Hierarchical Clustering via Spreading Metrics

Aurko Roy[1] and Sebastian Pokutta[2]

[1]College of Computing, Georgia Institute of Technology, Atlanta, GA, USA.
Email: aurko@gatech.edu
[2]ISyE, Georgia Institute of Technology, Atlanta, GA, USA.
Email: sebastian.pokutta@isye.gatech.edu

August 28, 2016

Abstract

We study the cost function for hierarchical clusterings introduced by [Dasgupta, 2016] where hierarchies are treated as first-class objects rather than deriving their cost from projections into flat clusters. It was also shown in [Dasgupta, 2016] that a top-down algorithm returns a hierarchical clustering of cost at most $O(\alpha_n \log n)$ times the cost of the optimal hierarchical clustering, where α_n is the approximation ratio of the Sparsest Cut subroutine used. Thus using the best known approximation algorithm for Sparsest Cut due to Arora-Rao-Vazirani, the top down algorithm returns a hierarchical clustering of cost at most $O\left(\log^{3/2} n\right)$ times the cost of the optimal solution. We improve this by giving an $O(\log n)$-approximation algorithm for this problem. Our main technical ingredients are a combinatorial characterization of ultrametrics induced by this cost function, deriving an Integer Linear Programming (ILP) formulation for this family of ultrametrics, and showing how to iteratively round an LP relaxation of this formulation by using the idea of *sphere growing* which has been extensively used in the context of graph partitioning. We also prove that our algorithm returns an $O(\log n)$-approximate hierarchical clustering for a generalization of this cost function also studied in [Dasgupta, 2016]. Experiments show that the hierarchies found by using the ILP formulation as well as our rounding algorithm often have better projections into flat clusters than the standard linkage based algorithms. We conclude with an inapproximability result for this problem, namely that no polynomial sized LP or SDP can be used to obtain a constant factor approximation for this problem.

Introduction

Hierarchical clustering is an important method in cluster analysis where a data set is recursively partitioned into clusters of successively smaller size. They are typically represented by rooted trees where the root corresponds to the entire data set, the leaves correspond to individual data points and the intermediate nodes correspond to a cluster of its descendant leaves. Such a hierarchy represents several possible *flat clusterings* of the data at various levels of granularity; indeed every pruning of this tree returns a possible clustering. Therefore in situations where the number of desired clusters is not known beforehand, a hierarchical clustering scheme is often preferred to flat clustering.

The most popular algorithms for hierarchical clustering are bottoms-up agglomerative algorithms like *single linkage*, *average linkage* and *complete linkage*. In terms of theoretical guarantees these algorithms are known to correctly recover a ground truth clustering if the similarity function on the data satisfies corresponding stability properties (see, e.g., [Balcan et al., 2008]). Often, however, one wishes to think of a good clustering as optimizing some kind of cost function rather than recovering a hidden "ground truth". This is the standard

approach in the classical clustering setting where popular objectives are k-means, k-median, min-sum and k-center (see Chapter 14, [Friedman et al., 2001]). However as pointed out by [Dasgupta, 2016] for a lot of popular hierarchical clustering algorithms including linkage based algorithms, it is hard to pinpoint explicitly the cost function that these algorithms are optimizing. Moreover, much of the existing cost function based approaches towards hierarchical clustering evaluate a hierarchy based on a cost function for flat clustering, e.g., assigning the k-means or k-median cost to a pruning of this tree. Motivated by this, [Dasgupta, 2016] introduced a cost function for hierarchical clustering where the cost takes into account the entire structure of the tree rather than just the projections into flat clusterings. This cost function is shown to recover the intuitively correct hierarchies on several synthetic examples like planted partitions and cliques. In addition, a top down graph partitioning algorithm is presented that outputs a tree with cost at most $O(\alpha_n \log n)$ times the cost of the optimal tree and where α_n is the approximation guarantee of the Sparsest Cut subroutine used. Thus using the Leighton-Rao algorithm [Leighton and Rao, 1988, Leighton and Rao, 1999] or the Arora-Rao-Vazirani algorithm [Arora et al., 2009] gives an approximation factor of $O\left(\log^2 n\right)$ and $O\left(\log^{3/2} n\right)$ respectively.

In this work we give a polynomial time algorithm to recover a hierarchical clustering of cost at most $O(\log n)$ times the optimal clustering according to this cost function. We also analyze a generalization of this cost function studied by [Dasgupta, 2016] and show that our algorithm still gives an $O(\log n)$ approximation in this setting. We do this by viewing the cost function in terms of the ultrametric it induces on the data, writing a convex relaxation for it and concluding by analyzing a popular rounding scheme used in graph partitioning algorithms. We also implement the integer program, its LP relaxation, and the rounding algorithm and test it on some synthetic and real world data sets to compare the cost of the rounded solutions to the true optimum as well as to compare its performance to other hierarchical clustering algorithms used in practice. Our experiments suggest that the hierarchies found by this algorithm are often better than the ones found by linkage based algorithms as well as the k-means algorithm in terms of the error of the best pruning of the tree compared to the ground truth.

Related Work

The immediate precursor to this work is [Dasgupta, 2016] where the cost function for evaluating a hierarchical clustering was introduced. Prior to this there has been a long line of research on hierarchical clustering in the context of phylogenetics and taxonomy (see, e.g., [Jardine and Sibson, 1971, Sneath et al., 1973, Felsenstein and Felenstein, 2004]). Several authors have also given theoretical justifications for the success of the popular linkage based algorithms for hierarchical clustering (see, e.g. [Jardine and Sibson, 1968, Zadeh and Ben-David, 2009, Ackerman et al., 2010]). In terms of cost functions, one approach has been to evaluate a hierarchy in terms of the k-means or k-median cost that it induces (see [Dasgupta and Long, 2005]). The cost function and the top-down algorithm in [Dasgupta, 2016] can also be seen as a theoretical justification for several graph partitioning heuristics that are used in practice.

Besides this prior work on hierarchical clustering we are also motivated by the long line of work in the classical clustering setting where a popular strategy is to study convex relaxations of these problems and to round an optimal fractional solution into an integral one with the aim of getting a good approximation to the cost function. A long line of work (see, e.g., [Charikar et al., 1999, Jain and Vazirani, 2001, Jain et al., 2003, Charikar and Li, 2012]) has employed this approach on LP relaxations for the k-median problem, including [Li and Svensson, 2013] which gives the best known approximation factor of $1+\sqrt{3}+\varepsilon$. Similarly, a few authors have studied LP and SDP relaxations for the k-means problem (see, e.g., [Peng and Xia, 2005, Peng and Wei, 2007, Awasthi et al., 2015]), while one of the best known algorithms for kernel k-means and spectral clustering is due to [Recht et al., 2012] which approximates the nonnegative matrix factorization (NMF) problem by LPs.

LP relaxations for hierarchical clustering have also been studied in [Ailon and Charikar, 2005] where the ob-

jective is to fit a tree metric to a data set given pairwise dissimalrities. While the LP relaxation and rounding algorithm in [Ailon and Charikar, 2005] is similar in flavor, the result is incomparable to ours (see Section 7 for a discussion). Another work that is indirectly related to our approach is [Di Summa et al., 2015] where the authors study an ILP to obtain a closest ultrametric to arbitrary functions on a discrete set. Our approach is to give a combinatorial characterization of the ultrametrics induced by the cost function of [Dasgupta, 2016] which allows us to use the tools from [Di Summa et al., 2015] to model the problem as an ILP. The natural LP relaxation of this ILP turns out to be closely related to LP relaxations considered before for several graph partitioning problems (see, e.g., [Leighton and Rao, 1988, Leighton and Rao, 1999, Even et al., 1999, Krauthgamer et al., 2009]) and we use a rounding technique studied in this context to round this LP relaxation.

Contribution

While studying convex relaxations of optimization problems is fairly natural, for the cost function introduced in [Dasgupta, 2016] however, it is not immediately clear how one would go about writing such a relaxation. Our first contribution is to give a combinatorial characterization of the family of ultrametrics induced by this cost function on hierarchies. Inspired by the approach in [Di Summa et al., 2015] where the authors study an integer linear program for finding the closest ultrametric, we are able to formulate the problem of finding the minimum cost hierarchical clustering as an integer linear program. Interestingly and perhaps unsurprisingly, the specific family of ultrametrics induced by this cost function give rise to linear constraints studied before in the context of finding balanced separators in weighted graphs. We then show how to round an optimal fractional solution using the *sphere growing* technique first introduced in [Leighton and Rao, 1988] (see also [Garg et al., 1996, Even et al., 1999, Charikar et al., 2003]) to recover a tree of cost at most $O(\log n)$ times the optimal tree for this cost function. The generalization of this cost function involves scaling every pairwise distances by an arbitrary strictly increasing function f satisfying $f(0) = 0$. We modify the integer linear program for this general case and show that the rounding algorithm still finds a hierarchical clustering of cost at most $O(\log n)$ times the optimal clustering in this setting. We also show a lower bound on the approximation ratios achievable by rounding convex relaxations of this problem. Specifically, we prove that every polynomial sized LP and SDP for this problem has an integrality gap that cannot be bounded by any constant not depending on n. We conclude with an experimental study of the integer linear program and the rounding algorithm on some synthetic and real world data sets to show that the approximation algorithm often recovers clusters close to the true optimum (according to this cost function) and that its projections into flat clusters often has a better error rate than the linkage based algorithms and the k-means algorithm.

Preliminaries

A similarity based clustering problem consists of a dataset V of n points and a *similarity function* $\kappa : V \times V \to \mathbb{R}_{\geq 0}$ such that $\kappa(i,j)$ is a measure of the similarity between i and j for any $i, j \in V$. We will assume that the similarity function is symmetric i.e., $\kappa(i,j) = \kappa(j,i)$ for every $i, j \in V$. Note that we do not make any assumptions about the points in V coming from an underlying metric space. For a given instance of a clustering problem we have an associated weighted complete graph K_n with vertex set V and weight function given by κ. A *hierarchical clustering* of V is a tree T with a designated root r and with the elements of V as its leaves, i.e., $\text{leaves}(T) = V$. For any set $S \subseteq V$ we denote the *lowest common ancestor* of S in T by $\text{lca}(S)$. For pairs of points $i, j \in V$ we will abuse the notation for the sake of simplicity and denote $\text{lca}(\{i,j\})$ simply by $\text{lca}(i,j)$. For a node v of T we denote the subtree of T rooted at v by $T[v]$. The following cost function was introduced by [Dasgupta, 2016] to measure the quality of the hierarchical clustering T

$$\text{cost}(T) := \sum_{\{i,j\} \in E(K_n)} \kappa(i,j) \, |\text{leaves}(T[\text{lca}(i,j)])| \, . \tag{1}$$

The intuition behind this cost function is as follows. Let T be a hierarchical clustering with designated root r so that r represents the whole data set V. Since $\text{leaves}(T) = V$, every internal node $v \in T$ represents a cluster of its descendant leaves, with the leaves themselves representing singleton clusters of V. Starting from r and going down the tree, every distinct pair of points $i, j \in V$ will be eventually separated at the leaves. If $\kappa(i,j)$ is large, i.e., i and j are very similar to each other then we would like them to be separated as far down the tree as possible if T is a good clustering of V. This is enforced in the cost function (1): if $\kappa(i,j)$ is large then the number of leaves of $\text{lca}(i,j)$ should be small i.e., $\text{lca}(i,j)$ should be far from the root r of T. Such a cost function is not unique however; see Section 7 for some other cost functions of a similar flavor. Note that while requiring κ to be non-negative might seem like an artificial restriction, the following lemma provides some justification for this choice. In particular Lemma 1 shows that analogous to the Minimum Spanning Tree (MST) problem, adding a large positive constant to every $\kappa(i,j)$ increases the cost of every hierarchical clustering T by the same amount.

Lemma 1. *Let $\kappa : V \times V \to \mathbb{R}$ be a similarity function on a data set V and $\kappa' : V \times V \to \mathbb{R}$ be obtained from κ as $\kappa'(i,j) := \kappa(i,j) + W$ for some $W \in \mathbb{R}$. Denote by $\text{cost}(T)$ and $\text{cost}'(T)$ the cost of a hierarchical clustering T of V according to similarity functions κ and κ' respectively. Then we have*

$$\text{cost}'(T) = \text{cost}(T) + \frac{W\left(n^3 - n\right)}{3},$$

with $|V| = n$.

Proof. Let $\text{cost}_{\mathbb{1}}(T)$ be the cost of any hierarchical clustering of V where the similarity between every pair $i, j \in V$ is 1. We have the following expression for $\text{cost}'(T)$ in terms of $\text{cost}_{\mathbb{1}}(T)$

$$\begin{aligned}\text{cost}'(T) &= \text{cost}(T) + W\left(\sum_{\{i,j\} \in E(K_n)} |\text{leaves}(T[\text{lca}(i,j)])|\right) \\ &= \text{cost}(T) + W \cdot \text{cost}_{\mathbb{1}}(T).\end{aligned}$$

By Theorem 3 of [Dasgupta, 2016], for any hierarchical clustering T on V we have $\text{cost}_{\mathbb{1}}(T) = \frac{n^3 - n}{3}$ from which the claim follows. □

While Lemma 1 guarantees that the optimal hierarchical clustering according to cost function (1) is translation invariant, it does not imply that an $O(\log n)$ approximate solution according to κ' is also an $O(\log n)$ approximate solution for κ. Therefore in the rest of this work we will assume that $\kappa \geq 0$. This is not a restriction compared to [Dasgupta, 2016], since the Sparsest Cut algorithm used as a subroutine in [Dasgupta, 2016] also requires this assumption. Let us now briefly recall the notion of an ultrametric.

Definition 2 (Ultrametric)**.** *An* ultrametric *on a set X of points is a distance function $d : X \times X \to \mathbb{R}$ satisfying the following properties for every $x, y, z \in X$*

1. ***Nonnegativity:*** *$d(x,y) \geq 0$ with $d(x,y) = 0$ iff $x = y$*
2. ***Symmetry:*** *$d(x,y) = d(y,x)$*
3. ***Strong triangle inequality:*** *$d(x,y) \leq \max\{d(y,z), d(z,x)\}$*

Under the cost function (1), one can interpret the tree T as inducing an ultrametric d_T on V given by $d_T(i,j) := |\text{leaves}(T[\text{lca}\,(i,j)])| - 1$. This is an ultrametric since $d_T(i,j) = 0$ iff $i = j$ and for any triple $i, j, k \in V$ we have $d_T(i,j) \leq \max\{d_T(i,k), d_T(j,k)\}$. The following definition introduces the notion of *non-trivial ultrametrics*. These turn out to be precisely the ultrametrics that are induced by tree decompositions of V corresponding to cost function (1), as we will show in Corollary 9.

