
Characterization of Linkage-based Clustering

Margareta Ackerman Shai Ben-David David Loker

D.R.C School of Computer Science

University of Waterloo

{mackerma, shai, dloker}@cs.uwaterloo.ca

Abstract

Clustering is a central unsupervised learning task with a wide variety of applications. Not surprisingly, there exist many clustering algorithms. However, unlike classification tasks, in clustering, different algorithms may yield dramatically different outputs for the same input sets. A major challenge is to develop tools that may help select the more suitable algorithm for a given clustering task. We propose to address this problem by distilling abstract properties of clustering functions that distinguish between the types of input-output behaviors of different clustering paradigms. In this paper we make a significant step in this direction by providing such property based characterization for the class of linkage based clustering algorithms.

Linkage-based clustering is one the most commonly used and widely studied clustering paradigms. It includes popular algorithms like Single Linkage and enjoys simple efficient algorithms.

On top of their potential merits for helping users decide when are such algorithms appropriate for their data, our results can be viewed as a convincing proof of concept for the research on taxonomizing clustering paradigms by their abstract properties.

1 Introduction

Having a clustering task at hand, a user needs to choose from a wide variety of clustering algorithms that, when run on the same input data, often produce very different clusterings. In spite of the wide use of clustering in many practical applications, the practices of choosing an algorithm for a given task are completely ad hoc – currently, there exists no principled method to guide the selection of a clustering algorithm. The choice of an appropriate clustering should, of course, be task dependent. A clustering 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. A major challenge that has hardly been addressed is how to turn domain knowledge into useful guidance to the clustering algorithm designer. One approach to providing guidance to clustering users in the selection of a clustering algorithm is to identify significant properties of clustering functions that, on one hand distinguish between different clustering paradigms, and on the other hand are relevant to the domain knowledge that a user might have. Based on domain expertise users could then choose which properties they want an algorithm to satisfy, and determine which algorithms satisfy each of these properties.

Ultimately, there would be a sufficiently rich set of properties that would provide detailed, property based, taxonomy of clustering methods, that could, in turn, be used as guidelines for a wide variety of clustering users.

One of the most basic challenges along this line is to find abstract properties of clustering functions that distinguish between the major families of common clustering algorithms, such as linkage based algorithms, center based algorithms and spectral clustering algorithms. Bosagh Zadeh and Ben-David [2] made progress in this direction by providing a set of abstract properties that characterize the single linkage clustering method.

In this paper we succeed in making the next major step by distilling a set of abstract properties that distinguish between linkage based clustering to any other type of clustering paradigm. Linkage-based clusterings 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 the algorithm in that family, while 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 other clustering algorithms.

On top of the importance of understanding linkage based clustering, we hope that this analysis will serve as a proof of concept to the research direction of distilling clustering properties towards providing useful guidelines for selecting clustering algorithms in practical applications.

2 Previous Work

Our work follows a theoretical study of clustering that began with Kleinberg's impossibility result [7], in which he proposes three axioms of clustering and shows that no clustering function can simultaneously satisfy these three axioms. Ackerman and Ben-David [1] subsequently showed these axioms to be consistent in the setting of clustering quality measures. Additionally, they discuss the distinction between axioms and properties, addressing the question of when a property is an axiom. In particular, while a property may be satisfied by some, but not all, objects of a class, an axiom is a property that is satisfied by all objects of the class. In the current paper we propose variations of some of Kleinberg's axioms and use these variations in our characterization of linkage-based clustering.

There are a couple of previous characterizations of the single-linkage algorithm. In 1975, Jardine and Sibson [6] gave a characterization of single linkage. More recently, Bosagh Zadeh and Ben-David [2] characterize single-linkage within Kleinberg's framework of clustering functions. To the best of our knowledge, ours is the first characterization of a commonly used class of clustering algorithms. We note that although single-linkage is a linkage-based algorithm, our characterization doesn't build upon the previous characterizations mentioned.

3 Preliminaries and Notation

Clustering is a very wide and heterogenous domain. We choose to focus on a basic sub-domain where the (only) 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 domain. This sub-domain is rich enough to capture many of the fundamental issues of clustering, while keeping the underlying structure as succinct as possible.

