Journal of Machine Learning Research 17 (2016) 1-17 Submitted 7/11; Revised 8/15; Published 12/16

A Characterization of Linkage-Based Hierarchical Clustering

Margareta Ackerman MARGARETA.ACKERMAN@SJSU.EDU
Department of Computer Science
San Jose State University
San Jose, CA

Shai Ben-David SHAI@CS.UWATERLOO.CA
D.R.C. School of Computer Science
University of Waterloo
Waterloo, ON

Editor: Marina Meila

Abstract

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 methods. 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.

It has been proposed to address this challenge by identifying significant properties that distinguish between different clustering paradigms (see, for example, Ackerman et al. (2010b) and Fisher and Van Ness (1971)). By focusing on the input-output behaviour of algorithms, these properties shed light on essential differences between them (Ackerman et al. (2010b, 2012)). Users could then choose desirable properties based on domain expertise, and select an algorithm that satisfies these properties.

In this paper, we focus hierarchical algorithms, a prominent class of clustering 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 leads 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

Our work falls within the larger framework of studying properties of clustering algorithms. Several authors study such properties from an axiomatic perspective. For instance, Wright (1973) 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. A recent, and influential, paper in this line of work is Kleinberg's impossibility result (Kleinberg (2003)), where he proposes three axioms of partitional clustering functions and proves that no clustering function can simultaneously satisfy these properties.

Properties have been used study different aspects of clustering. Ackerman and Ben-David (2008) consider properties satisfied by clustering quality measures, showing that properties analogous to Kleinberg's axioms are consistent in this setting. Meila (2005) 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. (2000) explore properties of clustering objective functions. They propose a few natural properties of clustering objective functions, and then focus on objective functions that arise by requiring functions to decompose into additive form.

Most relevant to our work are previous results distinguishing linkage-based algorithms based on their properties. Most of these results are concerned with the single-linkage algorithm. In the hierarchial clustering setting, Jardine and Sibson (1971) and Carlsson and Mémoli (2010) formulate a collection of properties that define single linkage.

Zadeh and Ben-David (2009) characterize single linkage in the partitional setting where instead of constructing a dendrogram, clusters are merged until a given number of clusters remain. Finally, Ackerman et al. (2010a) characterize linkage-based algorithms in the same partitional setting in terms of a few natural properties. These results enable a comparison

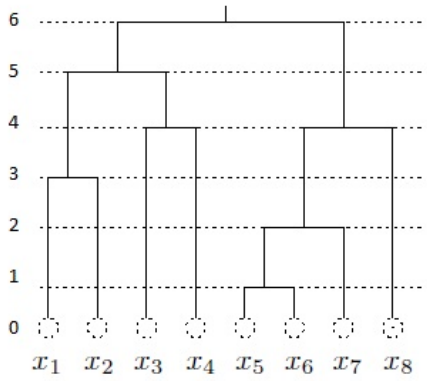

Figure 1: A dendrogram of domain set $\{x_1, \ldots, x_8\}$. The horizontal lines represent levels and every leaf is associated with an element of the domain.

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 and other hierarchical methods.

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

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. See Figure 1 for an illustration. Formally,

Definition 1 (dendrogram) *A* dendrogram *over (X, d) is a triple (T, M, η) 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 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 $y \in \text{leaves}(T')$, $M'(y) = M(y)$, and*
3. *For all $y, z \in V(T')$, $\eta'(y) < \eta'(z)$ if and only if $\eta(y) < \eta(z)$.*

Definition 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 $x \in \text{leaves}(T_1)$, $\phi(M_1(x)) = M_2(H(x))$.*

4. 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.

4.1 Hierarchical Algorithms

In addition to outputting 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* $c \in \mathbb{R}^+$ *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* $i \leqslant k$*), 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.

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* $\ell(X_1, X_2, d') \geqslant \ell(X_1, X_2, d)$*.*

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.

The following are the linkage-functions of some of the most popular linkage-based algorithms,

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

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 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 $\{\ell(S, T) : S \cap T = \emptyset, \eta(S) < m, \eta(T) < m, \eta(\text{parent}(S)) \geqslant m, \eta(\text{parent}(T)) \geqslant m\}$.*

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 (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.

