Discerning Linkage-Based Algorithms Among Hierarchical Clustering Methods

Margareta Ackerman and **Shai Ben-David**
D.R.C School of Computer Science
University of Waterloo
{mackerma, shai}@cs.uwaterloo.ca

Abstract

Selecting a clustering algorithm is a perplexing task. Yet since different algorithms may yield dramatically different outputs on the same data, the choice of algorithm is crucial. When selecting a clustering algorithm, users tend to focus on cost-related considerations (software purchasing costs, running times, etc). Differences concerning the output of the algorithms are not usually considered.

Recently, a formal approach for selecting a clustering algorithm has been proposed [2]. The approach involves distilling abstract properties of the input-output behavior of different clustering paradigms and classifying algorithms based on these properties.

In this paper, we extend the approach in [2] into the hierarchical setting. The class of linkage-based algorithms is perhaps the most popular class of hierarchical algorithms. We identify two properties of hierarchical algorithms, and prove that linkage-based algorithms are the only ones that satisfy both of these properties.

Our characterization clearly delineates the difference between linkage-based algorithms and other hierarchical algorithms. We formulate an intuitive notion of locality of a hierarchical algorithm that distinguishes between linkage-based and "global" hierarchical algorithms like bisecting k-means, and prove that popular divisive hierarchical algorithms produce clusterings that cannot be produced by any linkage-based algorithm.

1 Introduction

Clustering is a fundamental and immensely useful task, with many important applications. There are many clustering algorithms, and these algorithms often produce different results on the same data. Faced with a concrete clustering task, a user needs to choose an appropriate algorithm. Currently, such decisions are often made in a very ad hoc, if not completely random, manner. Users are aware of the costs involved in employing different clustering algorithms, such as running times, memory requirements, and software purchasing costs. However, there is very little understanding of the differences in the outcomes that these algorithms may produce.

Recently, a new approach for selecting a clustering algorithm has been proposed [2]. The approach involves identifying significant properties that distinguish between different clustering paradigms. By focusing on the input-output behaviour of algorithms, these properties shed light on essential differences between them. Users could then choose desirable properties based on domain expertise, and select an algorithm that satisfies these properties.

The study of properties of clustering algorithms has so far been focused on partitional algorithms ([2], [1], [3], [4]). Partitional algorithms produce a single partition of the data.

In this paper, we study the other prominent class of clustering algorithms, namely hierarchical algorithms. These algorithms output dendrograms, which the user can then traverse to obtain the desired clustering. Dendrograms provide a convenient method for exploring multiple clusterings of the data. Notably, 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 phylogeny, which aims to reconstruct the tree of life.

One popular class of hierarchical algorithms is linkage-based algorithms. These algorithms start with singleton clusters, and repeatedly merge pairs of clusters until a dendrogram is formed. This class includes commonly-used algorithms such as single-linkage, average-linkage, complete-linkage, and Ward's method.

In this paper, we provide a property-based characterization of hierarchical linkage-based algorithms. We identify 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.

2 Previous Work

There is some previous work on distinguishing linkage-based algorithms based on their properties in the partitional setting. The k-stopping criterion is used to formulate linkage-based algorithms in the partitional setting, where instead of constructing a dendrogram, clusters are merged until a given number of clusters is formed. Ben-David and Bogah-Zedah [3] provide three properties that uniquely identify single-linkage with the k-stopping criterion. More recently, Ackerman, Ben-David and Loker [1] characterize linkage-based algorithms with this stopping criterion in terms of three natural properties. These results enable a comparison of the input-output behaviour of (a partitional variant of) linkage-based algorithms with other partitional algorithms.

In this paper, we characterize hierarchical linkage-based algorithms, which map data sets to dendrograms. Our characterization is independent of any stopping criterion. It enables the comparison of linkage-based algorithms to other hierarchical algorithms, and clearly delineates the differences between the input/output behaviour of linkage-based algorithms from all other hierarchical algorithms.

3 Definitions

A *distance function* is a symmetric function $d : X \times X \to R^+$, such that $d(x,x) = 0$ for all $x \in X$. The data sets that we consider are pairs (X, d), where X is some finite domain set and d is a distance function over X. We say that a distance function d over X *extends* distance function d' over $X' \subseteq X$ if $d' \subseteq d$.

A *k-clustering* $C = \{C_1, C_2, \ldots, C_k\}$ of a data set X is a partition of X into k non-empty disjoint subsets of X (so, $\cup_i C_i = X$). A *clustering* of X is a k-clustering of X for some $1 \leqslant k \leqslant |X|$. 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.

