Formally,

**Definition 50.** *Two hierarchical functions $F_0$ and $F_1$* strongly disagree *if there exists a data set $(X, d)$ and a clustering $C$ of $X$ so that $C$ is in $F_i(X, d)$ but not in $F_{1-i}(X, d)$, for some $i \in \{0, 1\}$.*

**Theorem 30.** *If $\mathcal{F}$ is context-sensitive, then the $\mathcal{F}$-divisive function strongly disagrees with every linkage-based function.*

*Proof.* Let $L$ be any linkage-based function. Since $\mathcal{F}$ is context-sensitive, there exists distance functions $d \subset d'$ so that $\mathcal{F}(\{x, y, z\}, d) = \{\{x\}, \{y, z\}\}$ and $\mathcal{F}(\{x, y, z, w\}, d') = \{\{x, y\}, \{z, w\}\}$.

Assume that $L$ and $\mathcal{F}$-*divisive* produce the same output on $(\{x, y, z, w\}, d')$. Therefore, since $\{\{x, y\}, \{z\}\}$ is a clustering in $\mathcal{F}$-*divisive*$(\{x, y, z, w\}, d')$, it is also a clustering in $L(\{x, y, z, w\}, d')$. Since $L$ is linkage-based, by Theorem 28, $L$ is local. Therefore, $\{\{x, y\}, \{z\}\}$ is a clustering in $L(\{x, y, z\}, d')$. But it is not a clustering in $\mathcal{F}$-*divisive*$(\{x, y, z\}, d)$. □

**Corollary 3.** *The divisive algorithms that are based on the following 2-clustering functions strongly disagree with every linkage-based function: k-means, min-sum, min-diameter.*

*Proof.* Set $x = 1$, $y = 3$, $z = 4$, and $w = 6$ to show that these 2-clustering functions are context-sensitive. The result follows by Theorem 30. □

# CHAPTER 8

# CONCLUSIONS AND FUTURE WORK

## 8.1 Summary

Due to the ambiguous nature of clustering, its users have varied needs. No one algorithm fits all clustering applications. In this thesis, we develop a theoretically founded approach for selecting clustering algorithms based on differences in their input-output behaviour. To this end, we strive for a better understanding of how clustering algorithms differ. An understanding into core differences in the input-output behaviour of common clustering techniques makes it possible to make an informed choice when selecting an algorithm. In order to make the theory usable in practice, we formulate these difference in terms of concise, and mathematically precise, properties. A classification of clustering algorithms based on these properties can then be utilized by any clustering user to assist in the algorithm selection process.

By proposing new properties and using those proposed in previous work, we present a property-based classification of some common clustering methods. While this initial classification highlights differences among different clustering paradigms, it does not explain the popularity of the $k$-means method. To this end, we study the behaviour of clustering algorithms under the addition of small sets to the original data. We show that $k$-means is robust to the addition of a small number of elements, even when those are chosen in adversarial manner. On the other hand, the output of many other common clustering methods is highly volatile to the addition of few data points.

Our study of clustering properties leads to the first property-based characterization of linkage-based clustering. This characterization can be viewed as an alternative definition of this family of algorithms to the typical definition that relies on pseudo-code. By defining linkage-based algorithms based on their input-output behaviour, our definition enables a direct comparison with the behaviour of other clustering methods. We provide a property-based characterization of this family of algorithms in both the partitional and hierarchical settings.

In this thesis, we provide a foundation for an approach to selecting clustering algorithms based on differences in their input-output behaviour. It is not meant as a deliverable tool. Yet by continuing to improve our understanding of significant differences between clustering methods, we become better equipped to assist users in selecting an algorithm for a wider range of applications.

## 8.2 Previous Work Revisited

Before we wrap up, let us revisit previous work, and discuss how it connects with our results. In Figure 8.1, we present a diagram that illustrates how our work fits within the clustering literature. Namely, we display various branches of research on clustering properties. It is not exhaustive, and we have expanded only those branches that are most relevant to our contributions. We also note that, since there are many difference levels on which research papers related to one another, this diagram could have been structured in several other meaningful ways. This diagram helps illustrate only of the some ways that the contributions in the current thesis fit into clustering as a field. Lastly, for completeness, we also include in this diagram some of our own work that has not been included in this thesis.