A *distance function* is a symmetric function $d : X \times X \rightarrow R^+$, 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 d over X . These are the inputs for clustering functions.

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 $d' = d|_{X'}$, is defined by restricting the distance function d to $(X')^2$.

We say that a distance function d over X *extends* distance function d' over $X' \subseteq X$ if $d' \subseteq d$.

A *k-clustering* $C = \{c_1, c_2, \dots, c_k\}$ 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 $1 \leq k \leq |X|$.

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 1 (Isomorphisms between domain sets) *Two notions of isomorphism of structures are relevant to our discussion.*

1. We say that (X, d) and (X', d') are *isomorphic domains*, denoting it by $(X, d) \sim (X', d')$, if there exists a bijection $\phi : X \rightarrow X'$ so that $d(x, y) = d'(\phi(x), \phi(y))$ for all $x, y \in X$.
2. We say that two clusterings (or partitions) $C = (c_1, \dots, c_k)$ of some domain (X, d) and $C' = (c'_1, \dots, c'_k)$ of some domain (X', d') are *isomorphic clusterings*, denoted $(C, d) \cong (C', d')$, if there exists a bijection $\phi : X \rightarrow X'$ such that for all $x, y \in X$, $d(x, y) = d'(\phi(x), \phi(y))$ and, on top of that, $x \sim_C y$ if and only if $\phi(x) \sim_{C'} \phi(y)$. Note that this notion depends on both the underlying distance functions and the clusterings.

Definition 2 (Clustering functions) *A 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 $c \in R^+$ such that $d(a, b) = c \cdot d'(a, b)$ for all $a, b \in X$, then $F(X, d, k) = F(X, d', k)$.*

4 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. 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, [3] and [5]). Everitte et al. [4] call it “inter-object distance.” Common to all these formalisms is function that maps pairs of clusters to real numbers. No further detailing of the concept has been previously explored. In this paper, we zoom in on the concept of between-cluster distance and provide a rigorous, general definition.

Definition 3 (Linkage function) A linkage function is a function

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

such that,

1. ℓ is representation independent: For all (X_1, X_2) and (X'_1, X'_2) , if $(X_1, X_2, d) \cong (X'_1, X'_2, d')$ (i.e., they are clustering-isomorphic), then $\ell(X_1, X_2, d) = \ell(X'_1, X'_2, d')$.
2. ℓ 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 ℓ , 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 numbers¹.

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-points 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. Property (4) captures the intuition that by pulling two clusters away from each other they can be made arbitrarily “unlinked”.

We now define linkage-based clustering functions.

Definition 4 (linkage-based clustering function) A clustering function F is linkage-based if there exists a linkage function ℓ 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 ℓ . Formally,

$$F(X, d, k) = \{c \mid c \in F(X, d, k+1), c \neq c_i, c \neq c_j\} \cup \{c_i \cup c_j\},$$

such that $\{c_i, c_j\} = \operatorname{argmin}_{\{c_i, c_j\} \subseteq F(X, d, k+1)} \ell(c_i, c_j, d)$.

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 ℓ_{SL} , ℓ_{AL} , and ℓ_{CL} satisfy the conditions of Definition 3 and as such are linkage functions².

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

²A 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.

5 Properties of Clustering Functions

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

5.1 Hierarchical clustering

The term Hierarchical clustering is a widely used to denote clustering algorithms that operate in a "bottom up" manner, starting from singleton clusters and creating coarser and coarser clusterings by merging clusters. Sometimes, it is also used to denote the more specific family of linkage-based clustering algorithms. Here we offer a precise formalization on what makes a clustering algorithm hierarchical.