Definition 3. *An ultrametric d on a set of points V is* non-trivial *if the following conditions hold.*

1. *For every non-empty set $S \subseteq V$, there is a pair of points $i, j \in S$ such that $d(i,j) \geq |S| - 1$.*
2. *For any t if S_t is an equivalence class of V under the relation $i \sim j$ iff $d(i,j) \leq t$, then $\max_{i,j \in S_t} d(i,j) \leq |S_t| - 1$.*

Note that for an equivalence class S_t where $d(i,j) \leq t$ for every $i, j \in S_t$ it follows from Condition 1 that $t \geq |S_t| - 1$. Thus in the case when $t = |S_t| - 1$ the two conditions imply that the maximum distance between any two points in S is t and that there is a pair $i, j \in S$ for which this maximum is attained. The following lemma shows that non-trivial ultrametrics behave well under restrictions to equivalence classes S_t of the form $i \sim j$ iff $d(i,j) \leq t$.

Lemma 4. *Let d be a non-trivial ultrametric on V and let $S_t \subseteq V$ be an equivalence class under the relation $i \sim j$ iff $d(i,j) \leq t$. Then d restricted to S_t is a non-trivial ultrametric on S_t.*

Proof. Clearly d restricted to S_t is an ultrametric on S_t and so we need to establish that it satisfies Conditions 1 and 2 of Definition 3. Let $S \subseteq S_t$ be any set. Since d is a non-trivial ultrametric on V it follows that there is a pair $i, j \in S$ with $d(i,j) \geq |S| - 1$, and so d restricted to S_t satsifies Condition 1. If S'_r is an equivalence class in S_t under the relation $i \sim j$ iff $d(i,j) \leq r$ then clearly $S'_r = S_t$ if $r > t$. Since d is a non-trivial ultrametric on V, it follows that $\max_{i,j \in S'_r} d(i,j) = \max_{i,j \in S_t} d(i,j) \leq |S_t| - 1 = |S'_r| - 1$. Thus we may assume that $r \leq t$. Consider an $i \in S'_r$ and let $j \in V$ be such that $d(i,j) \leq r$. Since $r \leq t$ and $i \in S_t$, it follows that $j \in S_t$ and so $j \in S'_r$. In other words S'_r is an equivalence class in V under the relation $i \sim j$ iff $d(i,j) \leq r$. Since d is an ultrametric on V it follows that $\max_{i,j \in S'_r} d(i,j) \leq |S'_r| - 1$. Thus d restricted to S_t satisfies Condition 2. □

The intuition behind the two conditions in Definition 3 is as follows. Condition 1 imposes a certain lower bound by ruling out trivial ultrametrics where, e.g., $d(i,j) = 0$ for every $i, j \in V$. On the other hand Condition 2 discretizes and imposes an upper bound on d by restricting its range to the set $\{0, 1, \ldots, n-1\}$ (see Lemma 5). This rules out the other spectrum of triviality where for example $d(i,j) = n$ for every $i, j \in V$ with $|V| = n$.

Lemma 5. *Let d be a non-trivial ultrametric on the set V as in Definition 3. Then the range of d is contained in the set $\{0, 1, \ldots, n-1\}$ with $|V| = n$.*

Proof. We will prove this by induction on $|V|$. The base case when $|V| = 1$ is trivial. By Condition 1 there is a pair $i, j \in V$ such that $d(i,j) \geq n - 1$. Let $t = \max_{i,j \in V} d(i,j)$, then the only equivalence class under the relation $i \sim j$ iff $d(i,j) \leq t$ is V. By Condition 2 it follows that $\max_{i,j \in V} d(i,j) = t = n - 1$. Let $V_1, \ldots V_m$ denote the set of equivalence classes of V under the relation $i \sim j$ iff $d(i,j) \leq n - 2$. Note that $m > 1$ and each $V_l \subsetneq V$ and by Lemma 4, d restricted to each of these V_i's is a non-trivial ultrametric on those sets. The claim then follows immediately: for any $i, j \in V$ either $i, j \in V_l$ for some V_l in which case by the induction hypothesis $d(i,j) \in \{0, 1, \ldots, |V_l| - 1\}$, or $i \in V_l$ and $j \in V_{l'}$ for $l \neq l'$ in which case $d(i,j) = n - 1$. □

Ultrametrics and Hierarchical Clusterings

We start with the following easy lemma about the lowest common ancestors of subsets of V in a hierarchical clustering T of V.

Lemma 6. *Let $S \subseteq V$ of size ≥ 2. If $r = \mathrm{lca}(S)$ then there is a pair $i, j \in S$ such that $\mathrm{lca}(i,j) = r$.*

Proof. We will proceed by induction on $|S|$. If $|S| = 2$ then the claim is trivial and so we may assume $|S| > 2$. Let $i \in S$ be an arbitrary point and let $r' = \text{lca}(S \setminus \{i\})$. We claim that $r = \text{lca}(i, r')$. Clearly the subtree rooted at $\text{lca}(i, r')$ contains S and since $T[r]$ is the smallest such tree it follows that $r \in T[\text{lca}(i, r')]$. Conversely, $T[r]$ contains $S \setminus \{i\}$ and so $r' \in T[r]$ and since $i \in T[r]$, it follows that $\text{lca}(i, r') \in T[r]$ since $T[\text{lca}(i, r')]$ is the smallest subtree containing i and r'. Thus we may conclude that $r = \text{lca}(i, r')$.
If $\text{lca}(i, r') = r'$, then we are done by the inductive hypothesis. Thus we may assume that $i \notin T[r']$. Consider any $j \in S$ such that $j \in T[r']$. Then we have that $\text{lca}(i, j) = r$ as $\text{lca}(i, r') = r$ and $j \in T[r']$ and $i \notin T[r']$. □

We will now show that non-trivial ultramerics on V as in Definition 3 are exactly those that are induced by hierarchical clusterings on V under cost function (1). The following lemma shows the forward direction: the ultrametric d_T induced by any hierarchical clustering T is non-trivial.

Lemma 7. *Let T be a hierarchical clustering on V and let d_T be the ultrametric on V induced by it. Then d_T is non-trivial as in Definiton 3.*

Proof. Let $S \subseteq V$ be arbitrary and $r = \text{lca}(S)$, then $T[r]$ has at least $|S|$ leaves. By Lemma 6 there must be a pair $i, j \in S$ such that $r = \text{lca}(i, j)$ and so $d_T(i, j) \geq |S| - 1$. This satisfies Condition 1 of non-triviality.
For any t, let S_t be a non-empty equivalence class under the relation $i \sim j$ iff $d_T(i, j) \leq t$. Since d_T satisfies Condition 1 it follows that $|S_t| - 1 \leq t$. Let us assume for the sake of contradiction that there is a pair $i, j \in S_t$ such that $d_T(i, j) > |S_t| - 1$. Let $r = \text{lca}(S_t)$; using the definition of d_T it follows that $t + 1 \geq |\text{leaves}(T[r])| > |S_t|$ since $i, j \in S_t$. Let $k \in \text{leaves}(T[r]) \setminus S_t$ be an arbitrary point, then for every $l \in S_t$ it follows that $d_T(k, l) \leq |\text{leaves}(T[r])| - 1 \leq t$ since the subtree rooted at r contains both k and l. This is a contradiction to S_t being an equivalence class under $i \sim j$ iff $d_T(i, j) \leq t$ since $k \notin S_t$. Thus d_T also satisfies Condition 2 of Definition 3. □

The following crucial lemma shows the converse: every non-trivial ultrametric on V is realized by a hierarchical clustering T of V.

Lemma 8. *For every non-trivial ultrametric d on V there is a hierarchical clustering T on V such that for any pair $i, j \in V$ we have*

$$d_T(i, j) = |\text{leaves}(T[\text{lca}(i, j)])| - 1 = d(i, j).$$

Moreover this hierarchy can be constructed in time $O(n^2)$ by Algorithm 1 where $|V| = n$.

Proof. We will use induction on n. The base case when $n = 1$ is straightforward. We now suppose that the statement is true for sets of size $< n$. Note that $i \sim j$ iff $d(i, j) \leq n - 2$ is an equivalence relation on V and thus partitions V into m equivalence classes $V_1, \ldots, V_m$. We first observe that $m > 1$ since by Condition 1 there is a pair of points $i, j \in V$ such that $d(i, j) \geq n - 1$ and in particular $|V|_l < n$ for every $l \in \{1, \ldots, m\}$. By Lemma 4 d restricted to any V_l is a non-trivial ultrametric on V_l and there is a pair of points $i, j \in V_l$ such that $d(i, j) = |V_l| - 1$ by Conditions 1 and 2. Therefore by the inductive hypothesis we construct trees $T_1, \ldots, T_m$ such that for every $l \in \{1, \ldots, m\}$ we have $\text{leaves}(T_l) = V_l$. Further for any pair of points $i, j \in V_l$ for some $l \in \{1, \ldots, m\}$, we also have $d(i, j) = d_{T_l}(i, j)$.
Thus we construct the tree T as follows: we first add a root r and then connect the root r_l of T_l to r for every $l \in \{1, \ldots, m\}$. Consider a pair of points $i, j \in V$. If $i, j \in V_l$ for some $l \in \{1, \ldots, m\}$ then we are done since $d_{T_l}(i, j) = d_T(i, j)$ as $\text{lca}(i, j) \in T_l$. If $i \in V_l$ and $j \in V_{l'}$ for some $l \neq l'$ then $d(i, j) = n - 1$ since $d(i, j) \geq n - 1$ by definition of the equivalence relation and the range of d lies in $\{0, 1, \ldots, n - 1\}$ by Lemma 5. Moreover i and j are leaves in T_l and $T_{l'}$ respectively, and thus by construction of T we have $\text{lca}(i, j) = r$ i.e., $d_T(i, j) = n - 1$ and so the claim follows. Algorithm 1 simulates this inductive argument can be easily implemented to run in time $O(n^2)$. □

Lemmas 7 and 8 together imply the following corollary about the equivalence of hierarchical clusterings and non-trivial ultrametrics.

Corollary 9. *There is a bijection between the set of hierarchical clusterings T on V and the set of non-trivial ultrametrics d on V satisfying the following conditions.*

1. *For every hierarchical clustering T on V, there is a non-trivial ultrametric d_T defined as $d_T(i,j) \coloneqq |\text{leaves } T[\text{lca}(i,j)]| - 1$ for every $i, j \in V$.*
2. *For every non-trivial ultrametric d on V, there is a hierarchical clustering T on V such that for every $i, j \in V$ we have $|\text{leaves } T[\text{lca}(i,j)]| - 1 = d(i,j)$.*

Moreover this bijection can be computed in $O(n^2)$ time, where $|V| = n$.

Input: Data set V of n points, non-trivial ultrametric $d : V \times V \to \mathbb{R}_{\geq 0}$
Output: Hierarchical clustering T of V with root r

$r \leftarrow$ arbitrary choice of designated root in V
$X \leftarrow \{r\}$
$E \leftarrow \emptyset$
if $n = 1$ **then**
  $T \leftarrow (X, E)$
  return r, T
else
  Partition V into $\{V_1, \ldots V_m\}$ under the equivalence relation $i \sim j$ iff $d(i,j) < n-1$
  for $l \in \{1, \ldots, m\}$ **do**
    Let r_l, T_l be output of Algorithm 1 on V_l, $d|_{V_l}$
    $X \leftarrow X \cup V(T_l)$
    $E \leftarrow E \cup \{r, r_l\}$
  end
  $T \leftarrow (X, E)$
  return r, T
end

Algorithm 1: Hierarchical clustering of V from non-trivial ultrametric

Therefore to find the hierarchical clustering of minimum cost, it suffices to minimize $\langle \kappa, d \rangle$ over non-trivial ultrametrics $d : V \times V \to \{0, \ldots, n-1\}$, where V is the data set. A natural approach is to formulate this problem as an Integer Linear Program (ILP) and then study LP or SDP relaxations of it. We consider the following ILP for this problem that is motivated by [Di Summa et al., 2015]. We have the variables $x_{ij}^1, \ldots, x_{ij}^{n-1}$ for every distinct pair $i, j \in V$ with $x_{ij}^t = 1$ if and only if $d(i,j) \geq t$. For any positive integer n, let $[n] \coloneqq \{1, 2, \ldots, n\}$.

$$\min \quad \sum_{t=1}^{n-1} \sum_{\{i,j\} \in E(K_n)} \kappa(i,j) x_{ij}^t \tag{ILP-ultrametric}$$

$$\text{s.t.} \quad x_{ij}^t \geq x_{ij}^{t+1} \qquad \forall i,j \in V, t \in [n-2] \tag{2}$$

$$x_{ij}^t + x_{jk}^t \geq x_{ik}^t \qquad \forall i,j,k \in V, t \in [n-1] \tag{3}$$

$$\sum_{i,j \in S} x_{ij}^t \geq 2 \qquad \forall t \in [n-1], S \subseteq V, |S| = t+1 \tag{4}$$

$$\sum_{i,j \in S} x_{ij}^{|S|} \leq |S| \left(\sum_{i,j \in S} x_{ij}^t + \sum_{\substack{i \in S \\ j \notin S}} \left(1 - x_{ij}^t\right) \right) \forall t \in [n-1], S \subseteq V \tag{5}$$

$$x_{ij}^t = x_{ji}^t \qquad \forall i,j \in V, t \in [n-1] \tag{6}$$

$$x_{ij}^t \in \{0,1\} \qquad \forall i,j \in V, t \in [n-1] \tag{7}$$

Note that constraint 3 is the same as the *strong triangle inequality* (Definition 2) since the variables x_{ij}^t are in $\{0,1\}$. Constraint 6 ensures that the ultrametric is symmetric. Constraint 4 ensures the ultrametric satisfies Condition 1 of non-triviality: for every $S \subseteq V$ of size $t+1$ we know that there must be points $i,j \in S$ such that $d(i,j) = d(j,i) \geq t$ or in other words $x_{ij}^t = x_{ji}^t = 1$. Constraint 5 ensures that the ultrametric satisfies Condition 2 of non-triviality. To see this note that the constraint is active only when $\sum_{i,j \in S} x_{ij}^t = 0$ and $\sum_{i \in S, j \notin S} (1 - x_{ij}^t) = 0$. In other words $d(i,j) < t$ for every $i,j \in S$ and S is a maximal such set since if $i \in S$ and $j \notin S$ then $d(i,j) \geq t$. Thus S is an equivalence class under the relation $i \sim j$ iff $d(i,j) < t$ and so for every $i,j \in S$ we have $d(i,j) \leq |S| - 1$ or equivalently $x_{ij}^{|S|} = 0$. The ultrametric d represented by a feasible solution x_{ij}^t is given by $d(i,j) = \sum_{t=1}^{n-1} x_{ij}^t$.