This diagram partitions research on clustering properties based on the objects studied. Many different clustering objects have been considered, but with the exception of Chapter 3.2.2 that also clustering quality measures, this thesis is primary concerned with clustering functions. Another interesting object, on which we did not focus on in this thesis, is that of clustering distance functions, studied by Meila [39]. In addition, the study of data set clusterability, the degree of clustered structure inherent in data, has been explored by Balcan and Blum ([12], [11], and [13]) in the context of computational complexity, addressing the question: "If there is a unique desired clustering, what do we need to know about it so that clustering becomes computationally efficient?" There is also work on clusterability by Ackerman and Ben-David [1] not included in this thesis, where we compare different notions of clusterability proposed in the literature, and show that although all of these notions aim to evaluate the same intuitive property, they are provably pairwise distinct; for every pair of these notions, there is a data set that is arbitrarily well clusterable according to one of the notions, and arbitrarily poorly clusterable according to the other.

Properties of clustering objects can sometimes be converted from one context to another making it so that research on different clustering objects is often closely related. For instance, Puzicha, Hofmann, and Buhmann [43], propose several properties of clustering functions in the setting where the number of clusters, $k$, is fixed. In particular, they introduce several properties including scale-invariance, representation independence (called "permutation invariance" in their paper), and consistency (referred to as "monotonicity"). They then focus on functions that can be decomposed into a specific additive form. There are interesting connections between the work of Puzhica et al. and Kleinberg's impossibility result. Kleinberg [35] considers a slightly different setting, where the algorithm has to decide into how many clusters to partition the data. He then converts scale-invariance and consistency into the framework where the number of clusters is not fixed, and adds the richness property, which relies on $k$ not being fixed. He shows that the three properties cannot be simultaneously satisfied by the same clustering function. In Chapter 3.2.2, we translate Kleinberg's axioms to the setting of clustering quality measures, where the three properties become consistent. Translating Kleinberg's axioms into the setting of clustering functions where the number of clusters is fixed also leads to consistency of the three axioms. This illustrates that different clustering settings are related to each other, but also, that representing our intuition about clustering in different settings can lead to vastly different results.