Given a rooted 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, descendent, 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 (η); leaves are placed at level 0, parents have higher levels than their children, and no level is empty. This definition represents the common graphical depiction of a dendrogram. Formally,

Definition 1 (dendrogram). *A* dendrogram *over (X, d) is a triple (T, M, η) where T is a binary rooted tree, M :* leaves$(T) \to X$ *is a bijection, and $\eta : V(T) \to \{0, \ldots, h\}$ is onto (for some $h \in \mathbb{Z}^+ \cup \{0\}$) 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 $1 \leqslant i \leqslant k$. 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 2 (sub-dendrogram). *A* sub-dendrogram *of (T, M, η) rooted at $x \in V(T)$ is a dendrogram (T', M', η') where*

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

Definition 3 (Isomorphisms). *A few notions of isomorphisms of structures are relevant to our discussion.*

1. *We say that (X, d) and (X', d') are* isomorphic domains, *denoted $(X, d) \cong_X (X', d')$, if there exists a bijection $\phi : X \to X'$ so that $d(x, y) = d'(\phi(x), \phi(y))$ for all $x, y \in X$.*
2. *We say that two clusterings (or partitions) 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 $\phi(x) \sim_{C'} \phi(y)$.*
3. *We say that (T_1, η_1) and (T_2, η_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 $x \in V(T_1)$, $\eta_1(x) = \eta_2(H(x))$.*
4. *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 $x \in$* leaves(T_1)*, $\phi(M_1(x)) = M_2(H(x))$.*

4 Hierarchical and Linkage-Based Algorithms

Hierarchical algorithms are those that output dendrograms, while linkage-based algorithms are hierarchical algorithms that can be simulated by repeatedly merging close clusters. In this section, we formally define these two types of algorithms.

4.1 Hierarchical algorithms

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

Definition 4 (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, η). 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* $c \in R^+$ *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* $\{(X_1, d_1), \dots (X_k, d_k)\}$ *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* $i \leqslant k$*), so that the clustering* $\{C_1, \dots, C_k\}$ *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 clusters sufficiently far apart, then the resulting clustering should be in the dendrogram.

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

4.2 Linkage-based algorithms

The class of linkage-base algorithms includes some of the most popular hierarchical algorithms, such as single-linkage, average-linkage, complete-linkage, and Ward's method.

Every linkage-based algorithm has a linkage function that can be used to determine which clusters to merge at every step of the algorithm.

Definition 5 (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. ℓ *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. 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* $x \not\sim_{\{X_1, X_2\}} y$*,* $d(x, y) \leqslant d'(x, y)$ *then* $\ell(X_1, X_2, d') \geqslant \ell(X_1, X_2, d)$.

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

For example, the single-linkage linkage function is $\ell_{SL}(A, B, d) = \min_{a \in A, b \in B} d(a, b)$ and the average-linkage linkage function is $\ell_{AL}(A, B, d) = \frac{\sum_{a \in A, b \in B} d(a, b)}{|A| \cdot |B|}$.

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

We now define linkage-based functions.

Definition 6 (linkage-based function). *A hierarchical clustering function* F *is* linkage-based *if there exists a linkage function* ℓ *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* $\{\ell(S, T) : S \cap T = \emptyset, \eta(\text{parent}(S)) \geqslant m, \eta(\text{parent}(T)) \geqslant m\}$.

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 (wrt the clusters that they represent), and repeatedly merge the closest pairs of nodes that do yet have parents, until only one such node remains.

4.3 Locality

We introduce a new property of hierarchical algorithms. Locality states that if we select a clustering from a dendrogram (a union of 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 7 (Locality). *A hierarchical function* F *is* local *if for all* X*,* d*, and* $X' \subseteq X$*, whenever clustering* $C = \{C_1, C_2, \dots, C_k\}$ *of* X' *is in* $F(X, d) = (T, M, \eta)$*, then for all* $1 \leqslant i \leqslant k$

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 phylogeny, 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.

4.4 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 (or become even less similar), then it is natural to expect that, when clustering the resulting new data set, such points will not share the same cluster.

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 8 (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* $\eta(u) \leqslant \eta(v(A)).\}$.

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.

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

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

Definition 9 (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 a $(cut_A(F(X, d)), d)$-outer-consistent variant then $cut_A(F(X, d)) = cut_A(F(X, d'))$.*

5 Main result

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

Theorem 1. *A hierarchical function F is linkage-based if and only if F is outer-consistent and local.*

The proof comprises the rest of this section.