Definition 10. *For any* $\left\{ x_{ij}^t \mid t \in [n-1], i,j \in V \right\}$ *let* E_t *be defined as* $E_t := \left\{ \{i,j\} \mid x_{ij}^t = 0 \right\}$. *Note that if* x_{ij}^t *is feasible for ILP-ultrametric then* $E_t \subseteq E_{t+1}$ *for any* t *since* $x_{ij}^t \geq x_{ij}^{t+1}$. *The sets* $\{E_t\}_{t=1}^{n-1}$ *induce a natural sequence of graphs* $\{G_t\}_{t=1}^{n-1}$ *where* $G_t = (V, E_t)$ *with* V *being the data set.*

For a fixed $t \in \{1, \ldots, n-1\}$ it is instructive to study the combinatorial properties of the so called *layer-t problem*, where we restrict ourselves to the constraints corresponding to that particular t.

$$\min \quad \sum_{\{i,j\} \in E(K_n)} \kappa(i,j) x_{ij}^t \tag{ILP-layer}$$

$$\text{s.t.} \quad x_{ij}^t + x_{jk}^t \geq x_{ik}^t \qquad \forall i,j,k \in V \tag{8}$$

$$\sum_{i,j \in S} x_{ij}^t \geq 2 \qquad \forall S \subseteq V, |S| = t+1 \tag{9}$$

$$x_{ij}^t = x_{ji}^t \qquad \forall i,j \in V \tag{10}$$

$$x_{ij}^t \in \{0,1\} \qquad \forall i,j \in V \tag{11}$$

The following lemma provides a combinatorial characterization of feasible solutions to the layer-t problem.

Lemma 11. *Let* $G_t = (V, E_t)$ *be the graph as in Definition 10 corresponding to a solution* x_{ij}^t *to the layer-t problem ILP-layer. Then* G_t *is a disjoint union of cliques of size* $\leq t$. *Moreover this exactly characterizes all feasible solutions of ILP-layer.*

Proof. We first note that $G_t = (V, E_t)$ must be a disjoint union of cliques since if $\{i,j\} \in E_t$ and $\{j,k\} \in E_t$ then $\{i,k\} \in E_t$ since $x^t_{ik} \leq x^t_{ij} + x^t_{jk} = 0$ due to constraint 8. Suppose there is a clique in G_t of size $> t$. Choose a subset S of this clique of size $t+1$. Then $\sum_{i,j \in S} x^t_{ij} = 0$ which violates constraint 9.
Conversely, let E_t be a subset of edges such that $G_t = (V, E_t)$ is a disjoint union of cliques of size $\leq t$. Let $x^t_{ij} = 0$ if $\{i,j\} \in E_t$ and 1 otherwise. Clearly $x^t_{ij} = x^t_{ji}$ by definition. Suppose x^t_{ij} violates constraint 8, so that there is a pair $i,j,k \in V$ such that $x^t_{ik} = 1$ but $x^t_{ij} = x^t_{jk} = 0$. However this implies that the complement graph is not a disjoint union of cliques since $\{i,j\}, \{j,k\} \in E_t$ but $\{i,k\} \notin E_t$. Suppose x^t_{ij} violates constraint 9 for some set S of size $t+1$. This implies that for every $i,j \in S$, $x^t_{ij} = 0$ (since $x^t_{ij} = x^t_{ji}$ for every $i,j \in V$) and so S must be a clique of size $t+1$ in the complement graph which is a contradiction. □

By Lemma 11 the layer-t problem is to find a subset $\overline{E}_t \subseteq E(K_n)$ of minimum weight under κ, such that the complement graph $G_t = (V, E_t)$ is a disjoint union of cliques of size $\leq t$. Note that this implies that the number of components in the complement graph is $\geq \lceil n/t \rceil$.The converse however, is not necessarily true: when $t = n-1$ then the layer t-problem is the minimum (weighted) cut problem whose partitions may have size larger than 1. Our algorithmic approach is to solve an LP relaxation of ILP-ultrametric and then round the solution to obtain a feasible solution to ILP-ultrametric. The rounding however proceeds iteratively in a layer-wise manner and so we need to make sure that the rounded solution satisfies the inter-layer constraints (2) and (5). The following lemma gives a combinatorial characterization of solutions that satisfy these two constraints.

Lemma 12. *For every $t \in [n-1]$, let x^t_{ij} be feasible for the layer-t problem ILP-layer. Let $G_t = (V, E_t)$ be the graph as in Definition 10 corresponding to x^t_{ij}, so that by Lemma 11, G_t is a disjoint union of cliques $K^t_1, \ldots, K^t_{l_t}$ each of size at most t. Then x^t_{ij} is feasible for ILP-ultrametric if and only if the following conditions hold.*

Nested cliques *For any $s \leq t$ every clique K^s_p for some $p \in [l_s]$ in G_s is a subclique of some clique K^t_q in G_t where $q \in [l_t]$.*

Realization *If $\left|K^t_p\right| = s$ for some $s \leq t$, then G_s contains K^t_p as a component clique i.e., $K^s_q = K^t_p$ for some $q \in [l_s]$.*

Proof. Since x^t_{ij} is feasible for the layer-t problem ILP-layer it is feasible for ILP-ultrametric if and only if it satisfies constraints (2) and (5). The solution x^t_{ij} satisfies constraint (2) if and only if $E_t \subseteq E_{t+1}$ by definition and so Condition Nested cliques follows.
Let us now assume that x^t_{ij} is feasible for ILP-ultrametric, so that by the above argument Condition Nested cliques is satisfied. Constraint (5) ensures that an equivalence class $S_t \subseteq V$ of the relation $i \sim j$ iff $x^t_{ij} = 0$ must satisfy $x^{|S_t|}_{ij} = 0$ for every $i,j \in S_t$. Note that the set S_t corresponds to a clique K^t_p in G_t for some $p \in [l_t]$ and constraint (2) is equivalent to S_t being a subclique of some clique K^s_q where $s = |S_t| \leq t$ and $q \in [l_s]$. However since x^s_{ij} is feasible for the layer-s problem it follows that $S_t = K^s_q$, i.e., Condition Realization is satisfied.
Conversely, suppose that x^t_{ij} satisfies Conditions Nested cliques and Realization, so that by the argument in the paragraph above x^t_{ij} satisfies constraint (2). Let us assume for the sake of contradiction that for a set $S \subseteq V$ and a $t \in [n-1]$ constraint (5) is violated, i.e.,

$$\sum_{i,j \in S} x^{|S|}_{ij} > |S| \left(\sum_{i,j \in S} x^t_{ij} + \sum_{\substack{i \in S \\ j \notin S}} \left(1 - x^t_{ij}\right) \right).$$

Since $x_{ij}^t \in \{0,1\}$ it follows that $x_{ij}^t = 0$ for every $i,j \in S$ and $x_{ij}^t = 1$ for every $i \in S, j \notin S$ so that S is a clique in G_t. Note that $|S| < t$ since $\sum_{i,j\in S} x_{ij}^{|S|} > 0$. This contradicts Condition Realization however, since S is clearly not a clique in $G_{|S|}$. □

The combinatorial interpretation of the individual layer-t problems allow us to simplify the formulation of ILP-ultrametric by replacing the constraints for sets of a specific size (constraint 4) by a global constraint about all sets (constraint 12).

Lemma 13. *We may replace constraint 4 of ILP-ultrametric by the following equivalent constraint*

$$\sum_{j\in S} x_{ij}^t \geq |S| - t \qquad \forall t \in [n-1], S \subseteq V, i \in S. \tag{12}$$

Proof. Let x_{ij}^t be a feasible solution to ILP-ultrametric. Note that if $|S| \leq t$ then the constraints are redundant since $x_{ij}^t \in \{0,1\}$. Thus we may assume that $|S| > t$ and let i be any vertex in S. Let us suppose for the sake of a contradiction that $\sum_{j\in S} x_{ij}^t < |S| - t$. This implies that there is a $t+1$ sized subset $S' \subseteq S \setminus \{i\}$ such that for every $j \in S'$, we have $x_{ij'}^t = 0$. In other words $\{i, j'\}$ is an edge in $G_t = (V, E_t)$ for every $j' \in S'$ and since G_t is a disjoint union of cliques (constraint 3), this implies the existence of a clique of size $t+1$. Thus by Lemma 11, x_{ij}^t could not have been a feasible solution to ILP-ultrametric.
Conversely, suppose x_{ij}^t is feasible for the modified ILP where constraint 4 is replaced by constraint 12. Then again the $G_t = (V, E_t)$ is a disjoint union of cliques since x_{ij}^t satisfies constraint 3. Assume for contradiction that constraint (4) is violated: there is a set S of size $t+1$ such that $\sum_{i,j\in S} x_{ij}^t < 2$. Note that this implies that $\sum_{i,j} x_{ij}^t = 0$ since $x_{ij}^t = x_{ji}^t$ for every $i,j \in V$ and $t \in [n-1]$. Fix any $i \in S$, then $\sum_{j\in S} x_{ij}^t < 1 = |S| - t$ since $x_{ij}^t = x_{ji}^t$ by constraint 6, a violation of constraint (12). Thus x_{ij}^t is feasible for ILP-ultrametric since it satisfies every other constraint by assumption. □

Example 14. *As a simple example let us consider the following hierarchical clustering problem on the points* $\{x_0, x_1, x_2, x_3, x_4, x_5\}$ *in* $\mathbb{R}^4$.

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 1.2 & 1.3 & 1 \\ 0.9 & 1 & 1.0 & 0.8 \\ -1 & -1 & -1.2 & -1.33 \\ -1.3 & -0.9 & -1.0 & -0.7 \\ -5000 & -5000 & -1.0 & -1.0 \\ -5000 & -5000 & -0.999 & -0.8 \end{bmatrix}$$

If we use cosine similarity as the similarity function, i.e., $\kappa(x,y) = \frac{\langle x,y\rangle}{\|x\|_2\|y\|_2}$ *then we expect* $\{x_0, x_1\}$, $\{x_2, x_3\}$ *and* $\{x_4, x_5\}$ *as natural clusters. At a less granular level* $\{x_2, x_3, x_4, x_5\}$ *together with* $\{x_0, x_1\}$ *also seems like an acceptable clustering. This intuition is confirmed by actually solving ILP-ultrametric via an IP solver to find the optimal non-trivial ultrametric for this problem. The tree corresponding to the optimal ultrametric is shown in Figure 1.*

Rounding an LP relaxation

In this section we consider the following natural LP relaxation for ILP-ultrametric. We keep the variables x_{ij}^t for every $t \in [n-1]$ and $i,j \in V$ but relax the integrality constraint on the variables as well as drop constraint (5).

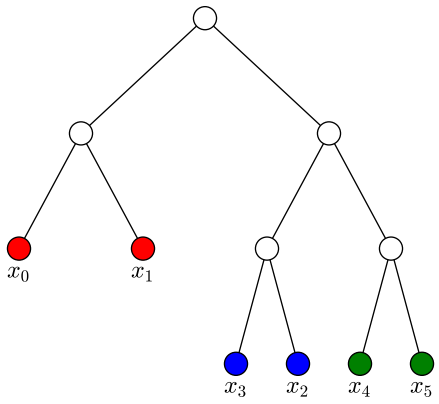

Figure 1: Optimal hierarchical clustering using cosine similarity on Example 14

$$\min \quad \sum_{t=1}^{n-1} \sum_{\{i,j\} \in E(K_n)} \kappa(i,j) x_{ij}^t \qquad \text{(LP-ultrametric)}$$

$$\text{s.t.} \quad x_{ij}^t \geq x_{ij}^{t+1} \qquad \forall i,j \in V, t \in [n-2] \tag{13}$$

$$x_{ij}^t + x_{jk}^t \geq x_{ik}^t \qquad \forall i,j,k \in V, t \in [n-1] \tag{14}$$

$$\sum_{j \in S} x_{ij}^t \geq |S| - t \qquad \forall t \in [n-1], S \subseteq V, i \in S \tag{15}$$

$$x_{ij}^t = x_{ji}^t \qquad \forall i,j \in V, t \in [n-1] \tag{16}$$

$$0 \leq x_{ij}^t \leq 1 \qquad \forall i,j \in V, t \in [n-1] \tag{17}$$

A feasible solution x_{ij}^t to LP-ultrametric induces a sequence $\{d_t\}_{t \in [n-1]}$ of distance metrics over V defined as $d_t(i,j) := x_{ij}^t$ if $i \neq j$ and $d_t(i,j) := 0$ if $i = j$. Constraint 15 is an additional structure on this metric: informally points in a "large enough" subset S should be spread apart according to the metric d_t. Metrics of type d_t are called *spreading metrics* and were first studied in [Even et al., 1999, Even et al., 2000] in relation to graph partitioning problems. The following lemma gives a technical interpretation of spreading metrics (see, e.g., [Even et al., 1999, Even et al., 2000, Krauthgamer et al., 2009]); we include a proof for completeness.

Lemma 15. *Let x_{ij}^t be feasible for LP-ultrametric and for a fixed $t \in [n-1]$, let d_t be the induced spreading metric. Let $i \in V$ be an arbitrary vertex and let $S \subseteq V$ be a set with $i \in S$ such that $|S| > (1+\varepsilon)t$ for some $\varepsilon > 0$. Then $\max_{j \in S} d_t(i,j) > \frac{\varepsilon}{1+\varepsilon}$.*

Proof. For the sake of a contradiction suppose that for every $j \in S$ we have $d_t(i,j) = x_{ij}^t \leq \frac{\varepsilon}{1+\varepsilon}$. This implies that x_{ij}^t violates constraint 15 leading to a contradiction:

$$\sum_{j \in S} x_{ij}^t \leq \frac{\varepsilon}{1+\varepsilon} |S| < |S| - t,$$

where the last inequality follows from $|S| > (1+\varepsilon)t$. □

The following lemma shows that we can optimize over LP-ultrametric in polynomial time.