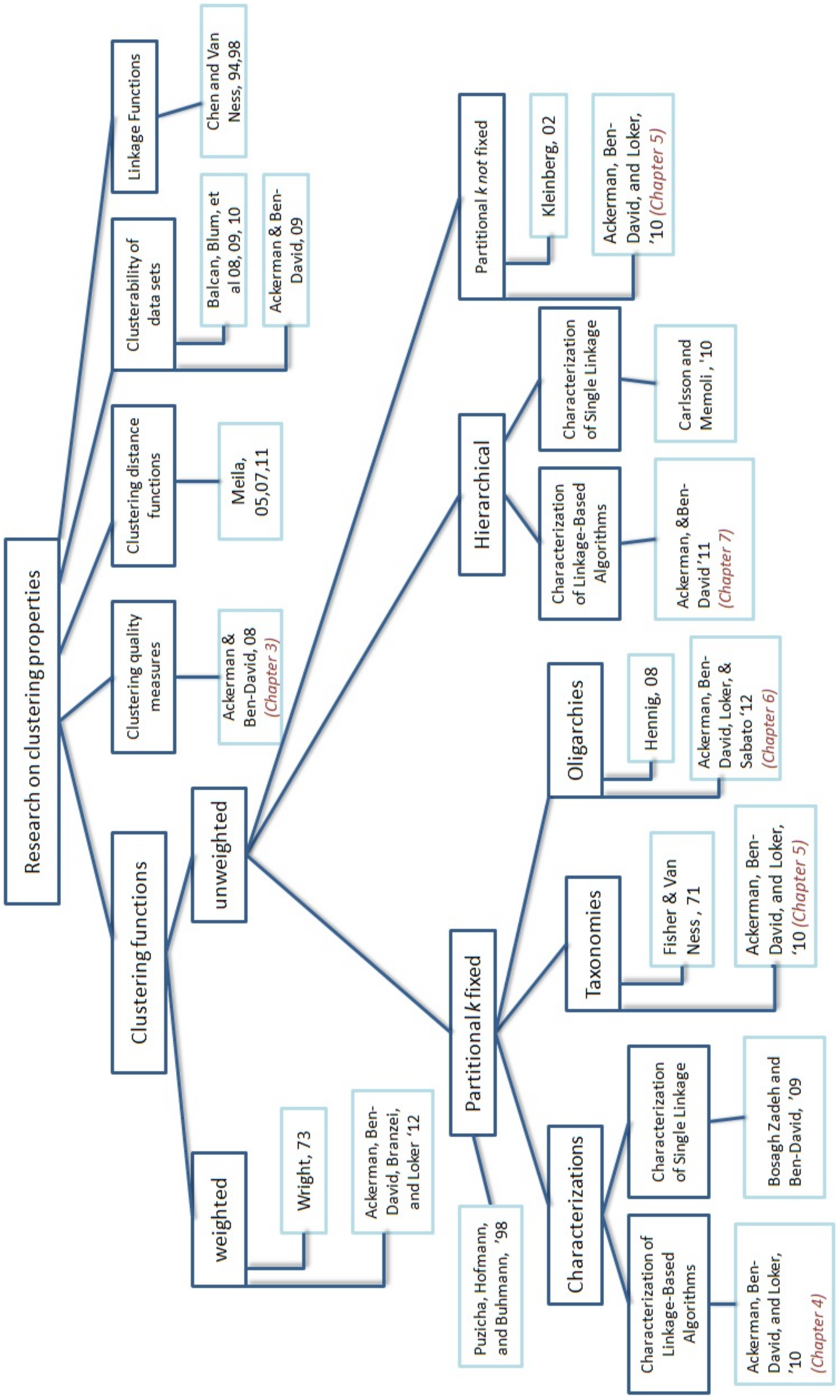

Research on clustering properties
Linkage Functions
Chen and Van Ness, 94,98
Clusterability of data sets
Balcan, Blum, et al 08, 09, 10
Ackerman & Ben-David, 09
Clustering distance functions
Meila, 05,07,11
Clustering quality measures
Ackerman & Ben-David, 08 (Chapter 3)
Clustering functions
unweighted
weighted
Wright, 73
Ackerman, Ben-David, Branzei, and Loker '12
Partitional k not fixed
Kleinberg, 02
Ackerman, Ben-David, and Loker, '10 (Chapter 5)
Hierarchical
Characterization of Single Linkage
Carlsson and Memoli, '10
Characterization of Linkage-Based Algorithms
Ackerman, &Ben-David '11 (Chapter 7)
Partitional k fixed
Puzicha, Hofmann, and Buhmann, '98
Oligarchies
Hennig, 08
Ackerman, Ben-David, Loker, & Sabato '12 (Chapter 6)
Taxonomies
Fisher & Van Ness , 71
Ackerman, Ben-David, and Loker, '10 (Chapter 5)
Characterizations
Characterization of Single Linkage
Bosagh Zadeh and Ben-David, '09
Characterization of Linkage-Based Algorithms
Ackerman, Ben-David, and Loker, '10 (Chapter 4)

Figure 8.1: Research on clustering properties, organized by object studied.

If we focus on clustering functions, one basic differentiation is between the weighted and unweighted settings. This thesis is concerned with the unweighted paradigm. In the weighted clustering setting, we have an additional source of information; every point is assigned a real valued weight. The weighted model was used to study clustering since the early work of Wright [47] in 1973. Very recently, Ackerman, Ben-David, Branzei, and Loker [3] revisited this model and proposed properties within it that can be used to differential between the behaviour of clustering algorithms.

When considering the unweighted clustering functions, we study three frameworks; (1) partitional clustering functions where the input includes the number of clusters, $k$, as well as a domain with a distance function over it, (2) partitional clustering functions whose only input is a domain endowed with a disatnce function, and (3) a hierarchical clustering setting.

Kleinberg's impossibility result [35] was proved for partitional clustering functions where the number of clusters is not specified. Our only results in this setting are extensions of his impossibility result obtained by relaxing one the original properties (consistency), these results appear in Chapter 5.2.

It appears that the framework in which the number of clusters is specified is richer and more flexible. When translated into this setting, the properties in Kleinberg's impossibility result become consistent. We provide a property-based classification of such algorithms in Chapter 5.2.