Definition 5 (Clustering Refinement) *A clustering C of X is a refinement of clustering C' of X if every cluster in C is a subset of some cluster in C' , or, equivalently, if every cluster of C' is a union of clusters of C .*

Definition 6 (Hierarchical Functions) *A clustering function is hierarchical if for every $1 \leq k \leq k' \leq |X|$, $F(X, d, k')$ is a refinement of $F(X, d, k)$.*

5.2 Locality

We now introduce a new property of clustering algorithms that we call "locality". Intuitively, a 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 7 (Locality) *A clustering function F is local if for any clustering C output by F and every subset of clusters, $C' \subseteq C$,*

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

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 $c_1, c_2, \dots, c_{k'}$ of C , and run the clustering algorithm on the points that the selected clusters consist of, namely, $c_1 \cup c_2 \cup \dots \cup c_{k'}$ asking for k' clusters. If the algorithm is local, then on the second run of the algorithm it will output $\{c_1, c_2, \dots, c_{k'}\}$.

5.3 Consistency

Consistency, introduced by Kleinberg [7], requires that the output of a clustering function, be invariant to shrinking within-cluster distances, and stretching between-cluster distances.

Definition 8 (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. $d'_X(x, y) \leq d_X(x, y)$ whenever $x \sim_C y$, and
2. $d'_X(x, y) \geq d_X(x, y)$ 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 $F(X, dk) = F(X, d', k)$.

We introduce two relaxations of consistency.

Definition 9 (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. $d'_X(x, y) = d_X(x, y)$ whenever $x \sim_C y$, and
2. $d'_X(x, y) \geq d_X(x, y)$ whenever $x \not\sim_C y$.

A 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 10 (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. $d'_X(x, y) \leq d_X(x, y)$ whenever $x \sim_C y$, and
2. $d'_X(x, y) = d_X(x, y)$ whenever $x \not\sim_C y$.

A 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.

Outer-consistency is satisfied by many common clustering functions. In Lemma 26, we will show that any linkage-based clustering function is outer-consistent. We also show below that the average-linkage and complete linkage clustering functions are not inner consistent, and therefore (since they satisfy the other two of Kleinberg's axioms, and no function satisfies all three axioms) they are not consistent.

5.4 Richness

Kleinberg introduced a Richness property as one of his axioms. A clustering function is rich if by modifying the distances any output can be obtained.

Definition 11 (Richness) A clustering function F satisfies richness if for any domain set X and partition of that set, $X = X_1 \cup X_2 \dots \cup X_n$, there exists a distance function d over X so that $F(X, d, n) = \{X_1, X_2, \dots, X_n\}$.

We propose an extension on richness. A clustering function satisfies extended 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 12 (Extended 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 $F(\bigcup_{i=1}^n X_i, \hat{d}, n) = \{X_1, X_2, \dots, X_n\}$.

6 Main result

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

Theorem 13 A clustering function is linkage based if and only if it is hierarchical and it satisfies: Outer Consistency, Locality and Extended Richness.

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

6.1 The clustering function 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 14 If a clustering function F is hierarchical and it satisfies Outer Consistency, Locality and Extended Richness, then F is linkage-based.

The proof comprises the rest of this section.

Proof:

Since F is hierarchical, 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} .

We define a relation over pairs of pairs of subsets $<_F$ and later prove that it is a (partial) ordering.

Definition 15 (The (pseudo-) partial ordering $<_F$) $<_F$ is a binary relation over equivalence classes, with respect to clustering-isomorphism. Namely $(A, B, d) \simeq (A', B', d')$ if the two pairs are isomorphic as clusters (see definition in the Preliminary section). We denote equivalence classes by square brackets. So, the domain of $<_F$ is

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

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' (namely, $d \subseteq d^*$ and $d' \subseteq d^*$), 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 $D \in F(X, d^*, k)$.

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 isomorphism invariant. For the sake of simplifying notation, we will omit the square brackets in the following discussion.

To show that $<_F$ can be extended to a partial ordering, we first prove the following property.

Cycle freeness: Given a clustering function F that is outer-consistent, hierarchical, local and satisfies extended richness, there exists no finite sequence $(A_1, B_1, d_1) \dots (A_n, B_n, d_n)$, 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 $(A_1, B_1, d_1) = (A_n, B_n, d_n)$.

This is shown in Lemma 17.