Lemma 16. *An optimal solution to LP-ultrametric can be computed in time polynomial in n and $\log\left(\max_{i,j}\kappa(i,j)\right)$.*

Proof. We argue in the standard fashion via the application of the Ellipsoid method [Schrijver, 1998]. As such it suffices to verify that the encoding length of the numbers is small (which is indeed the case here) and that the constraints can be separated in polynomial time in the size of the input, i.e., in n and the logarithm of the absolute value of the largest coefficient. Since constraints of type 13, 14 and 16 are polynomially many in n, we only need to check separation for constraints of type 15. Given a claimed solution x_{ij}^t we can check constraint 15 by iterating over all $t \in [n-1]$, vertices $i \in V$ and sizes m of the set S from $t+1$ to n. For a fixed t, i and set size m sort the vertices in $V \setminus \{i\}$ in increasing order of distance from i (according to the metric d_t) and let S_t be the first m vertices in this ordering. If $\sum_{j\in S_t} x_{ij}^t < m - t$ then clearly x_{ij}^t is not feasible for LP-ultrametric, so we may assume that $\sum_{j\in S_t} x_{ij}^t \geq m - t$. Moreover this is the only set to check: for any set $S \subseteq V$ containing i such that $|S| = m$, $\sum_{j\in S} x_{ij}^t \geq \sum_{j\in S_t} x_{ij}^t \geq m - t$. Thus for a fixed $t \in [n-1]$, $i \in V$ and set size m, it suffices to check that x_{ij}^t satisfies constraint 15 for this subset S_t. □

From now on we will simply refer to a feasible solution to LP-ultrametric by the sequence of spreading metrics $\{d_t\}_{t\in[n-1]}$ it induces. The following definition introduces the notion of an open ball $\mathcal{B}_U(i,r,t)$ of radius r centered at $i \in V$ according to the metric d_t and restricted to the set $U \subseteq V$.

Definition 17. *Let $\{d_t \mid t \in [n-1]\}$ be the sequence of spreading metrics feasible for LP-ultrametric. Let $U \subseteq V$ be an arbitrary subset of V. For a vertex $i \in U$, $r \in \mathbb{R}$, and $t \in [n-1]$ we define the* open ball $\mathcal{B}_U(i,r,t)$ *of radius r centered at i as*

$$\mathcal{B}_U(i,r,t) := \{j \in U \mid d_t(i,j) < r\}.$$

If $U = V$ then we denote $\mathcal{B}_U(i,r,t)$ simply by $\mathcal{B}(i,r,t)$.

Remark 18. *For every pair $i, j \in V$ we have $d_t(i,j) \geq d_{t+1}(i,j)$ by constraint (13). Thus for any subset $U \subseteq V$, $i \in U$, $r \in \mathbb{R}$, and $t \in [n-2]$, it holds $\mathcal{B}_U(i,r,t) \subseteq \mathcal{B}_U(i,r,t+1)$.*

To round LP-ultrametric to get a feasible solution for ILP-ultrametric, we will use the technique of *sphere growing* which was introduced in [Leighton and Rao, 1988] to show an $O(\log n)$ approximation for the maximum multicommodity flow problem. Recall from Lemma 11 that a feasible solution to ILP-layer consists of a decomposition of the graph G_t into a set of disjoint cliques of size at most t. One way to obtain such a decomposition is to choose an arbtirary vertex, grow a ball around this vertex until the expansion of this ball is below a certain threshold, chop off this ball and declare it as a partition and then recurse on the remaining vertices. This is the main idea behind sphere growing, and the parameters are chosen depending on the constraints of the specific problem (see, e.g., [Garg et al., 1996, Even et al., 1999, Charikar et al., 2003] for a few representative applications of this technique). The first step is to associate to every ball $\mathcal{B}_U(i,r,t)$ a volume $\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)$ and a boundary $\partial\mathcal{B}_U(i,r,t)$ so that its expansion is defined. For any $t \in [n-1]$ and $U \subseteq V$ we denote by γ_t^U the value of the layer-t objective for solution d_t restricted to the set U, i.e.,

$$\gamma_t^U := \sum_{\substack{i,j\in U \\ i<j}} \kappa(i,j) d_t(i,j).$$

When $U = V$ we refer to γ_t^U simply by γ_t. Since $\kappa : V \times V \to \mathbb{R}_{\geq 0}$, it follows that $\gamma_t^U \leq \gamma_t$ for any $U \subseteq V$. We are now ready to define the volume, boundary and expansion of a ball $\mathcal{B}_U(i,r,t)$. We use the definition of [Even et al., 1999] modified for restrictions to arbitrary subsets $U \subseteq V$.

Definition 19. *[Even et al., 1999] Let U be an arbitrary subset of V. For a vertex $i \in U$, radius $r \in \mathbb{R}$, and $t \in [n-1]$, let $\mathcal{B}_U(i,r,t)$ be the ball of radius r as in Definition 17. Then we define its* volume *as*

$$\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right) := \frac{\gamma_t^U}{n \log n} + \sum_{\substack{j,k \in \mathcal{B}_U(i,r,t) \\ j<k}} \kappa(j,k) d_t(j,k) + \sum_{\substack{j \in \mathcal{B}_U(i,r,t) \\ k \notin \mathcal{B}_U(i,r,t) \\ k \in U}} \kappa(j,k)\left(r - d_t(i,j)\right).$$

The boundary *of the ball $\partial\mathcal{B}_U(i,r,t)$ is the partial derivative of volume with respect to the radius:*

$$\partial\mathcal{B}_U(i,r,t) := \frac{\partial\, \mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)}{\partial r} = \sum_{\substack{j \in \mathcal{B}_U(i,r,t) \\ k \notin \mathcal{B}_U(i,r,t) \\ k \in U}} \kappa(j,k).$$

The expansion *$\phi(\mathcal{B}_U(i,r,t))$ of the ball $\mathcal{B}_U(i,r,t)$ is defined as the ratio of its boundary to its volume, i.e.,*

$$\phi\left(\mathcal{B}_U(i,r,t)\right) := \frac{\partial\mathcal{B}_U(i,r,t)}{\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)}.$$

Input: Data set V, $\{d_t\}_{t\in[n-1]} : V \times V$, ε, $\kappa : V \times V \to \mathbb{R}_{\geq 0}$

Output: A solution set of the form $\left\{x_{ij}^t \in \{0,1\} \mid t \in \left[\left\lfloor \frac{n-1}{1+\varepsilon} \right\rfloor\right], i,j \in V\right\}$

$m_\varepsilon \leftarrow \left\lfloor \frac{n-1}{1+\varepsilon} \right\rfloor$
$t \leftarrow m_\varepsilon$
$\mathcal{C}_{t+1} \leftarrow \{V\}$
$\Delta \leftarrow \frac{\varepsilon}{1+\varepsilon}$
while $t \geq 1$ **do**
  $\mathcal{C}_t \leftarrow \emptyset$
  for $U \in \mathcal{C}_{t+1}$ **do**
    if $|U| \leq (1+\varepsilon)t$ **then**
      $\mathcal{C}_t \leftarrow \mathcal{C}_t \cup \{U\}$
      Go to line 7
    end
    while $U \neq \emptyset$ **do**
      Let i be arbitrary in U
      Let $r \in (0, \Delta]$ be s.t. $\phi\left(\mathcal{B}_U(i,r,t)\right) \leq \frac{1}{\Delta} \log\left(\frac{\mathrm{vol}(\mathcal{B}_U(i,\Delta,t))}{\mathrm{vol}(\mathcal{B}_U(i,0,t))}\right)$
      $\mathcal{C}_t \leftarrow \mathcal{C}_t \cup \{\mathcal{B}_U(i,r,t)\}$
      $U \leftarrow U \setminus \mathcal{B}_U(i,r,t)$
    end
  end
  $x_{ij}^t = 1$ if $i \in U_1 \in \mathcal{C}_t$, $j \in U_2 \in C_t$ and $U_1 \neq U_2$, else $x_{ij}^t = 0$
  $t \leftarrow t-1$
end
return $\left\{x_{ij}^t \mid t \in [m_\varepsilon], i,j \in V\right\}$

Algorithm 2: Iterative rounding algorithm to find a low cost ultrametric

The following theorem establishes that the rounding procedure of Algorithm 2 ensures that the cliques in $\mathcal{C}_t$ are "small" and that the cost of the edges removed to form them are not too high. It also shows that Algorithm 2 can be implemented to run in time polynomial in n. Let $m_\varepsilon := \left\lfloor \frac{n-1}{1+\varepsilon} \right\rfloor$ as in Algorithm 2.

Theorem 20. *Let* $\left\{ x^t_{ij} \mid t \in [m_\varepsilon], i,j \in V \right\}$ *be the output of Algorithm 2 on a feasible solution* $\{d_t\}_{t\in[n-1]}$ *of LP-ultrametric and any choice of* $\varepsilon \in (0,1)$. *For any* $t \in [m_\varepsilon]$, x^t_{ij} *is feasible for the layer-*$\lfloor (1+\varepsilon)\, t \rfloor$ *problem ILP-layer and there is a constant* $c(\varepsilon) > 0$ *depending only on* ε *such that*

$$\sum_{\{i,j\}\in E(K_n)} \kappa(i,j) x^t_{ij} \leq c(\varepsilon)(\log n)\gamma_t.$$

Moreover, Algorithm 2 can be implemented to run in time polynomial in n.

Proof. We first show that for a fixed t, the constructed solution x^t_{ij} is feasible for the layer-$\lfloor (1+\varepsilon)t \rfloor$ problem ILP-layer. Let $\mathcal{C}_t$ be as in Algorithm 2 so that $x^t_{ij} = 1$ if i,j belong to different sets in $\mathcal{C}_t$ and $x^t_{ij} = 0$ otherwise. Note that for any $t \in [m_\varepsilon]$, every $V_i \in \mathcal{C}_t$ is a clique by construction (line 19) and for every distinct pair $V_i, V_j \in \mathcal{C}_t$ we have $V_i \cap V_j = \emptyset$ (lines 15 and 16). Therefore by Lemma 11, it suffices to prove that for any $V_i \in \mathcal{C}_t$, it holds $|V_i| \leq \lfloor (1+\varepsilon)t \rfloor$. If V_i is added to $\mathcal{C}_t$ in line 9 then there is nothing to prove.
Thus let us assume that V_i is of the form $\mathcal{B}_U(i,r,t)$ for some $U \subseteq V$ as in line 14 so that $\phi\left(\mathcal{B}_U(i,r,t)\right) \leq \frac{1}{\Delta}\log\left(\frac{\mathrm{vol}(\mathcal{B}_U(i,\Delta,t))}{\mathrm{vol}(\mathcal{B}_U(i,0,t))}\right)$. Note that by Lemma 15 it suffices to show that there is such an $r \in (0,\Delta]$. This property follows from the rounding scheme due to [Even et al., 1999] as we will explain now. First note that for any fixed $U \subseteq V$, $\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)$ is a monotone non-decreasing function in r since for a pair $j,k \in U$ such that $j \in \mathcal{B}_U(i,r,t)$ and $k \notin \mathcal{B}_U(i,r,t)$ we have $r - d_t(i,j) \leq d_t(j,k)$ otherwise $r - d_t(i,j) > d_t(j,k)$ so that $r > d_t(i,j) + d_t(j,k) \geq d_t(i,k)$, a contradiction to the fact that $k \notin \mathcal{B}_U(i,r,t)$. Therefore adding the vertex k to the ball centered at i is only going to increase its volume as $r - d_t(i,j) \leq d_t(j,k)$ (see Definition 17). Thus $\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)$ is differentiable with respect to r in the interval $(0,\Delta]$ except at finitely many points which correspond to a new vertex from U being added to the ball. Let the set of non-differentiable points be X. Then we claim that there must be an $r \in (0,\Delta]\setminus X$ such that $\phi\left(\mathcal{B}_U(i,r,t)\right) \leq \frac{1}{\Delta}\log\left(\frac{\mathrm{vol}(\mathcal{B}_U(i,\Delta,t))}{\mathrm{vol}(\mathcal{B}_U(i,0,t))}\right)$. Let us assume for the sake of a contradiction that for every $r \in (0,\Delta]\setminus X$ we have $\phi\left(\mathcal{B}_U(i,r,t)\right) > \frac{1}{\Delta}\log\left(\frac{\mathrm{vol}(\mathcal{B}_U(i,\Delta,t))}{\mathrm{vol}(\mathcal{B}_U(i,0,t))}\right)$. However integrating both sides from 0 to Δ results in a contradiction:

$$\begin{aligned}
\int_{r=0}^{\Delta} \phi\left(\mathcal{B}_U(i,r,t)\right) dr &= \int_{r=0}^{\Delta} \frac{\partial \mathcal{B}_U(i,r,t)}{\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)} dr \\
&= \int_{r=0}^{\Delta} \frac{d\left(\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)\right)}{\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)} \\
&= \log \mathrm{vol}\left(\mathcal{B}_U(i,\Delta,t)\right) - \log \mathrm{vol}\left(\mathcal{B}_U(i,0,t)\right) \\
&= \int_{r=0}^{\Delta} \frac{1}{\Delta}\log\left(\frac{\mathrm{vol}\left(\mathcal{B}_U(i,\Delta,t)\right)}{\mathrm{vol}\left(\mathcal{B}_U(i,0,t)\right)}\right) dr.
\end{aligned}$$

For any $t \in [m_\varepsilon]$ the set $\mathcal{C}_t$ is a disjoint partition of V with balls of the form $\mathcal{B}_U(i,r,t')$ for some $t' \geq t$ and $U \subseteq U_l \in \mathcal{C}_{t'+1}$: this is easily seen by induction since $\mathcal{C}_{m_\varepsilon+1}$ is initialized as V. Further, a cluster V_i is added to $\mathcal{C}_t$ either in line 15 in which case it is a ball of the form $\mathcal{B}_U(i,r,t)$ for some $U \in \mathcal{C}_{t+1}$, $i \in U$, and $r \in \mathbb{R}$ or it is added in line 9 in which case it must have been a ball $\mathcal{B}_U(i',r',t')$ for some $t' > t$, $U \subseteq U_l \in \mathcal{C}_{t'+1}$, $i' \in V$, and $r' \in \mathbb{R}$. Note that for any $t' \geq t$ and $U \subseteq V$, it holds $\gamma^U_{t'} \leq \gamma^U_t$ since for every pair $i,j \in V$ we have $\kappa(i,j) \geq 0$ and $d_t(i,j) \geq d_{t'}(i,j)$ because of constraint (13). Moreover, for any subset $U \subseteq V$ we have $\gamma^U_t \leq \gamma_t$ since $\kappa, d_t \geq 0$.