A property-based classification of some common clustering methods was also presented in a 1971 paper by Fisher and Van Ness [27]. Although we classify their work under the partitional clustering functions with a fixed number of clusters, they actually consider several clustering objects within the same taxonomy. In particular, their taxonomy addressed both partitional and hierarchical methods, and one of the properties falls under weighted clustering. Just like in our taxonomy, the purpose of their property-based classification is to aid users in selecting a clustering algorithm. In their paper, they survey properties of clustering algorithms and for each algorithm considered, they prove whether each property is satisfied. They consider five algorithms, including single-linkage, complete-linkage, and $k$-means. A total of nine properties are used to evaluate the algorithms, two of which only apply in Euclidean space. The property that falls under the weighted clustering setting requires that the clustering function output not change if data weights are modified. Several of their other properties are variations on the $\alpha$-separability-detecting condition, discussed in Chapter 6.

The only follow up work that we are aware of to the 1971 paper of Fisher and Van Ness [27] is by Chen and Van Ness ([19, 18, 20]), which focuses on properties of linkage functions (used to drive linkage-based algorithms) instead of properties of clustering functions. In Figure 8.1, which is organized by object studied, the work of Chen and Van Ness falls under the linkage-functions category. However, since linkage-functions are used to formulate linkage-based clustering functions, the work of Fisher and Van Ness can help differentiate between different linkage-based algorithms. Note that in our work on linkage-based algorithms, in particular our characterization of this class, we rely on properties of clustering functions.

In additional to our characterization of linkage-based algorithms, there is a characterization of a specific linkage-based algorithm, namely, single-linkage, in terms of properties of clustering functions. That result by Bosagh Zadeh and Ben-David [48] uses consistency, richness, order invariance, and another property by the name of *MST coherence* to characterize the single-linkage algorithm. MST coherence requires that the output of an algorithm should be the same whenever the graphs corresponding to the input distance functions have identical minimum spanning trees.

Under partitional clustering methods with a fixed number of clusters, we also display our work on clustering oligarchies, where we study how algorithms respond to the addition of a small number of points. There we also include previous work by Hennig [30], where the diameter of the data is not bounded, and so different results are obtained. At the end of Chapter 6, we also discuss how work on planted partitions on random graphs is related to our work on clustering oligarchies.

Finally, looking at the hierarchical clustering setting, there we also provide a characterization of linkage-based clustering. In addition, Carlsson and Memoli [17] provide a characterization of single-linkage in this setting. It is interesting to note that while our characterizations of the linkage-based family of algorithms are fundamentally similar to each other, the characterizations of the single-linkage in the partitional clustering setting by Bosagh Zadeh and Ben-David [48] is substantially different from that of Carlsson and Memoli [17] in the hierarchical setting. Finally, it is curious why of all algorithms, the family of linkage-based algorithms has been the focus of property-base characterizations.

Now that we have discussed how our work connects with current literature, we will conclude by proposing avenues of investigation for future work.

## 8.3 Future Directions

There are many interesting avenues for future investigation. Since our framework for selecting a clustering algorithm is compatible with any clustering application, it would be interesting to explore what properties are desirable for specific applications. One common application of hierarchical clustering is Phylogeny, which aims to reconstruct the tree of life. We began exploring some properties that are prevalent in this field [7], and showed which algorithms satisfy, and which fail these properties. In addition to continuing exploration within the field of Phylogeny, it would interesting to explore other applications, such as document clustering, marketing, and city planing. Applications will differ on their desirable properties, and we could then go ever further. We could classify clustering applications based on their clustering needs, which could act as a short-cut for new clustering users. But also, this could be used for focusing research efforts on developing algorithms that possess properties that are desirable across many common applications.

In addition, it would be interesting to continue exploring the advantages of common clustering methods, such as k-means and corresponding heuristics. Studying properties of algorithms that prove to be successful in some domain can lead to the discovery of other important properties. Further, we could focus on a group of similar algorithms, such as k-means, k-median, and k-medoids, and study differences among them.