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

Lemma 16 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 clustering function F ,

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

Proof:

Consider a data set on 4 points, $S = \{x, y, x', y'\}$. Let $a = d_1(x, y)$, $b = d_2(x', y')$. We construct a distance function d over S . Set $d(x, y) = b$ and $d(p, q) = a$ for all $\{p, q\} \neq \{x, y\}$.

Then either $(\{x\}, \{y\}, d_1) <_F (\{x'\}, \{y'\}, d_2)$ or $(\{x'\}, \{y'\}, d_2) <_F (\{x\}, \{y\}, d_1)$. Assume by way of contradiction that $d(x, y) < d'(x', y')$ but $(\{x'\}, \{y'\}, d_2) <_F (\{x\}, \{y\}, d_1)$. Then since $(\{x'\}, \{y'\}, d) <_F (\{x\}, \{y\}, d)$, $F(S, d, 3) = \{\{x, y\}, \{x'\}, \{y'\}\}$.

Set $c = b/a$. Note that $c > 1$. Let d' be such that $d'(x, y) = b$, $d'(x', y') = cb$, $d'(p, q) = a$ for all other pairs of elements in S . Then d' is an $(F(S, d, 3), d)$ -outer-consistent variant. 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 2, $F(S, d'', 3) = F(S, d, 3)$. Finally, let d''' be such that $d'''(x', y') = b$, and $d'''(p, q) = a$ for all $\{p, q\} \neq \{x', y'\}$. Note that d''' is an $(F(S, d'', 3), d'')$ -outer-consistent variant. Therefore, $F(S, d''', 3) = F(S, d, 3) = \{\{x, y\}, \{x'\}, \{y'\}\}$. However, since $\ell(X_1, X_2, d_2) < \ell(X_1, X_2, d_1)$, $F(S, d''', 3) = \{\{x', y'\}, \{x\}, \{y\}\}$ - a contradiction. ■

Lemma 17 Given a clustering function F that is outer-consistent, hierarchical, local and satisfies extended richness, there exists no finite sequence $(A_1, B_1, d_1) \dots (A_n, B_n, d_n)$, 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 $X = \bigcup_{i=1}^n A_i \cup B_i$.

Using extended 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 $F(X, \hat{d}, n) = \{C_1, C_2, \dots, C_n\}$.

Let us consider what happens for $F(X, \hat{d}, n+1)$. Since F is hierarchical, the $(n+1)$ -clustering must split one of the C_i 's. Given $1 \leq i < n$, we will show that you 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' 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 a $(F(C_i \cup C_{i+1}, \hat{d}/C_i \cup C_{i+1}, k), \hat{d}/C_i \cup C_{i+1})$ -outer consistent variant, and a $(F(C_i \cup C_{i+1}, d', k), d')$ -outer consistent variant.

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 $C_i \neq C_{i+1}$. ■

Lemma 18 *If $R(,)$ is a binary relation over some domain D that satisfies antisymmetry and cycle-freeness then there exists a partial ordering $R^*(,)$ over D that extends R (i.e., for every $a, b \in D$, if $R(a, b)$ holds then so does $R^*(a, b)$). In fact, the transitive closure of R is such an extension.*

Let $<_F^*$ be the transitive closure of $<_F$. Applying the above lemma it is a partial ordering. The next step is to use the partial ordering $<_F^*$ to define a linkage function that demonstrates that F is a linkage-based clustering.

We shall apply the following basic universality result for partial orderings:

Lemma 19 *Let \prec be a partial ordering over some finite or countable set D , and let h be an order preserving mapping of some $D' \subseteq D$ into the positive reals³, then there exist an extension of h , $\hat{h} : D \rightarrow \mathcal{R}^+$ that is order preserving.*

Finally, we define the embedding

$$\ell_F : \{[(A, B, d)] : A \subseteq \mathcal{N}, B \subseteq \mathcal{N}, A \cap B = \emptyset \text{ and } d \text{ is a distance function over } A \cup B\} \rightarrow \mathcal{R}^+$$

by applying Lemma 19 to $<_F^*$.