We claim that for any $t \in [m_\varepsilon]$ the total volume of the balls in $\mathcal{C}_t$ is at most $\left(2 + \frac{1}{\log n}\right)\gamma_t$. First note that the affine term $\frac{\gamma_{t'}^U}{n \log n}$ in the volume of a ball $\mathcal{B}_U(i, r, t') \in \mathcal{C}_t$ is upper bounded by $\frac{\gamma_t}{n \log n}$ and appears at most n times. Next we claim that the contribution to the total volume from the term involving the edges inside and crossing a ball $\mathcal{B}_U(i, r, t') \in \mathcal{C}_t$ is at most $2\gamma_t$. This is because the balls are disjoint, $r - d_{t'}(i,k) \leq d_{t'}(j,k) \leq d_t(j,k)$ for the crossing edges of a ball $\mathcal{B}_U(i, r, t') \in \mathcal{C}_t$ and a crossing edge contributes to the volume of at most 2 balls in $\mathcal{C}_t$. Finally, using the stopping condition of line 14 and the fact that for any $U \subseteq V$, $i \in U$ and $r \in \mathbb{R}$, we have $\operatorname{vol}(\mathcal{B}_U(i, r, t)) \in \left[\frac{\gamma_t^U}{n \log n}, \left(1 + \frac{1}{n \log n}\right)\gamma_t^U\right]$ it follows that

$$
\begin{aligned}
\sum_{\{i,j\} \in E(K_n)} \kappa(i,j) x_{ij}^t &= \sum_{\substack{\{i,j\} \in E(K_n) \\ i,j \text{ separated in } \mathcal{C}_t}} \kappa(i,j) \\
&= \frac{1}{2} \underbrace{\sum_{\substack{\mathcal{B}_U(i,r,t') \in \mathcal{C}_t: \\ t' \geq t \\ U \subseteq U_l \in \mathcal{C}_{t'+1}}} \sum_{\substack{j \in \mathcal{B}_U(i,r,t') \\ k \notin \mathcal{B}_U(i,r,t')}} \kappa(j,k)}_{\text{Since } \kappa \text{ is symmetric}} \\
&= \frac{1}{2} \sum_{\substack{\mathcal{B}_U(i,r,t') \in \mathcal{C}_t: \\ t' \geq t \\ U \subseteq U_l \in \mathcal{C}_{t'+1}}} \partial \mathcal{B}_U(i, r, t') \\
&= \frac{1}{2} \sum_{\substack{\mathcal{B}_U(i,r,t') \in \mathcal{C}_t \\ t' \geq t \\ U \subseteq U_l \in \mathcal{C}_{t'+1}}} \phi\left(\mathcal{B}_U(i, r, t')\right) \operatorname{vol}\left(\mathcal{B}_U(i, r, t')\right) \\
&\leq \sum_{\substack{\mathcal{B}_U(i,r,t') \in \mathcal{C}_t \\ t' \geq t \\ U \subseteq U_l \in \mathcal{C}_{t'+1}}} \frac{1}{2\Delta} \log\left(\frac{\operatorname{vol}\left(\mathcal{B}_U(i, \Delta, t')\right)}{\operatorname{vol}\left(\mathcal{B}_U(i, 0, t')\right)}\right) \operatorname{vol}\left(\mathcal{B}_U(i, r, t')\right) \\
&\leq \frac{1}{2\Delta} \underbrace{(\log(n \log n + 1))}_{\text{via interval bounds}} \sum_{\substack{\mathcal{B}_U(i,r,t') \in \mathcal{C}_t: \\ t' \geq t \\ U \subseteq U_l \in \mathcal{C}_{t'+1}}} \operatorname{vol}\left(\mathcal{B}_U(i, r, t')\right) \\
&\leq \frac{1+\varepsilon}{2\varepsilon} (\log(n \log n + 1)) \underbrace{\left(2 + \frac{1}{\log n}\right)\gamma_t}_{\substack{\text{contribution of affine term} \leq \frac{\gamma_t}{\log n} \\ \text{contribution of edge terms} \leq 2\gamma_t}} \\
&\leq c(\varepsilon)(\log n)\gamma_t,
\end{aligned}
$$

for some constant $c(\varepsilon) > 0$ depending only on ε.

For the run time of Algorithm 2 note that the loop in line 5 runs for at most $n - 1$ steps, while the loop in line 7 runs for at most n steps. To compute the ball $\mathcal{B}_U(i, r, t)$ of least radius r such that $\phi\left(\mathcal{B}_U(i, r, t)\right) \leq \frac{1}{\Delta} \log\left(\frac{\operatorname{vol}(\mathcal{B}_U(i,\Delta,t))}{\operatorname{vol}(\mathcal{B}_U(i,0,t))}\right)$ for any $U \subseteq V$, sort the vertices in $U \setminus \{i\}$ in increasing order of distance from i according to d_t. Let the vertices in $U \setminus \{i\}$ in this sorted order be $\{j_1, \ldots, j_{|U|-1}\}$. Then it suffices to check the expansion of the balls $\{i\}$ and $\{i\} \cup \{j_1, \ldots, j_k\}$ for every $k \in [|U| - 1]$. It is straightforward to see that all the other steps in Algorithm 2 run in time polynomial in n. □

We are now ready to prove the main theorem showing that we can obtain a low cost non-trivial ultrametric from Algorithm 2.

Theorem 21. *Let* $\{x_{ij}^t \mid t \in [m_\varepsilon], i,j \in V\}$ *be the output of Algorithm 2 on an optimal solution* $\{d_t\}_{t\in[n-1]}$ *of LP-ultrametric for any choice of* $\varepsilon \in (0,1)$. *Define the sequence* $\left\{y_{ij}^t\right\}$ *for every* $t \in [n-1]$ *and* $i,j \in V$ *as*

$$y_{ij}^t := \begin{cases} x_{ij}^{\lfloor t/(1+\varepsilon)\rfloor} & \text{if } t > 1+\varepsilon \\ 1 & \text{if } t \le 1+\varepsilon. \end{cases}$$

Then y_{ij}^t *is feasible for ILP-ultrametric and satsfies*

$$\sum_{t=1}^{n-1} \sum_{\{i,j\}\in E(K_n)} \kappa(i,j) y_{ij}^t \le (2c(\varepsilon)\log n)\,\mathrm{OPT}$$

where OPT *is the optimal solution to ILP-ultrametric and* $c(\varepsilon)$ *is the constant in the statement of Theorem 21.*

Proof. Note that by Theorem 20 for every $t \in [m_\varepsilon]$, x_{ij}^t is feasible for the layer-$\lfloor(1+\varepsilon)t\rfloor$ problem ILP-layer and that there is a constant $c(\varepsilon) > 0$ such that for every $t \in [m_\varepsilon]$, $\sum_{\{i,j\}\in E(K_n)} \kappa(i,j)x_{ij}^t \le (c(\varepsilon)\log n)\,\gamma_t$. Let y_{ij}^t be as in the statement of the theorem. The graph $G_t = (V, E_t)$ as in Definition 10 corresponding to y_{ij}^t for $t \le 1+\varepsilon$ consists of isolated vertices i.e., cliques of size 1. By definition y_{ij}^t is feasible for the layer-t problem ILP-layer. The collection $\mathcal{C}_1$ corresponding to x_{ij}^1 consists of cliques of size at most $1+\varepsilon$, however since $0 < \varepsilon < 1$ it follows that the cliques in $\mathcal{C}_1$ are isolated vertices and so $x_{ij}^1 = 1$ for every $\{i,j\} \in E(K_n)$. Thus $\sum_{i,j} \kappa(i,j) y_{ij}^t = \sum_{i,j} \kappa(i,j) x_{ij}^1 \le (c(\varepsilon)\log n)\,\gamma_1$ for $t \le 1+\varepsilon$ by Theorem 20. Moreover for every $t > 1+\varepsilon$, we have $\sum_{i,j} \kappa(i,j) y_{ij}^t \le (c(\varepsilon)\log n)\gamma_{\lfloor t/(1+\varepsilon)\rfloor}$ again by Theorem 20. We claim that y_{ij}^t is feasible for ILP-ultrametric. The solution y_{ij}^t corresponds to the collection $\mathcal{C}_{\lfloor \frac{t}{1+\varepsilon}\rfloor}$ for $t > 1+\varepsilon$ or to the collection $\mathcal{C}_1$ for $t \le 1+\varepsilon$ from Algorithm 2. For any $t < m_\varepsilon$, every ball $\mathcal{B}_U(i,r,t) \in \mathcal{C}_t$ comes from the refinement of a ball $\mathcal{B}_{U'}(i',r',t')$ for some $i' \in V$, $r \in \mathbb{R}$, $t' \ge t$ and $U' \supseteq U$. Thus y_{ij}^t satisfies Condition Nested cliques of Lemma 12. On the other hand line 8 ensures that if $|\mathcal{B}_U(i,r,t)| = \lfloor(1+\varepsilon)s\rfloor$ for some $U \subseteq V$ and $s < t$ then $\mathcal{B}_U(i,r,t)$ also appears as a ball in $\mathcal{C}_s$. Therefore y_{ij}^t also satisfies Condition Realization of Lemma 12 and so is feasible for ILP-ultrametric. The cost of y_{ij}^t is at most

$$\begin{aligned} \sum_{t=1}^{n-1} \sum_{\{i,j\}\in E(K_n)} \kappa(i,j) y_{ij}^t &\le (c(\varepsilon)\log n)\left(\gamma_1 + \sum_{t=2}^{n-1} \gamma_{\lfloor t/(1+\varepsilon)\rfloor}\right) \\ &\le 2c(\varepsilon)\log n \sum_{t=1}^{n-1} \gamma_t \\ &\le 2c(\varepsilon)\log n\,\mathrm{OPT}, \end{aligned}$$

where we use the fact that $\sum_{t=1}^{n-1} \gamma_t = \mathrm{OPT}(LP) \le \mathrm{OPT}$ since LP-ultrametric is a relaxation of ILP-ultrametric. □

Theorem 21 implies the following corollary where we put everything together to obtain a hierarchical clustering of V in time polynomial in n with $|V| = n$. Let $\mathcal{T}$ denote the set of all possible hierarchical clusterings of V.

Corollary 22. *Given a data set* V *of* n *points and a similarity function* $\kappa : V \times V \to \mathbb{R}_{\ge 0}$, *Algorithm 3 returns a hierarchical clustering* T *of* V *satisfying*

$$\mathrm{cost}(T) \le O(\log n) \min_{T' \in \mathcal{T}} \mathrm{cost}(T').$$

Moreover Algorithm 3 runs in time polynomial in n *and* $\log\left(\max_{i,j\in V} \kappa(i,j)\right)$.

Input: Data set V of n points, similarity function $\kappa : V \times V \to \mathbb{R}_{\geq 0}$
Output: Hierarchical clustering of V

Solve LP-ultrametric to obtain optimal sequence of spreading metrics $\{d_t \mid d_t : V \times V \to [0,1]\}$
Fix a choice of $\varepsilon \in (0,1)$
$m_\varepsilon \leftarrow \left\lfloor \frac{n-1}{1+\varepsilon} \right\rfloor$
Let $\left\{ x_{ij}^t \mid t \in [m_\varepsilon] \right\}$ be the output of Algorithm 2 on $V, \kappa, \{d_t\}_{t \in [n-1]}$
Let $y_{ij}^t := \begin{cases} x_{ij}^{\lfloor t/(1+\varepsilon) \rfloor} & \text{if } t > 1+\varepsilon \\ 1 & \text{if } t \leq 1+\varepsilon \end{cases}$ for every $t \in [n-1], i, j \in E(K_n)$
$d(i,j) \leftarrow \sum_{t=1}^{n-1} y_{ij}^t$ for every $i, j \in E(K_n)$
$d(i,i) \leftarrow 0$ for every $i \in V$
Let r, T be the output of Algorithm 1 on V, d
return r, T

Algorithm 3: Hierarchical clustering of V for cost function (1)

Proof. Note that the claim about $\text{cost}(T)$ follows from Corollary 9 and Theorem 21. We can find an optimal solution to LP-ultrametric due to Lemma 16 using the Ellipsoid algorithm in time polynomial in n and $\log\left(\max_{i,j \in V} \kappa(i,j)\right)$. Algorithm 2 runs in time polynomial in n due to Theorem 20. Finally, Algorithm 1 runs in time $O\left(n^2\right)$ due to Lemma 8. □

Example 23. *Recall the toy data set from Example 14. Applying Algorithm 3 with any $\varepsilon \leq 0.5$ recovers the optimal clustering as shown in Figure 1 of cost -33.16 according to cost function (1). For $\varepsilon > 0.5$ we get the clustering in Figure 2 of value -21.68 which is slightly less than the optimal value according to cost function (1). Interestingly the optimal solution to LP-ultrametric is* not *integral in this case.*

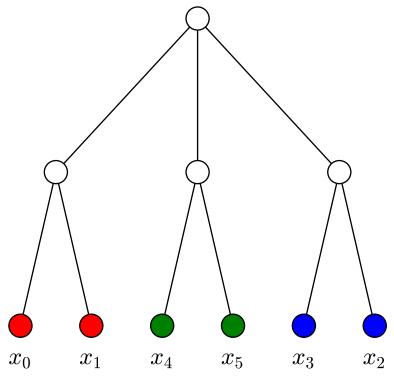

Figure 2: Hierarchical clustering produced by Algorithm 3 on Example 14 for $\varepsilon > 0.5$

Generalized Cost Function

In this section we study the following natural generalization of cost function (1) also introduced by [Dasgupta, 2016] where the distance between the two points is scaled by a function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ i.e.,

$$\text{cost}_f(T) := \sum_{\{i,j\} \in E(K_n)} \kappa(i,j) f\left(|\text{leaves } T[\text{lca}(i,j)]|\right). \tag{18}$$

In order that cost function (18) makes sense, f should be strictly increasing and satisfy $f(0) = 0$. Possible choices for f could be $\{x^2, e^x - 1, \log(1+x)\}$. The top down heuristic in [Dasgupta, 2016] finds the optimal hierarchical clustering up to an approximation factor of $c_n \log n$ with c_n being defined as

$$c_n := 3\alpha_n \max_{1 \leq n' \leq n} \frac{f(n')}{f\left(\lceil n'/3 \rceil\right)}$$

and where α_n is the approximation factor from the Sparsest Cut algorithm used. Note that this approximation can be arbitrarily bad: for example if f is of the form $f(x) = x^k$ for some $k = \Omega(\log n)$ or $f(x) = e^x - 1$ then the approximation guarantee is $\Omega(n)$ and $e^{\Omega(n)}$ respectively.
A naive approach to solving this problem using the ideas of Algorithm 2 would be to replace the objective function of ILP-ultrametric by

$$\sum_{\{i,j\} \in E(K_n)} \kappa(i,j) f\left(\sum_{t=1}^{n-1} x_{ij}^t\right).$$

This makes the corresponding analogue of LP-ultrametric non-linear however, and for a general κ and f it is not clear how to compute an optimum solution in polynomial time. One possible solution is to assume that f is convex and use the Frank-Wolfe algorithm to compute an optimum solution. That still leaves the problem of how to relate $f\left(\sum_{t=1}^{n-1} x_{ij}^t\right)$ to $\sum_{t=1}^{n-1} f\left(x_{ij}^t\right)$ as one would have to do to get a corresponding version of Theorem 21. The following simple observation points to an alternate way of tackling this problem.