It is also important to explore additional clustering frameworks. In this thesis, we have looked at a general partitional clustering setting, as well as a hierarchical one. As clustering is a highly versatile domain, there are many interesting and useful clustering settings where our framework for selecting clustering algorithms can be used. In particular, it would be interesting to investigate properties of clustering algorithms in the setting where a noise bucket is allowed; namely, one of the clusters is reserved for collecting points that do not fit well into any other cluster. Another interesting framework is that of fuzzy clustering, where every elements is assigned values indicating how well it fits within every cluster. Lastly, since data is often categorical, it is also worth investigating properties of clustering algorithms designed for categorical data.

One of the most fundamental open problems is that of axioms of clustering functions. We touch on this subject in Chapter 5, where we provide three potential axioms of clustering, which may be necessary, but are not sufficient to define clustering. In Chapter 3, we also proposed a set of consistent axioms of clustering quality measures. But when converted to the setting of clustering functions, these axioms become inconsistent, as shown by Kleinberg [35]. Finding a consistent set of axioms of clustering function is still open. By investigating many different clustering objectives, we as a community have already identified many of the important facets of what clustering is. Perhaps all that is left is to synthesize our collective insights into a set of axioms. A consistent set of axioms of clustering functions would be great step forward in the theory of clustering, and it may be within our grasp.

# BIBLIOGRAPHY

[1] M. Ackerman and S. Ben-David. Clusterability: A theoretical study. *Proceedings of AISTATS-09, JMLR: W&CP*, 5(1-8):53, 2009.

[2] M. Ackerman and S. Ben-David. Machine learning-discerning linkage-based algorithms among hierarchical clustering methods. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1140, 2011.

[3] M. Ackerman, S. Ben-David, S. Branzei, and D. Loker. Weighted clustering. *Proc. 26th AAAI Conference on Artificial Intelligence*, 2012.

[4] M. Ackerman, S. Ben-David, and D. Loker. Characterization of linkage-based clustering. COLT, 2010.

[5] M. Ackerman, S. Ben-David, and D. Loker. Characterization of linkage-based clustering. COLT, 2010.

[6] M. Ackerman, S. Ben-David, and D. Loker. Towards property-based classification of clustering paradigms. NIPS, 2010.

[7] M. Ackerman, D. Brown, and D. Loker. Effects of rooting via outgroups on ingroup topology in phylogeny. 2012.

[8] P.K. Agarwal and C.M. Procopiuc. Exact and approximation algorithms for clustering. *Algorithmica*, 33(2):201–226, 2002.

[9] A. Aggarwal, H. Imai, N. Katoh, and S. Suri. Finding $k$ points with minimum diameter and related problems. *Journal of Algorithms*, 12(1):38–56, 1991.

[10] F.B. Baker and L.J. Hubert. Measuring the power of hierarchical cluster analysis. *Journal of the American Statistical Association*, pages 31–38, 1975.

[11] M.F. Balcan, A. Blum, and A. Gupta. Approximate clustering without the approximation. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1068–1077, 2009.

[12] M.F. Balcan, A. Blum, and S. Vempala. A discriminative framework for clustering via similarity functions. In *Proceedings of the 40th annual ACM symposium on Theory of Computing*, pages 671–680. ACM, 2008.

[13] M.F. Balcan and P. Gupta. Robust hierarchical clustering. In *Proceedings of the Conference on Learning Theory (COLT)*, 2010.

[14] S. Ben-David. A framework for statistical clustering with a constant time approximation algorithms for $k$-median clustering. *Learning Theory*, pages 415–426, 2004.

[15] N.H. Bshouty and P.M. Long. Finding planted partitions in nearly linear time using arrested spectral clustering. 2010.

[16] S. Bubeck, M. Meila, and U. Von Luxburg. How the initialization affects the stability of the k-means algorithm. *Arxiv preprint arXiv:0907.5494*, 2009.

[17] G. Carlsson and F. Mémoli. Characterization, stability and convergence of hierarchical clustering methods. *The Journal of Machine Learning Research*, 11:1425–1470, 2010.

[18] Z. Chen and J. Van Ness. Characterizations of nearest and farthest neighbor algorithms by clustering admissibility conditions. *Pattern Recognition*, 31(10):1573–1578, 1998.

[19] Z. Chen and J.W. Van Ness. Metric admissibility and agglomerative clustering. *Communications in Statistics-Simulation and Computation*, 23(3):833–845, 1994.