Lemma 20 *The function ℓ_F is a linkage function for any hierarchical function F that satisfies locality, outer-consistency, and extended richness.*

Proof: ℓ_F satisfies condition 1 of Definition 3 since it is defined on equivalence classes of isomorphic sets. The function ℓ_F satisfies condition 2 of Definition 3 by lemma 21. By Lemma 22 ℓ_F satisfied condition 3 in Definition 3. ■

Lemma 21 *Consider d_1 over $X_1 \cup X_2$ and d_2 an $(\{X_1, X_2\}, d_1)$ -outer-consistent variant, then $(X_1, X_2, d_2) \not<_F (X_1, X_2, d_1)$.*

³any dense linear ordering with no first element and no last element has the same universality property

Proof: Assume that there exists 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 that is an $(\{X_1, X_2\}, d_1)$ -outer-consistent variant and d_2 is an $(\{X_1, X_2\}, d_3)$ -outer-consistent variant.

By extended 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 extended 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 extended 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 $F(X^*, d^{***}, 3) = \{X'_1 \cup X'_2, X_1, X_2\}$. ■

Claim 22 *The function ℓ_F , for any hierarchical function F that satisfies locality, outer-consistency, and extended richness, satisfies condition 3 of Definition 3.*

Proof: Let r be in the range of ℓ_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 extended 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 an $(F(X, \hat{d}, 4), \hat{d})$ -outer consistent variant 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 an $(F(X_3 \cup X_4, d', 2), d')$ -outer consistent variant. Thus, $\ell_F(X_3, X_4, \tilde{d}/X_3 \cup X_4) \geq r$.

Also by extended 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 an $(F(X, \tilde{d}, 4), \tilde{d})$ -outer consistent variant and an $F(X, \hat{d}', 3), \hat{d}'$ -outer consistent variant 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, $\ell_F(X_1, X_2, \tilde{d}') > \ell_F(X_3, X_4, \tilde{d}') \geq r$. ■

Claim 23 *For every clustering function F , the linkage-based clustering that ℓ_F defines agrees with F on any input data set.*

Proof: For every (X, d) , the linkage based clustering that ℓ_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 ℓ_F . Therefore, F would merge the same clusters to obtain the $k-1$ clustering, and so ℓ_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.

6.2 Every linkage-based clustering function is hierarchical, local, and outer-consistent

If a clustering function is linkage-based, then by construction it is hierarchical.

Lemma 24 *Every linkage-based clustering function is hierarchical.*

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 25 *Every linkage-based clustering function F is local.*

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

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 .

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 ℓ so that when ℓ is used in the algorithm in Definition 4, 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 $F(X, d, j) \cap C$. The pair of clusters with minimal ℓ value in $F(X, d, j) \cap C$ is the same as the pair of clusters with minimal ℓ value in $F(X', d/X', i)$. Therefore, $j' = j^*$. ■

Lemma 26 *Every linkage-based clustering function F is outer-consistent.*

Proof: By the monotonicity condition in Definition 3, whenever two clusters are pulled further apart from each other, the corresponding ℓ value does not decrease. Consider some data set (X, d) and d' an $(F(X, d, k), d)$ -outer-consistent variant. 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 ℓ 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 an $(F(X, d, k), d)$ -outer-consistent variant, 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 27 *Every linkage-based function satisfies extended richness.*

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

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 $r = \max_{i, A, B \subseteq X_i, i \in \{1, 2\}} \ell(A, B)$. Since ℓ satisfies property 4 of Definition 3, 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 ℓ satisfies property 2 of Definition 3, $\ell(C, D) > r$ 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 is any clusters $A, B \subseteq X_i$ for some i remaining. This gives that $F(\bigcup_{i=1}^n X_i, d, n) = \{X_1, X_2, \dots, X_n\}$. ■

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

Theorem 13 restated *A clustering function is linkage based if and only if it is hierarchical and it satisfies: Outer Consistency, Locality and Extended Richness.*

Proof: By Theorem 4, if a clustering function is outer-consistent, hierarchical, and local, then it is linkage-based. By Lemma 24, every linkage-based clustering function is hierarchical. By and Lemma 25 every linkage-based clustering function is local. By Lemma 26, every linkage-based clustering function is outer-consistent. Finally, by Lemma 27, every linkage based function satisfies extended richness. ■

7 Relaxations of a Linkage Function and Corresponding Characterizations

7.1 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 clustering functions that are based of linkage functions that are not required to obey the conditions of Definition 3.