Observation 24. *Let $d : V \times V \to \mathbb{R}$ be an ultrametric and $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a strtictly increasing function such that $f(0) = 0$. Define the function $f(d) : V \times V \to \mathbb{R}$ as $f(d)(i,j) := f(d(i,j))$. Then $f(d)$ is also an ultrametric on V.*

Therefore by Corollary 9 to find a minimum cost hierarchical clustering T of V according to the cost function (18), it suffices to minimize $\langle \kappa, d \rangle$ where d is the f-image of a non-trivial ultrametric as in Definition 3. The following lemma lays down the analogue of Conditions 1 and 2 from Definition 3 that the f-image of a non-trivial ultrametric satisfies.

Lemma 25. *Let $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a strictly increasing function satisfying $f(0) = 0$. An ultrametric d on V is the f-image of a non-trivial ultrametric on V iff*

1. *for every non-empty set $S \subseteq V$, there is a pair of points $i, j \in S$ such that $d(i,j) \geq f(|S| - 1)$,*
2. *for any t if S_t is the equivalence class of V under the relation $i \sim j$ iff $d(i,j) \leq t$, then $\max_{i,j \in S_t} d(i,j) \leq f(|S_t| - 1)$.*

Proof. If d is the f-image of a non-trivial ultrametric d' on V then clearly d satisfies Conditions 1 and 2. Conversely, let d be an ultrametric on V satisfying Conditions 1 and 2. Note that f is strictly increasing and V is a finite set and thus f^{-1} exists and is strictly increasing as well, with $f^{-1}(0) = 0$. Define d' as $d'(i,j) := f^{-1}(d(i,j))$ for every $i, j \in V$. By Observation 24 d' is an ultrametric on d satisfying Conditions 1 and 2 of Definition 3 and so d' is a non-trivial ultrametric on V. □

Lemma 25 allows us to write the analogue of ILP-ultrametric for finding the minimum cost ultrametric that is the f-image of a non-trivial ultrametric on V. Note that by Lemma 5 the range of such an ultrametric is the set $\{f(0), f(1), \ldots, f(n-1)\}$. We have the binary variables x_{ij}^t for every distinct pair $i, j \in V$ and $t \in [n-1]$, where $x_{ij}^t = 1$ if $d(i,j) \geq f(t)$ and 0 if $d(i,j) < f(t)$.

$$\min \quad \sum_{t=1}^{n-1} \sum_{\{i,j\} \in E(K_n)} \kappa(i,j)\,(f(t) - f(t-1))\, x_{ij}^t \qquad \text{(f-ILP-ultrametric)}$$

$$\text{s.t.} \quad x_{ij}^t \geq x_{ij}^{t+1} \qquad \forall i,j \in V, t \in [n-2] \tag{19}$$

$$x_{ij}^t + x_{jk}^t \geq x_{ik}^t \qquad \forall i,j,k \in V, t \in [n-1] \tag{20}$$

$$\sum_{i,j \in S} x_{ij}^t \geq 2 \qquad \forall t \in [n-1], S \subseteq V, |S| = t+1 \tag{21}$$

$$\sum_{i,j \in S} x_{ij}^{|S|} \leq |S| \left(\sum_{i,j \in S} x_{ij}^t + \sum_{\substack{i \in S \\ j \notin S}} \left(1 - x_{ij}^t\right) \right) \forall t \in [n-1], S \subseteq V \tag{22}$$

$$x_{ij}^t = x_{ji}^t \qquad \forall i,j \in V, t \in [n-1] \tag{23}$$

$$x_{ij}^t \in \{0,1\} \qquad \forall i,j \in V, t \in [n-1] \tag{24}$$

If x_{ij}^t is a feasible solution to f-ILP-ultrametric then the ultrametric represented by it is defined as

$$d(i,j) := \sum_{t=1}^{n-1} (f(t) - f(t-1)) x_{ij}^t.$$

Constraint (21) ensures that d satisfies Condition 1 of Lemma 25, since for every $S \subseteq V$ of size $t+1$ we have a pair $i, j \in S$ such that $d(i,j) \geq f(t)$. Similarly constraint (22) ensures that d satisfies Condition 2 of Lemma 25 since it is active if and only if S is an equivalence class of V under the relation $i \sim j$ iff $d(i,j) < f(t)$. In this case Condition 2 requires $\max_{i,j \in S} d(i,j) \leq f(|S|-1)$ or in other words $x_{ij}^{|S|} = 0$ for every $i, j \in S$.

Similar to ILP-layer we define an analogous *layer-t problem* where we fix a choice of $t \in [n-1]$ and drop the constraints that relate the different layers to each other.

$$\min \quad \sum_{\{i,j\} \in E(K_n)} \kappa(i,j)\,(f(t) - f(t-1))\, x_{ij}^t \qquad \text{(f-ILP-layer)}$$

$$\text{s.t.} \quad x_{ij}^t + x_{jk}^t \geq x_{ik}^t \qquad \forall i,j,k \in V \tag{25}$$

$$\sum_{i,j \in S} x_{ij}^t \geq 2 \qquad \forall S \subseteq V, |S| = t+1 \tag{26}$$

$$x_{ij}^t = x_{ji}^t \qquad \forall i,j \in V \tag{27}$$

$$x_{ij}^t \in \{0,1\} \qquad \forall i,j \in V \tag{28}$$

Note that f-ILP-ultrametric and f-ILP-layer differ from ILP-ultrametric and ILP-layer respectively only in the objective function. Therefore Lemmas 11 and 12 also give a combinatorial characterization of the set of feasible solutions to f-ILP-layer and f-ILP-ultrametric respectively.

Example 26. *Let us revisit the toy data set from Examples 14 and 23. As before f-ILP-ultrametric gives us a way to recover the optimal hierarchical clustering of this data set for various choices of f. Choices of f that are sublinear, e.g., $f(x) = \log(1+x)$ return Figure 1 as the optimal hierarchical clustering. On the other hand choosing $f(x) \in \{x^2, x^3, e^x - 1\}$ leads to Figure 3 as the optimal hierarchical clustering.*

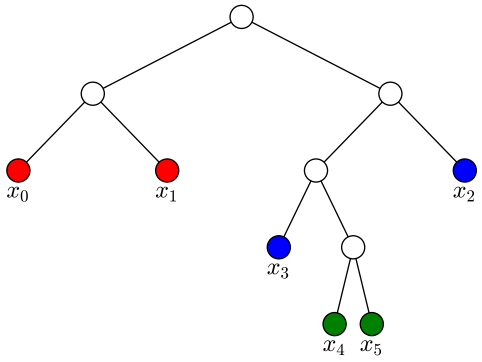

Figure 3: Optimal hierarchical clustering using cosine similarity with $f(x) \in \{x^2, x^3, e^x - 1\}$ on Example 14

Similarly by Lemma 13 we may replace constraint 21 by the following equivalent constraint over all subsets of V

$$\sum_{j \in S} x_{ij}^t \geq |S| - t \qquad \forall t \in [n-1], S \subseteq V, i \in S.$$

This gives us the analogue of LP-ultrametric in which we drop constraint 22 and enforce it in the rounding procedure.

$$\begin{aligned}
\min \quad & \sum_{t=1}^{n-1} \sum_{\{i,j\} \in E(K_n)} \kappa(i,j) \left(f(t) - f(t-1)\right) x_{ij}^t && \text{(f-LP-ultrametric)} \\
\text{s.t.} \quad & x_{ij}^t \geq x_{ij}^{t+1} \qquad \forall i,j \in V, t \in [n-2] && (29) \\
& x_{ij}^t + x_{jk}^t \geq x_{ik}^t \qquad \forall i,j,k \in V, t \in [n-1] && (30) \\
& \sum_{j \in S} x_{ij}^t \geq |S| - t \qquad \forall t \in [n-1], S \subseteq V, i \in S && (31) \\
& x_{ij}^t = x_{ji}^t \qquad \forall i,j \in V, t \in [n-1] && (32) \\
& 0 \leq x_{ij}^t \leq 1 \qquad \forall i,j \in V, t \in [n-1] && (33)
\end{aligned}$$

Since f-LP-ultrametric differs from LP-ultrametric only in the objective function, it follows from Lemma 16 that an optimum solution to f-LP-ultrametric can be computed in time polynomial in n. As before, a feasible solution x_{ij}^t of f-LP-ultrametric induces a sequence $\{d_t\}_{t \in [n-1]}$ of spreading metrics on V defined as

$d_t(i,j) := x^t_{ij}$. Note that in contrast to the ultrametric d, the spreading metrics $\{d_t\}_{t\in[n-1]}$ are independent of the function f.
Let $\mathcal{B}_U(i,r,t)$ be a ball of radius r centered at $i \in U$ for some set $U \subseteq V$ as in Definition 17. For a subset $U \subseteq V$, let γ^U_t be defined as before to be the value of the layer-t objective corresponding to a solution d_t of f-LP-ultrametric restricted to U i.e.,

$$\gamma^U_t := \sum_{\substack{i,j\in U\\ i<j}} (f(t)-f(t-1))\,\kappa(i,j)d_t(i,j).$$

As before, we denote γ^V_t by γ_t. We will associate a volume $\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)$ and a boundary $\partial\mathcal{B}_U(i,r,t)$ to the ball $\mathcal{B}_U(i,r,t)$ as in Section 4.

Definition 27. *Let U be an arbitrary subset of V. For a vertex $i \in U$, radius $r \in \mathbb{R}$, and $t \in [n-1]$, let $\mathcal{B}_U(i,r,t)$ be the ball of radius r as in Definition 17. Then we define its* volume *as*

$$\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right) := \frac{\gamma^U_t}{n\log n} + (f(t)-f(t-1))\left(\sum_{\substack{j,k\in\mathcal{B}_U(i,r,t)\\ j<k}} \kappa(j,k)d_t(j,k) + \sum_{\substack{j\in\mathcal{B}_U(i,r,t)\\ k\notin\mathcal{B}_U(i,r,t)\\ k\in U}} \kappa(j,k)\left(r-d_t(i,j)\right)\right).$$

The boundary *of the ball $\partial\mathcal{B}_U(i,r,t)$ is the partial derivative of volume with respect to the radius:*

$$\partial\mathcal{B}_U(i,r,t) := (f(t)-f(t-1))\left(\frac{\partial\,\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)}{\partial r}\right) = (f(t)-f(t-1))\left(\sum_{\substack{j\in\mathcal{B}_U(i,r,t)\\ k\notin\mathcal{B}_U(i,r,t)\\ k\in U}} \kappa(j,k)\right).$$

The expansion *$\phi\left(\mathcal{B}_U(i,r,t)\right)$ of the ball $\mathcal{B}_U(i,r,t)$ is defined as the ratio of its boundary to its volume, i.e.,*

$$\phi\left(\mathcal{B}_U(i,r,t)\right) := \frac{\partial\mathcal{B}_U(i,r,t)}{\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)}.$$

Note that the expansion $\phi\left(\mathcal{B}_U(i,r,t)\right)$ of Definition 27 is the same as as in Definition 19 since the $(f(t)-f(t-1))$ term cancels out. Thus one could run Algorithm 2 with the same notion of volume as in Definition 19, however in that case the analogous versions of Theorems 20 and 21 do not follow as naturally. Let $m_\varepsilon := \left\lfloor \frac{n-1}{1+\varepsilon} \right\rfloor$ as in Algorithm 2. The following is then a simple corollary of Theorem 20.

Corollary 28. *Let $\left\{x^t_{ij} \mid t \in [n-1], i,j \in V\right\}$ be the output of Algorithm 2 using the notion of volume, boundary and expansion as in Definition 27, on a feasible solution to f-LP-ultrametric and any choice of $\varepsilon \in (0,1)$. For any $t \in [m_\varepsilon]$, x^t_{ij} is feasible for the layer-$\lfloor(1+\varepsilon)t\rfloor$ problem f-ILP-layer and there is a constant $c(\varepsilon) > 0$ depending only on ε such that*

$$\sum_{\{i,j\}\in E(K_n)} \kappa(i,j)\left(f(t)-f(t-1)\right)x^t_{ij} \le \left(c(\varepsilon)\log n\right)\gamma_t.$$

Corollary 28 allows us to prove the analogue of Theorem 21, i.e., we can use Algorithm 2 to get an ultrametric that is an f-image of a non-trivial ultrametric and whose cost is at most $O(\log n)$ times the cost of an optimal hierarchical clustering according to cost function 18.

Theorem 29. *Let* $\{x_{ij}^t \mid t \in [m_\varepsilon], i,j \in V\}$ *be the output of Algorithm 2 using the notion of volume, boundary and expansion as in Definition 27, on an optimal solution* $\{d_t\}_{t\in[n-1]}$ *of f-LP-ultrametric for any choice of* $\varepsilon \in (0,1)$*. Define the sequence* $\left\{y_{ij}^t\right\}$ *for every* $t \in [n-1]$ *and* $i,j \in V$ *as*

$$y_{ij}^t := \begin{cases} x_{ij}^{\lfloor t/(1+\varepsilon)\rfloor} & \text{if } t > 1+\varepsilon \\ 1 & \text{if } t \leq 1+\varepsilon. \end{cases}$$

Then y_{ij}^t *is feasible for f-ILP-ultrametric and there is a constant* $c(\varepsilon) > 0$ *such that*

$$\sum_{t=1}^{n-1} \sum_{\{i,j\}\in E(K_n)} \kappa(i,j)\left(f(t) - f(t-1)\right) y_{ij}^t \leq \left(c(\varepsilon)\log n\right) \mathrm{OPT}$$

where OPT *is the optimal solution to f-ILP-ultrametric.*

Proof. Immediate from Corollary 28 and Theorem 21. □

Finally we put everything together to obtain the corresponding Algorithm 4 that outputs a hierarchical clustering of V of cost at most $O(\log n)$ times the optimal clustering according to cost function (18).