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 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 7 (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 $1 \leqslant i \leqslant k$*

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 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.

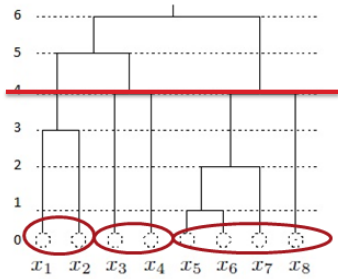

Figure 2: An example of an A-cut.

4.4 Outer Consistency

Clustering aims to group similar elements and separate dissimilar ones. These two requirements are often contradictory and algorithms vary in how they resolve this contradiction. Kleinberg (2003) proposed a formalization of these requirements in his "consistency" axiom for partitional clustering algorithms. 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.

Since then it was found that while many natural clustering functions fail consistency, most satisfy a relaxation, which requires that the output of an algorithm is not changed by increasing between-cluster distances (Ackerman et al. (2010b)). 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. Here we propose a variation of this requirement for the hierarchical clustering 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 $v(A)$. For convenience, if node u is the root of the dendrogram, then assume its parent has infinite level, $\eta(\mathit{parent}(u)) = \infty$.

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.

For example, consider the diagram in Figure 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 $d'(x, y) \geqslant d(x, y)$ 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 $(cut_A(F(X, d)), d)$-outer-consistent 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 10 *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.

5.1 All Local, Outer-Consistent Hierarchical Functions are Linkage-Based

Lemma 11 *If a hierarchical function F is outer-consistent and local, then F is linkage-based.*

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 12 (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 13 *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* $\eta(v(A_2 \cup B_2)) \leqslant \eta(v(A_1 \cup B_1))$, *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 12.

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 12, $cut_{A_1 \cup B_1} F(X, d') = \{A_1 \cup B_1, A_2, B_2\}$. 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 14 ($\cong_F$) $[(A, B, d)]$ *and* $[(A', B', d')]$ *are* F*-equivalent, denoted* $[(A, B, d)] \cong_F [(A', B', d')]$, *if either they are isomorphic as clusterings,* $(\{A, B\}, d) \cong_C (\{A', B'\}, d')$ *or*

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* $F(A \cup B \cup A' \cup B', d^*) = (T, \eta)$ *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 15 *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 4), 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 $(C, D, d_2) <_F (E, F, d_3)$. 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 16 *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 13. ■

We make use of the following general result.

Lemma 17 *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 $x_1, ..., x_k$ so that $P(a, x_1), P(x_2, x_3), ..., P(x_k, b)$. 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 $x_{i_1}...x_{i_k}$ be all the members of $\{x_1, ...x_n\}$ that are below x_{n+1} in the partial order. Let $r_1 = \max\{\phi(x_{i_1}), ...\phi(x_{i_k}\}$, 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 ϕ maps $\{x_1, ...x_n, x_{n+1}\}$ in a way that respects the partial order. ■

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 17 to $<_F$. Namely, $\ell_F : \{[(A, B, d)] : A \subseteq \mathbb{N}, B \subseteq \mathbb{N}, A \cap B = \emptyset$ *and* d *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 18 *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 19, whose proof follows. ■

Lemma 19 *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 $(X_1', X_2', d_2) \cong_F (X_1, X_2, d_2)$ and $(X_1'', X_2'', d_3) \cong_F (X_1, X_2, d_3)$.

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 $F(X, d^*)$.

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 (η value) of $v(X_1'' \cup X_2'')$ with respect to the levels of $v(X_1' \cup X_2')$ and $v(X_1 \cup X_2)$ in $F(X, d^*)$.

We now consider a few cases.

Case 1: $\eta(v(X_1'' \cup X_2'')) \leqslant \eta(v(X_1' \cup X_2'))$. 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 dendrograms, one in which the further pair $((X_1', X_2')$ with distance function $d_2)$ is not below the medium pair $((X_1'', X_2'')$ with distance function $d_3)$, and the other in which the medium pair is above the furthest pair.