Proof. We begin by showing that every local, outer-consistent hierarchical function F is linkage-based. To prove this direction, we show that there exists a linkage function ℓ so that when ℓ is used in Definition 6 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, $\mathbb{N}$.

Definition 10 (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 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

1. *d^* extends both d and d' (namely, $d \subseteq d^*$ and $d' \subseteq d^*$),*
2. *There exist $(x, y), (x, z) \in E(T)$ such that $\mathcal{C}(x) = A \cup B$, $\mathcal{C}(y) = A$, and $\mathcal{C}(z) = B$*
3. *For all $D \in \{A', B'\}$, either $D \subseteq A \cup B$, or $D \in cut_{A\cup B}F(X, d^*)$.*
4. *$\eta(v(A')) < \eta(v(A \cup B))$ and $\eta(v(B')) < \eta(v(A \cup B))$.*

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 1. *Given a local and outer-consistent hierarchial function F, whenever $(A, B, d) <_F (A', B', d')$, then there is no data set (X, d) so that $\eta(v(A' \cup B') \leqslant \eta(v(A \cup B))$, where $F(X, d) = (T, M, \eta)$.*

Proof. By way of contradiction, assume that such (X, d) exists. Let $X' = A \cup B \cup A' \cup B'$. Since $(A, B, d) <_F (A', B', d')$, there exists d' that satisfies the conditions of Definition 10.

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

By pushing A away from B, and A' away from B' it is easy to see that there exists a d^* over X' that is both an $(\{A \cup B, A' \cup B'\}, d/X')$-outer consistent variant and an $(\{A \cup B, A', B'\}, d')$-outer consistent variant. Note that $\{A \cup B, A' \cup B'\}$ is an $(A \cup B)$-cut of $F(X', d/X')$. Therefore, by outer-consistency, $cut_{A\cup B}(F(X', d^*)) = \{A' \cup B', A \cup B\}$.

Since d' satisfies the conditions in Definition 10, $cut_{A\cup B}F(X, d') = \{A \cup B, A', B'\}$. By outer-consistency we get that $cut_{A\cup B}(F(X', d^*)) = \{A' \cup B', A, B\}$. Since these sets are all non-empty, this is a contradiction. □

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

Definition 11 ($\cong_F$). *$[(A, B, d)]$ and $[(A', B', d')]$ are F-equivalent, denoted $[(A, B, d)] \cong_F [(A', B', d')]$, if 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*

1. *d^* extends both d and d',*
2. *There exist $(x, y), (x, z) \in E(T)$ such that $\mathcal{C}(x) = A \cup B$, and $\mathcal{C}(y) = A$, and $\mathcal{C}(z) = B$,*
3. *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*
4. *$\eta(x) = \eta(x')$*

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

Lemma 2. *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 $(C, D, d_2) <_F (E, F, d_3)$*

The proof is omitted to save space.

Note that $<_F$ is not transitive. To show that $<_F$ can be extended to a partial ordering, we first prove the following "anti-cycle" property.

Lemma 3. *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 $\eta(v(A_{i_0} \cup B_{i_0}) \leqslant \eta(v(A_j \cup B_j))$ 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 1. □

The following is a well-known result.

Lemma 4. *For any cycle-free, anti-symmetric relation $P(\ ,\ )$ over a finite or countable domain D there exists an embedding h into 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 4 to $<_F$. Namely, $\ell_F : \{[(A, B, d)] : A \subseteq \mathbb{N}, B \subseteq \mathbb{N}, A \cap B = \emptyset \textit{ and } d \textit{ is a distance function over } A \cup B\} \to \mathbb{R}^+$ so that $[(A, B, d)] <_F [(A', B', d')]$ implies $\ell_F[(A, B, d)] < \ell_F[(A, B, d)]$.

Lemma 5. *The function ℓ_F is a linkage function for any hierarchical function F that satisfies locality and outer-consistency.*

Proof. Since ℓ_F is defined on $\cong_F$-equivalence classes, representation independence of hierarchical functions implies that ℓ_F satisfies condition 1 of Definition 5. The function ℓ_F satisfies condition 2 of Definition 5 by lemma 6. □

Lemma 6. *Consider d_1 over $X_1 \cup X_2$ and d_2 an $(\{X_1, X_2\}, d_1)$-outer-consistent variant, then $(X_1, X_2, d_2) \not<_F (X_1, X_2, d_1)$, 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 that is both an $(\{X_1, X_2\}, d_1)$-outer-consistent variant and d_2 an $(\{X_1, X_2\}, d_3)$-outer-consistent variant.