[20] Z. Chen and J.W. Van Ness. Space-contracting, space-dilating, and positive admissible clustering algorithms. *Pattern recognition*, 27(6):853–857, 1994.

[21] A. Condon and R.M. Karp. Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms*, 18(2):116–140, 2001.

[22] S. Dasgupta and L. Schulman. A probabilistic analysis of em for mixtures of separated, spherical gaussians. *The Journal of Machine Learning Research*, 8:203–226, 2007.

[23] C. Ding and X. He. Cluster aggregate inequality and multi-level hierarchical clustering. *Proceedings of Knowledge Discovery in Databases: PKDD 2005*, pages 71–83, 2005.

[24] J.C. Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4(1):95–104, 1974.

[25] S. Epter, M. Krishnamoorthy, and M. Zaki. Clusterability detection and initial seed selection in large datasets. In *The International Conference on Knowledge Discovery in Databases*, volume 7, 1999.

[26] BS Everitt, S. Landau, and M. Leese. Cluster analysis. 2001.

[27] L. Fisher and J.W. Van Ness. Admissible clustering procedures. *Biometrika*, 58(1):91–104, 1971.

[28] E.W. Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21:768–769, 1965.

[29] D. Greene, G. Cagney, N. Krogan, and P. Cunningham. Ensemble non-negative matrix factorization methods for clustering protein–protein interactions. *Bioinformatics*, 24(15):1722–1728, 2008.

[30] C. Hennig. Dissolution point and isolation robustness: robustness criteria for general cluster analysis methods. *Journal of Multivariate Analysis*, 99(6):1154–1176, 2008.

[31] D.S. Hochbaum and D.B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, pages 180–184, 1985.

[32] L.J. Hubert and J.R. Levin. A general statistical framework for assessing categorical clustering in free recall. *Psychological Bulletin*, 83(6):1072, 1976.

[33] N. Jardine and R. Sibson. Mathematical taxonomy. *London*, 1971.

[34] I. Katsavounidis, C.C. Jay Kuo, and Z. Zhang. A new initialization technique for generalized Lloyd iteration. *Signal Processing Letters*, 1(10):144–146, 1994.

[35] J. Kleinberg. An impossibility theorem for clustering. *Proceedings of International Conferences on Advances in Neural Information Processing Systems*, pages 463–470, 2003.

[36] M. Mahajan, P. Nimbhorkar, and K. Varadarajan. The planar $k$-means problem is $np$-hard. *WALCOM: Algorithms and Computation*, pages 274–285, 2009.

[37] F. McSherry. Spectral partitioning of random graphs. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 529–537. IEEE, 2001.

[38] N. Megiddo and K.J. Supowit. On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13(1):182–196, 1984.

[39] M. Meila. Comparing clusterings: an axiomatic view. In *Proceedings of the 22nd international conference on Machine learning*, pages 577–584. ACM, 2005.

[40] M. Meila. The uniqueness of a good optimum for k-means. In *Proceedings of the 23rd international conference on Machine learning*, pages 625–632. ACM, 2006.

[41] G.W. Milligan. A Monte Carlo study of thirty internal criterion measures for cluster analysis. *Psychometrika*, 46(2):187–199, 1981.

[42] R. Ostrovsky, Y. Rabani, L.J. Schulman, and C. Swamy. The effectiveness of Lloyd-type methods for the $k$-means problem. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 165–176, 2006.

[43] J. Puzicha, T. Hofmann, and J.M. Buhmann. A theory of proximity based clustering: Structure detection by optimization. *Pattern Recognition*, 33(4):617–634, 2000.

[44] L. Vendramin, R. Campello, and E.R. Hruschka. On the comparison of relative clustering validity criteria. In *Proceedings of the SIAM International Conference on Data Mining*, pages 733–744, 2009.

[45] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[46] U. Von Luxburg and S. Ben-David. Towards a statistical theory of clustering. In *PASCAL workshop on Statistics and Optimization of Clustering*, 2005.

[47] W.E. Wright. A formalization of cluster analysis. *Pattern Recognition*, 5(3):273–282, 1973.

[48] R.B. Zadeh and S. Ben-David. A uniqueness theorem for clustering. UAI, 2009.