Definition 28 (Simplified linkage function) A linkage function ℓ takes a data set (X, d) and a partition (X_1, X_2) of the domain X .

We then define a *simplified linkage-based function* as in Definition 4, but with a simplified linkage function instead of the linkage function in Definition 3.

We obtain an interesting characterization of simplified linkage-based function that satisfy outer-consistency and extended richness.

Theorem 29 A clustering function that satisfies outer-consistency and extended richness is simplified linkage-based if and only if it hierarchical and local.

Proof: Since a linkage function is a simplified linkage function with additional constraints, by Theorem 14 we get that an outer-consistent, hierarchical and local clustering function is simplified linkage-based. The results and proofs of Theorem 24 and Theorem 25 also apply for simplified linkage functions, thus showing that simplified linkage-based functions are hierarchical and local. ■

7.2 General linkage function

Unlike linkage-based clustering functions defined in Definition 4 or simplified linkage-based functions, a *general linkage-based 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 30 (General linkage function) A general linkage function is given a data set (X, d) and $A, B \subseteq X$, and 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 clustering function* as in Definition 4, except using a general linkage function instead of the linkage function in definition 3.

Definition 31 (general linkage-based clustering function) A clustering function F is linkage-based if there exists a general linkage function ℓ 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 \bar{d} . Formally,

$$F(X, d, k) = \{c \mid c \in F(X, d, k+1), c \neq c_i, c \neq c_j\} \cup \{c_i \cup c_j\},$$

such that $\{c_i, c_j\} = \operatorname{argmin}_{\{c_i, c_j\} \subseteq F(X, d, (k+1))} \ell((X, d), c_i, c_j)$.

For example, a 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 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.

Theorem 32 A clustering function is hierarchical if and only if it is a general linkage-based clustering function.

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

Assume that F is hierarchical. Then whenever $k' > k$, $F(X, d, k)$ can be obtained from $F(X, d, k')$ by merging clusters in $F(X, d_X, k')$. 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 ℓ 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 the function resulting from using the general linkage function ℓ 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 $F'(X, d_X, k) = F(X, d, k)$. Since F' is a general linkage based 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 hierarchical, 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 ℓ 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 $F'(X, d, k) = F(X, d, k)$. ■

8 Conclusions

We address the task of understanding the consequences of choosing one clustering algorithm over another. As we have argued in the introduction, we view it as a very important task, if only to help users make better educated choices about the clustering algorithms that they apply to their data. In spite of its importance, very little has been previously done along these lines.

In any attempt to taxonomize clusterings, a natural family of clustering that one needs to distinguish is the class of linkage based clustering. In this work we succeeded in characterizing this rich and popular class by a relatively small set of intuitive abstract properties. We sincerely believe that these results will encourage other researchers to follow up this ambitious task of providing guidelines in the zoo of clustering algorithms.

9 Acknowledgements

We wish to thank Reza Bosagh Zadeh for stimulating discussions of the axiomatic approach to clustering and for proposing the locality property in that context.

References

- [1] M. Ackerman and S. Ben-David. Measures of Clustering Quality: A Working Set of Axioms for Clustering. NIPS, 2008.
- [2] Reza Bosagh Zadeh and Shai Ben-David. "A Uniqueness Theorem for Clustering." The 25th Annual Conference on Uncertainty in Artificial Intelligence (UAI 09), 2009.
- [3] Chris Ding and Xiaofeng He. "Cluster Aggregate Inequality and Multi-level Hierarchical Clustering." Knowledge Discovery in Databases (PKDD), 2005. LNCS, Springer Berlin / Heidelberg, V. 3721, pp. 71-83
- [4] Brian Everitt, Sabine Landau, and Morven Leese. Cluster Analysis, 4th edn. Arnold, London. 2001.
- [5] Derek Greene, Gerard Cagney, Nevan Krogan, and Pdraig Cunningham. "Ensemble non-negative matrix factorization methods for clustering proteinprotein interactions." Bioinformatics Vol. 24 no. 15 2008, pages 1722-1728
- [6] N. Jardine and R. Sibson. The construction of hierarchic and non-hierarchic classifications. Multivariate Statistical Methods, Among-groups Covariation, 1975.
- [7] Jon Kleinberg. "An Impossibility Theorem for Clustering." Advances in Neural Information Processing Systems (NIPS) 15, 2002.
- [8] U. von Luxburg. A Tutorial on Spectral Clustering. Statistics and Computing 17(4): 395-416, 2007