Corollary 30. *Given a data set* V *of* n *points and a similarity function* $\kappa : V \times V \to \mathbb{R}_{\geq 0}$*, Algorithm 4 returns a hierarchical clustering* T *of* V *satisfying*

$$\mathrm{cost}_f(T) \leq O(\log n) \min_{T' \in \mathcal{T}} \mathrm{cost}_f(T').$$

Moreover Algorithm 4 runs in time polynomial in n *and* $\log\left(\max_{i,j\in V} \kappa(i,j)\right)$*.*

Proof. Note that the claim about $\mathrm{cost}_f(T)$ follows from Corollary 9, Lemma 25 and Theorem 29. We can find an optimal solution to f-LP-ultrametric due to Lemma 16 using the Ellipsoid algorithm in time polynomial in n and $\log\left(\max_{i,j\in V} \kappa(i,j)\right)$. Algorithm 2 runs in time polynomial in n due to Theorem 20. Finally, Algorithm 1 runs in time $O\left(n^2\right)$ due to Lemma 8. □

Input: Data set V of n points, similarity function $\kappa : V \times V \to \mathbb{R}_{\geq 0}$, $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ strictly increasing with $f(0) = 0$
Output: Hierarchical clustering of V

1 Solve f-LP-ultrametric to obtain optimal sequence of spreading metrics $\{d_t \mid d_t : V \times V \to [0,1]\}$
2 Fix a choice of $\varepsilon \in (0,1)$
3 $m_\varepsilon \leftarrow \left\lfloor \frac{n-1}{1+\varepsilon} \right\rfloor$
4 Let $\left\{x_{ij}^t \mid t \in [m_\varepsilon]\right\}$ be the output of Algorithm 2 on $V, \kappa, \{d_t\}_{t\in[n-1]}$
5 Let $y_{ij}^t := \begin{cases} x_{ij}^{\lfloor t/(1+\varepsilon)\rfloor} & \text{if } t > 1+\varepsilon \\ 1 & \text{if } t \leq 1+\varepsilon \end{cases}$ for every $t \in [n-1], i,j \in E(K_n)$
6 $d(i,j) \leftarrow \sum_{t=1}^{n-1} \left(f(t) - f(t-1)\right) y_{ij}^t$ for every $i,j \in E(K_n)$
7 $d(i,i) \leftarrow 0$ for every $i \in V$
8 Let r, T be the output of Algorithm 1 on $V, f^{-1}(d)$
9 **return** r, T

Algorithm 4: Hierarchical clustering of V for cost function (18)

Note that the approximation factor in this case is $O(\log n)$ irrespective of the choice of f as long as f is strictly increasing and $f(0) = 0$.

Example 31. *As a final example let us once again recall the toy data set from Examples 14, 23 and 26. We use Algorithm 4 for $f \in \{\log(1+x), x^2, x^3, e^x - 1\}$ to find a hierarchical clustering of this data set. For small values of ε we recover the tree from Figure 1. Note that this not optimal for f-ILP-ultrametric when f is super-linear. For ε in $(0.5, 0.6)$ we get the clustering of Figure 4 while for higher values of ε we once again get the clustering of Figure 2.*

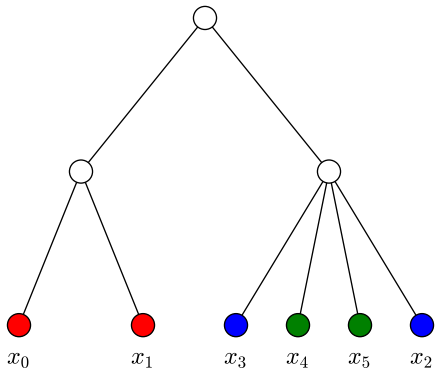

Figure 4: Hierarchical clustering output by Algorithm 4 with $f(x) \in \{\log(1+x), x^2, x^3, e^x - 1\}$ and $\varepsilon \in (0.5, 0.6)$ on Example 14

Experiments

Finally, we describe the experiments we performed. For small data sets ILP-ultrametric and f-ILP-ultrametric describe integer programming formulations that allow us to compute the exact optimal hierarchical clustering for cost functions (1) and (18) respectively. We implement f-ILP-ultrametric where one can plug in any strictly increasing function f satisfying $f(0) = 0$. In particular, setting $f(x) = x$ gives us ILP-ultrametric. We use the Mixed Integer Programming (MIP) solver Gurobi 6.5 [Gurobi Optimization, 2015] on a Python interface. Similarly, we also implement Algorithms 1, 2, and 4 using Gurobi as our LP solver. Note that Algorithm 4 needs to fix a choice of parameter $\varepsilon \in (0, 1)$. In Sections 4 and 5 we did not discuss the effect of the choice of the parameter ε in detail. In particular, we need to choose an ε small enough such that for every $U \subseteq V$ encountered in Algorithm 2, $\text{vol}\left(\mathcal{B}_U(i, \Delta, t)\right)$ is of the same sign as $\text{vol}\left(\mathcal{B}_U(i, 0, t)\right)$ for every $t \in [n-1]$, so that $\log\left(\frac{\text{vol}(\mathcal{B}_U(i,\Delta,t))}{\text{vol}(\mathcal{B}_U(i,0,t))}\right)$ is defined. In our experiments we start with a particular value of ε (say 0.5) and halve it till the volumes have the same sign. For the sake of exposition, we limit ourselves to the following choices for the function f

$$\{x, x^2, \log(1+x), e^x - 1\}.$$

By Lemma 16 we can optimize over f-LP-ultrametric in time polynomial in n using the Ellipsoid method. In practice however, we use the *dual simplex* method where we separate triangle inequality constraints (30) and spreading constraints (31) to obtain fast computations. For the similarity function $\kappa : V \times V \to \mathbb{R}$ we limit ourselves to using *cosine similarity* and the *Gaussian kernel* with $\sigma = 1$. They are defined formally below.

Definition 32 (Cosine similarity)**.** *Given a data set $V \in \mathbb{R}^m$ for some $m \geq 0$, the cosine similarity κ_{cos} is defined as $\kappa_{cos}(x,y) := \frac{\langle x,y\rangle}{\|x\|\|y\|}$.*

Since the LP rounding Algorithm 2 assumes that $\kappa \geq 0$ in practice we implement $1 + \kappa_{cos}$ rather than κ_{cos}. Note that by Lemma 1 this does not change the relative cost of hierarchical clusterings on V according to cost function (1).

Definition 33 (Gaussian kernel)**.** *Given a data set $V \in \mathbb{R}^m$ for some $m \geq 0$, the Gaussian kernel κ_{gauss} with standard deviation σ is defined as $\kappa_{gauss}(x,y) := \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$.*

The main aim of our experiments was to answer the following two questions.

1. How good is the hierarchal clustering obtained from Algorithm 4 as opposed to the true optimal output by f-ILP-ultrametric?

2. How good does Algorithm 4 perform compared to other hierarchical clustering methods?

For the first question, we are restricted to working with small data sets since computing an optimum solution to f-ILP-ultrametric is expensive. In this case we consider synthetic data sets of small size and samples of some data sets from the UCI database [Lichman, 2013]. The synthetic data sets we consider are mixtures of Gaussians in various small dimensional spaces. Figure 5 shows a comparison of the cost of the hierarchy (according to cost function (18)) returned by solving f-ILP-ultrametric and by Algorithm 4 for various forms of f when the similarity function is κ_{cos} and κ_{gauss}. Note that we normalize the cost of the tree returned by f-ILP-ultrametric and Algorithm 4 by the cost of the trivial clustering r, T^* where T^* is the star graph with V as its leaves and r as the internal node. In other words $d_{T^*}(i,j) = n-1$ and so the normalized cost of any tree lies in the interval $(0,1]$.
For the second question some of the popular algorithms for hierarchical clustering are *single linkage*, *average linkage*, *complete linkage*, and *Ward's method* [Ward Jr, 1963]. To get a numerical handle on how good a hierarchical clustering T of V is, we prune the tree to get the *best* k flat clusters and measure its error relative to the target clustering. We use the following notion of error also known as *Classification Error* that is standard in the literature for hierarchical clustering (see, e.g., [Meilă and Heckerman, 2001]). Note that we may think of a flat k-clustering of the data V as a function h mapping elements of V to a label set $\mathcal{L} := \{1,\ldots,k\}$. Let S_k denote the group of permutations on k letters.

Definition 34 (Classification Error)**.** *Given a proposed clustering $h : V \to \mathcal{L}$ its* classification error *relative to a target clustering $g : V \to \mathcal{L}$ is denoted by* $\operatorname{err}(g,h)$ *and is defined as*

$$\operatorname{err}(g,h) := \min_{\sigma \in S_k} \left[\Pr_{x \in V} [h(x) \neq \sigma(g(x))] \right].$$

Example 35. *Recall the data set from Example 14. Let $k = 3$ and g be the target clustering defined as $g(x_0) = g(x_1) = 1$, $g(x_2) = g(x_3) = 2$, and $g(x_4) = g(x_5) = 3$. Then the error of the best pruning of the hierarchal clustering in Figures 1 and 2 is 0 while for Figures 3 and 4 it is $\frac{1}{6}$.*

We compare the error of Algorithm 4 with the various linkage based algorithms that are commonly used for hierarchical clustering, as well as Ward's method and the k-means algorithm. We test Algorithm 4 most extensively for $f(x) = x$ while doing a smaller number of tests for $f(x) \in \{x^2, \log(1+x), e^x - 1\}$. Note that both Ward's method and the k-means algorithm work on the squared Euclidean distance $\|x-y\|_2^2$ between two points $x, y \in V$, i.e., they both require an embedding of the data points into a normed vector space. For the linkage based algorithms we use the same notion of simlarity $1 + \kappa_{cos}$ or κ_{gauss} that we use

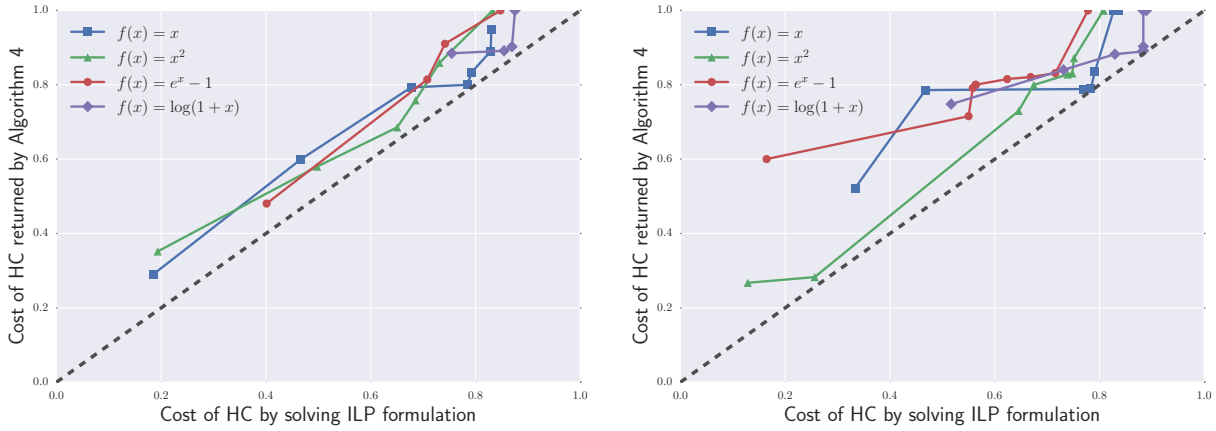

Figure 5: Comparison of f-ILP-ultrametric and Algorithm 4 for $1 + \kappa_{cos}$ (left) and κ_{gauss} (right)

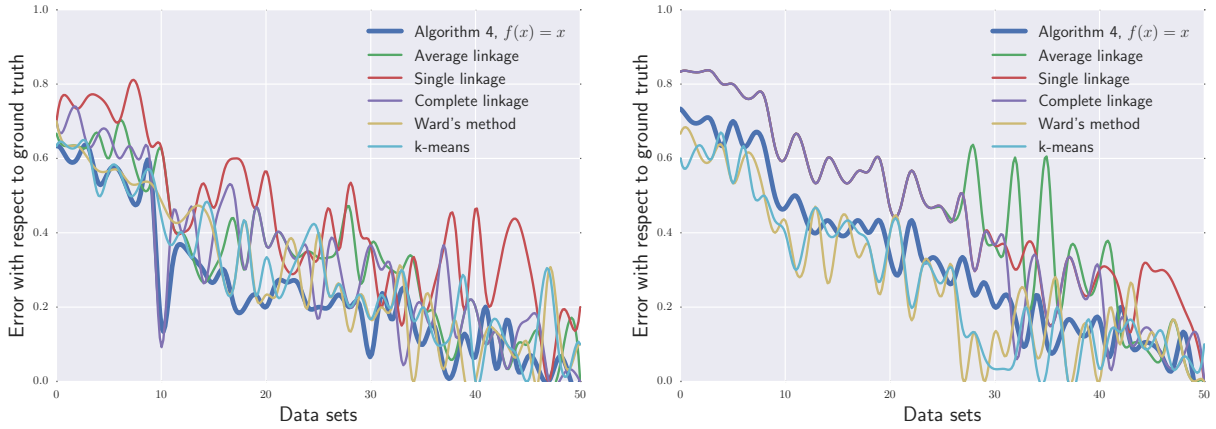

Figure 6: Comparison of Algorithm 4 using $f(x) = x$, with other algorithms for clustering using $1 + \kappa_{cos}$ (left) and κ_{gauss} (right)

for Algorithm 4. For comparison we use a mix of synthetic data sets as well as the Wine, Iris, Soybean-small, Digits, Glass, and Wdbc data sets from the UCI repository [Lichman, 2013]. For some of the larger data sets, we sample uniformly at random a smaller number of data points and take the average of the error over the different runs. Figures 6, 7, 8, and 9 show that the hierarchical clustering returned by Algorithm 4 with $f(x) \in \{x, x^2, \log(1+x), e^x - 1\}$ often has better projections into flat clusterings than the other algorithms. This is especially true when we compare it to the linkage based algorithms, since they use the same pairwise similarity function as Algorithm 4, as opposed to Ward's method and k-means.