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 $1 \leqslant i \leqslant 3$ so that $\{X_1 \cup X_2, X_1' \cup X_2', X_1'' \cup X_2''\}$ is a clustering in $F(X, d^*) = (T, \eta)$.

Now, $(X_1', X_2', d_2) <_F (X_1, X_2, d_1)$, 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 (η value) of $v(X_1'' \cup X_2'')$ with respect to the levels of $v(X_1' \cup X_2')$ and $v(X_1 \cup X_2)$.

The proof proceeds by analyzing the following three cases: $\eta(v(X_1'' \cup X_2'')) \leqslant \eta(v(X_1' \cup X_2'))$, $\eta(v(X_1'' \cup X_2'')) \geqslant \eta(v(X_1 \cup X_2))$ and $\eta(X_1 \cup X_2) < \eta(X_1'' \cup X_2'') < \eta(X_1' \cup X_2')$. The details of these cases are omitted for brevity. □

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

Lemma 7. *Given any hierarchical function F that satisfies locality and outer-consistency, let ℓ_F be the linkage function defined above. Let L_{ℓ_F} denote the linkage-based algorithm that ℓ_F defines. Then L_{ℓ_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 ℓ_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 $\eta(\text{parent}(y_1)) \geqslant k$. 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, $(x_1, x_2, d) <_F (x_1, y_1, d)$ and $\ell_F(x_1, x_2, d) < \ell_F(x_1, y_1, d)$. Similarly, $\ell_F(x_1, x_2, d) < \ell_F(x_2, y_1, d)$

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 $\eta(\text{parent}(y_2)) \geqslant k$. If $\text{parent}(y_1) = \text{parent}(y_2)$ and $\eta(\text{parent}(y_1)) = k$, then $(x_1, x_2, d) \cong_F (y_1, y_2, d)$ and so $\ell_F(x_1, x_2, d) = \ell_F(y_1, y_2, d)$.

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, $(x_1, x_2, d) <_F (y_1, y_2, d)$ and so $\ell_F(x_1, x_2) < \ell_F(y_1, y_2)$. □

The other direction of the proof is straight forward, and so the proof of the following Lemma is omitted to save space.

Lemma 8. *Every linkage-based hierarchical clustering function satisfies locality and outer-consistency.*

□

6 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 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 12 ($\mathcal{F}$-Divisive). *A hierarchical clustering function is $\mathcal{F}$-Divisive wrt 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 12 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 13 (Context sensitive). $\mathcal{F}$ *is* context-sensitive *if there exist distance functions* $d \subset d'$ *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 2). 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 2. *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 $\mathcal{F}(\{x, y, z, w\}, d')$. Then $\{\{x, y\}, \{z\}\}$ is a clustering in $\mathcal{F}\text{-}divisive(\{x, y, z\}, d)$. But cluster $\{x, y\}$ is not in $\mathcal{F}(\{x, y, z\}, d)$, therefore $\mathcal{F}$-divisive is not local. □

Applying Theorem 1, we get:

Corollary 1. *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 14. *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 3. *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 $\mathcal{F}\text{-}divisive$ produce the same output on $(\{x, y, z, w\}, d')$. Therefore, since $\{\{x, y\}, \{z\}\}$ is a clustering in $\mathcal{F}\text{-}divisive(\{x, y, z, w\}, d')$, it is also a clustering in $L(\{x, y, z, w\}, d')$. Since L is linkage-based, by Theorem 1, L is local. Therefore, $\{\{x, y\}, \{z\}\}$ is a clustering in $L(\{x, y, z\}, d')$. But it is not a clustering in $\mathcal{F}\text{-}divisive(\{x, y, z\}, d)$. □

Corollary 2. *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 3.

□

References

[1] M. Ackerman, S. Ben-David, and D. Loker. Characterization of Linkage-based Clustering. COLT, 2010.

[2] M. Ackerman, S. Ben-David, and D. Loker. Towards Property-Based Classification of Clustering Paradigms. NIPS, 2010.

[3] R. Bosagh Zadeh and S. Ben-David. A Uniqueness Theorem for Clustering. UAI, 2009.

[4] J. Kleinberg. An Impossibility Theorem for Clustering. NIPS, 2002.