10 Appendix: spectral clustering does not satisfy locality

In this appendix we show that spectral clustering does not satisfy locality. This illustrates that locality is a property of clustering functions (one that is satisfied by some, but not all, reasonable clustering functions), and not an axiom (which is satisfied by all reasonable clustering functions).

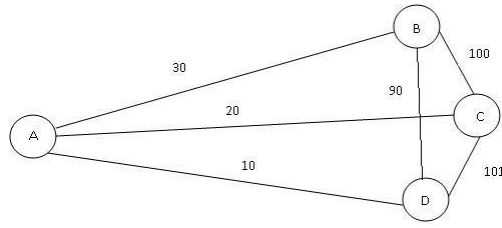


Figure 1: A data set used to illustrate that Ratio-Cut does not satisfy locality.

We discuss two clustering functions from spectral clustering: ratio-cut and normalized-cut. For more on spectral clustering, see a tutorial by Luxburg [8].

In spectral clustering we assume that there is an underlying similarity function s , instead of a distance function. The only difference between a distance functions and a similarity functions is that higher values of represent greater similarity when using similarity functions, while the opposite holds for distance functions.

Let $|c|$ denote the number of elements in cluster c . \bar{c} denotes the data set X without the points in c . For $c_1, c_2 \subseteq X$, let $cut(c_1, c_2) = \sum_{a \in c_1, b \in c_2} s(a, b)$.

Definition 33 (ratio-cut clustering function) Given a data set (X, d) and an integer $1 \leq k \leq |X|$, the ratio-cut clustering function finds a k -clustering $\{c_1, c_2, \dots, c_k\}$ that minimizes

$$\sum_1^k \frac{cut(c_i, \bar{c}_i)}{|c_i|}.$$

The normalized-cut clustering function takes into account within-cluster similarity. Let $vol(c_i) = \sum_{a, b \in c_i} s(a, b)$.

Definition 34 (normalized-cut clustering function) Given a data set (X, d) and an integer $1 \leq k \leq |X|$, the normalized-cut clustering function finds a k -clustering $\{c_1, c_2, \dots, c_k\}$ that minimizes

$$\sum_1^k \frac{cut(c_i, \bar{c}_i)}{vol(c_i)}.$$

Theorem 35 *Ratio-Cut is not local.*

Proof:

Figure 1 illustrates a data set (with the similarity indicated on the arrows) 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 1), the clustering $\{\{B\}, \{C, D\}\}$ has lower ratio-cut than $\{\{B, C\}, \{D\}\}$. ■

We now show that normalized-cut is not local.

Theorem 36 *Normalized-Cut is not local.*

Proof: Figure 2 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 2), the clustering $\{\{B, B'\}, \{C, C', D, D'\}\}$ has lower normalized-cut than $\{\{B, B', C, C'\}, \{D, D'\}\}$. ■

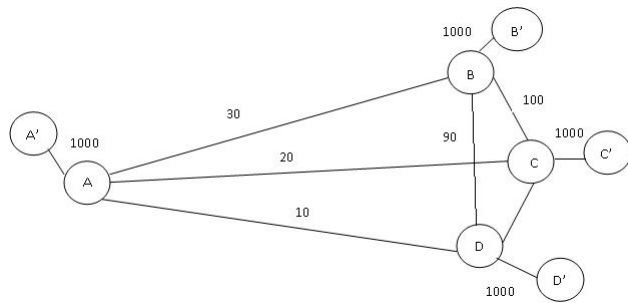


Figure 2: A data set used to illustrate that Ratio-Cut does not satisfy locality.