Case 2: $\eta(v(X_1'' \cup X_2'')) \geqslant \eta(v(X_1 \cup X_2))$. 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 in 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 20 *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, $(\mathcal{C}(x_1), \mathcal{C}(x_2), d) <_F (\mathcal{C}(x_1), \mathcal{C}(y_1), d)$ 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 $\eta(\text{parent}(y_2)) \geqslant k$. 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 $\ell_F(\mathcal{C}(x_1), \mathcal{C}(x_2), d) < \ell_F(\mathcal{C}(y_1), \mathcal{C}(y_2), d)$. ■

5.2 All Linkage-Based Functions are Local and Outer-Consistent

Lemma 21 *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 $1 \leqslant i \leqslant k$, the sub-dendrogram rooted at $v(C_i)$ in $F(X, d)$ also appears in $F(X, d')$, with the same relative levels. ■

Lemma 22 *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 $1 \leqslant i \leqslant k$. Let d' be (C, d)-outer-consistent. Then for all $1 \leqslant i \leqslant k$, 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 $i \neq j$, $\ell(X_1, X_2, d) \leqslant \ell(X_1, X_2, d')$ by monotonicity. Therefore, for all $1 \leqslant j \leqslant k$, 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 $F(X, d)$. ■

5.3 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 ℓ is not a linkage-function since it fails the monotonicity condition. The function ℓ 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 ℓ in a greedy fashion to construct a dendrogram (by repeatedly merging the closest clusters according to ℓ), then the function F is local by the same argument as the proof of Lemma 21. 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.

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 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 23 ($\mathcal{F}$-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 23 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 24 (Context sensitive) *$\mathcal{F}$ is* context-sensitive *if there exist x, y, z, w and 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 29, 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 25 *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 10, we get:

Corollary 26 *If $\mathcal{F}$ is context-sensitive, then the $\mathcal{F}$-divisive function is not linkage-based.*

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

Definition 27 *Two hierarchical functions F_0 and F_1* 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 28 *If $\mathcal{F}$ is context-sensitive, then the $\mathcal{F}$-divisive function 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 10, 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 29 *The divisive algorithms that are based on the following 2-clustering functions 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 28. ■

7. Conclusions

In this paper, we provide the first property-based characterization of hierarchical linkage-based clustering. Our characterization shows the existence of hierarchical methods that cannot be simulated by any linkage-based method, revealing inherent input-output differences between agglomeration and divisive hierarchical algorithms.

This work falls in the larger framework of property-based analysis of clustering algorithms, which aims to provide a better understanding of these techniques as well as aid users in the crucial task of algorithm selection. It is important to note that our characterization is not intended to demonstrate the superiority of linkage-based methods over other hierarchical techniques, but rather to enable users to make informed trade-offs when choosing algorithms. In particular, properties investigated in previous work should also be considered, while future work will continue to investigate important properties with the ultimate goal of providing users with a property-based taxonomy of popular clustering methods that would enable selecting suitable methods for a wide range of applications.

8. Acknowledgements

We would like to thank David Loker for several helpful discussions. We would also like to thank the anonymous referees whose comments and suggestions greatly improved this paper.

References

M. Ackerman and S. Ben-David. Measures of clustering quality: A working set of axioms for clustering. In *Proceedings of Neural Information Processing Systems (NIPS)*, pages 121–128, 2008.

M. Ackerman, S. Ben-David, and D. Loker. Characterization of linkage-based clustering. In *Proceedings of The 23rd Conference on Learning Theory*, pages 270–281, 2010a.

M. Ackerman, S. Ben-David, and D. Loker. Towards property-based classification of clustering paradigms. *Lafferty et al.*, pages 10–18, 2010b.

M. Ackerman, S. Ben-David, S. Branzei, and D. Loker. Weighted clustering. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 858–863, 2012.

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

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

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

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

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

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.

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

R.B. Zadeh and S. Ben-David. A uniqueness theorem for clustering. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 639–646. AUAI Press, 2009.