Discussion

In this work we have studied the cost functions (1) and (18) for hierarchical clustering given a pairwise similarity function over the data and shown an $O(\log n)$ approximation algorithm for this problem. As briefly mentioned in Section 2 however, such a cost function is not unique. Further, there is an intimate connection between hierarchical clusterings and ultrametrics over discrete sets which points to other directions for formulating a cost function over hierarchies. In particular we briefly mention the related notion of *hierarchically well-separated trees* (HST) as defined in [Bartal, 1996] (see also [Bartal et al., 2001, Bartal et al., 2003]). A

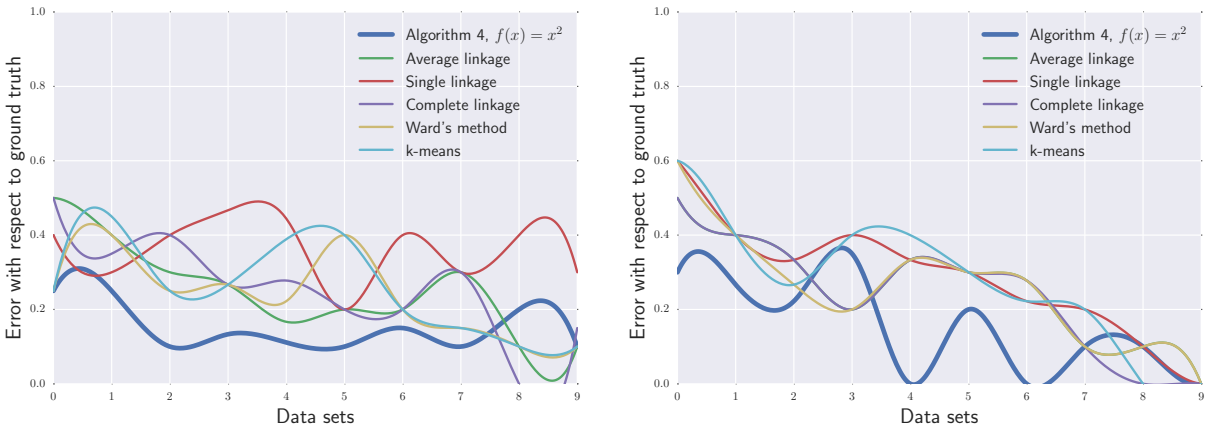

Figure 7: Comparison of Algorithm 4 using $f(x) = x^2$, with other algorithms for clustering using $1 + \kappa_{cos}$ (left) and κ_{gauss} (right)

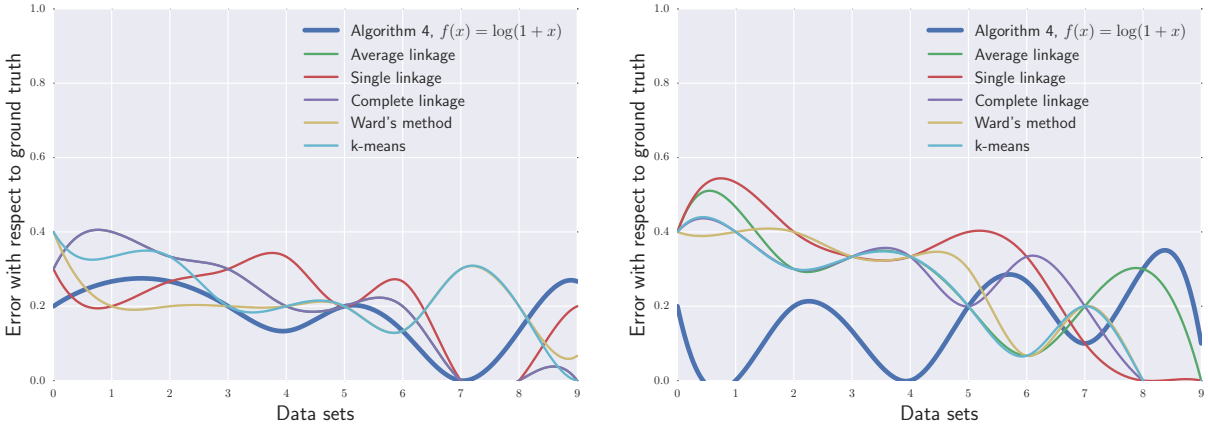

Figure 8: Comparison of Algorithm 4 using $f(x) = \log(1 + x)$, with other algorithms for clustering using $1 + \kappa_{cos}$ (left) and κ_{gauss} (right)

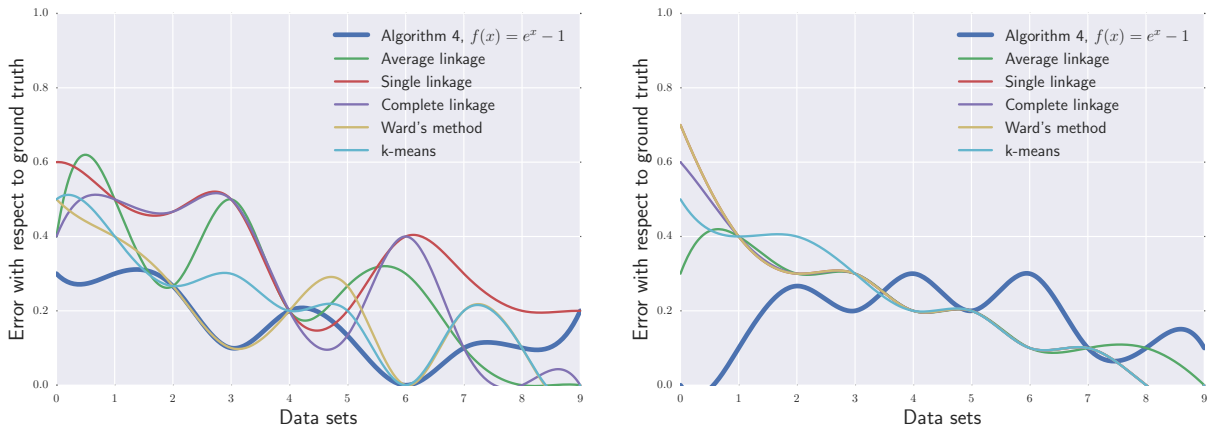

Figure 9: Comparison of Algorithm 4 using $f(x) = e^x - 1$, with other algorithms for clustering using $1 + \kappa_{cos}$ (left) and κ_{gauss} (right)

k-HST for $k \geq 1$ is a tree T such that each vertex $u \in T$ has a label $\Delta(u) \geq 0$ such that $\Delta(u) = 0$ if and only if u is a leaf of T. Further, if u is a child of v in T then $\Delta(u) \leq \Delta(v)/k$. It is well known that any ultrametric d on a finite set V is equivalent to a 1-HST where V is the set of leaves of T and $d(i,j) = \Delta(\mathrm{lca}(i,j))$ for every $i, j \in V$. Thus in the special case when $\Delta(u) = |\text{leaves } T[u]| - 1$ we get the cost function (1), while if $\Delta(u) = f(|\text{leaves } T[u]| - 1)$ for a strictly increasing function f with $f(0) = 0$ then we get cost function (18). It turns out this assumption on Δ enables us to prove the combinatorial results of Section 3 and give a $O(\log n)$ approximation algorithm to find the optimal cost tree according to these cost functions. It is an interesting problem to investigate cost functions and algorithms for hierarchical clustering induced by other families of Δ that arise from a k-HST on V, i.e., if the cost of T is defined as

$$\mathrm{cost}_\Delta(T) := \sum_{i,j \in E(K_n)} \kappa(i,j)\Delta(\mathrm{lca}(i,j)). \tag{34}$$

Note that not all choices of Δ lead to a meaningful cost function. For example, choosing $\Delta(u) = \mathrm{diam}(T[u] - 1)$ gives rise to the following cost function

$$\mathrm{cost}(T) := \sum_{\{i,j\} \in E(K_n)} \kappa(i,j)\,\mathrm{dist}_T(i,j) \tag{35}$$

where $\mathrm{dist}_T(i,j)$ is the length of the unique path from i to j in T. In this case, the trivial clustering r, T^* where T^* is the star graph with V as its leaves and r as the root is always a minimizer; in other words, there is no incentive for spreading out the hierarchical clustering. Also worth mentioning is a long line of related work on fitting tree metrics to metric spaces (see e.g., [Ailon and Charikar, 2005, Räcke, 2008, Fakcharoenphol et al., 2003]). In this setting, the data points V are assumed to come from a metric space d_V and the objective is to find a hierarchical clustering T so as to minimize $\|d_V - d_T\|_p$. If the points in V lie on the unit sphere and the simlarity function κ is the cosine similarity $\kappa_{cos}(i,j) = 1 - d_V(i,j)/2$, then the problem of fitting a tree metric with $p = 2$ minimizes the same objective as cost function (35). Since $d_V \leq 1$ in this case, the minimizer is the trivial tree r, T^* (as remarked above). In general, when the points in V are not constrained to lie on the unit sphere, the two problems are incomparable.

Another direction for future research is to investigate combinatorial algorithms for solving LP-ultrametric and f-LP-ultrametric, since using an LP solver does not scale well to large data sets. Fast combinatorial algorithms

for approximately computing spreading metrics on a graph were discussed in [Even et al., 1999] using the framework of [Young, 1995, Plotkin et al., 1995] which may be seen as a special case of the Multiplicative Weights Update (MWU) method [Arora et al., 2012]. It is not too difficult to show that one can use ideas from [Even et al., 1999] to cast LP-ultrametric and f-LP-ultrametric as *packing* and *covering* type problems; however whether using the MWU algorithm on such a formulation leads to a practical speedup is something we have not explored yet.

References

[Ackerman et al., 2010] Ackerman, M., Ben-David, S., and Loker, D. (2010). Characterization of linkage-based clustering. In *COLT*, pages 270–281. Citeseer. 2

[Ailon and Charikar, 2005] Ailon, N. and Charikar, M. (2005). Fitting tree metrics: Hierarchical clustering and phylogeny. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 73–82. IEEE. 2, 3, 27

[Arora et al., 2012] Arora, S., Hazan, E., and Kale, S. (2012). The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164. 28

[Arora et al., 2009] Arora, S., Rao, S., and Vazirani, U. (2009). Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):5. 2, 31

[Awasthi et al., 2015] Awasthi, P., Bandeira, A. S., Charikar, M., Krishnaswamy, R., Villar, S., and Ward, R. (2015). Relax, no need to round: Integrality of clustering formulations. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 191–200. ACM. 2

[Balcan et al., 2008] Balcan, M.-F., Blum, A., and Vempala, S. (2008). A discriminative framework for clustering via similarity functions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 671–680. ACM. 1

[Bartal, 1996] Bartal, Y. (1996). Probabilistic approximation of metric spaces and its algorithmic applications. In *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, pages 184–193. IEEE. 25

[Bartal et al., 2001] Bartal, Y., Bollobás, B., and Mendel, M. (2001). A ramsey-type theorem for metric spaces and its applications for metrical task systems and related problems. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 396–405. IEEE. 25

[Bartal et al., 2003] Bartal, Y., Linial, N., Mendel, M., and Naor, A. (2003). On metric ramsey-type phenomena. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 463–472. ACM. 25

[Braun et al., 2015] Braun, G., Pokutta, S., and Roy, A. (2015). Strong reductions for extended formulations. *CoRR*, abs/1512.04932. 31, 32, 34

[Chan et al., 2013] Chan, S. O., Lee, J., Raghavendra, P., and Steurer, D. (2013). Approximate constraint satisfaction requires large lp relaxations. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 350–359. IEEE. 34

[Charikar et al., 1999] Charikar, M., Guha, S., Tardos, É., and Shmoys, D. B. (1999). A constant-factor approximation algorithm for the k-median problem. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 1–10. ACM. 2

[Charikar et al., 2003] Charikar, M., Guruswami, V., and Wirth, A. (2003). Clustering with qualitative information. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 524–533. IEEE. 3, 12

[Charikar and Li, 2012] Charikar, M. and Li, S. (2012). A dependent lp-rounding approach for the k-median problem. In *Automata, Languages, and Programming*, pages 194–205. Springer. 2

[Dasgupta, 2016] Dasgupta, S. (2016). A cost function for similarity-based hierarchical clustering. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 118–127. 1, 2, 3, 4, 18, 32, 33, 34

[Dasgupta and Long, 2005] Dasgupta, S. and Long, P. M. (2005). Performance guarantees for hierarchical clustering. *Journal of Computer and System Sciences*, 70(4):555–569. 2

[Di Summa et al., 2015] Di Summa, M., Pritchard, D., and Sanità, L. (2015). Finding the closest ultrametric. *Discrete Applied Mathematics*, 180:70–80. 3, 7

[Even et al., 1999] Even, G., Naor, J., Rao, S., and Schieber, B. (1999). Fast approximate graph partitioning algorithms. *SIAM Journal on Computing*, 28(6):2187–2214. 3, 11, 12, 13, 14, 28

[Even et al., 2000] Even, G., Naor, J. S., Rao, S., and Schieber, B. (2000). Divide-and-conquer approximation algorithms via spreading metrics. *Journal of the ACM (JACM)*, 47(4):585–616. 11

[Fakcharoenphol et al., 2003] Fakcharoenphol, J., Rao, S., and Talwar, K. (2003). A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 448–455. ACM. 27

[Felsenstein and Felenstein, 2004] Felsenstein, J. and Felenstein, J. (2004). *Inferring phylogenies*, volume 2. Sinauer Associates Sunderland. 2

[Friedman et al., 2001] Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin. 2

[Garg et al., 1996] Garg, N., Vazirani, V. V., and Yannakakis, M. (1996). Approximate max-flow min-(multi) cut theorems and their applications. *SIAM Journal on Computing*, 25(2):235–251. 3, 12

[Gurobi Optimization, 2015] Gurobi Optimization, I. (2015). Gurobi optimizer reference manual. 23

[Jain et al., 2003] Jain, K., Mahdian, M., Markakis, E., Saberi, A., and Vazirani, V. V. (2003). Greedy facility location algorithms analyzed using dual fitting with factor-revealing lp. *Journal of the ACM (JACM)*, 50(6):795–824. 2

[Jain and Vazirani, 2001] Jain, K. and Vazirani, V. V. (2001). Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of the ACM (JACM)*, 48(2):274–296. 2

[Jardine and Sibson, 1968] Jardine, N. and Sibson, R. (1968). The construction of hierarchic and non-hierarchic classifications. *The Computer Journal*, 11(2):177–184. 2

[Jardine and Sibson, 1971] Jardine, N. and Sibson, R. (1971). Mathematical taxonomy. *London etc.: John Wiley*. 2

[Krauthgamer et al., 2009] Krauthgamer, R., Naor, J. S., and Schwartz, R. (2009). Partitioning graphs into balanced components. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 942–949. Society for Industrial and Applied Mathematics. 3, 11

[Lee et al., 2015] Lee, J. R., Raghavendra, P., and Steurer, D. (2015). Lower bounds on the size of semidefinite programming relaxations. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 567–576. ACM. 34

[Leighton and Rao, 1988] Leighton, T. and Rao, S. (1988). An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Foundations of Computer Science, 1988., 29th Annual Symposium on*, pages 422–431. IEEE. 2, 3, 12

[Leighton and Rao, 1999] Leighton, T. and Rao, S. (1999). Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM (JACM)*, 46(6):787–832. 2, 3

[Li and Svensson, 2013] Li, S. and Svensson, O. (2013). Approximating k-median via pseudo-approximation. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 901–910. ACM. 2

[Lichman, 2013] Lichman, M. (2013). UCI machine learning repository. 24, 25

[Meilă and Heckerman, 2001] Meilă, M. and Heckerman, D. (2001). An experimental comparison of model-based clustering methods. *Machine learning*, 42(1-2):9–29. 24

[Peng and Wei, 2007] Peng, J. and Wei, Y. (2007). Approximating k-means-type clustering via semidefinite programming. *SIAM Journal on Optimization*, 18(1):186–205. 2

[Peng and Xia, 2005] Peng, J. and Xia, Y. (2005). A new theoretical framework for k-means-type clustering. In *Foundations and advances in data mining*, pages 79–96. Springer. 2

[Plotkin et al., 1995] Plotkin, S. A., Shmoys, D. B., and Tardos, É. (1995). Fast approximation algorithms for fractional packing and covering problems. *Mathematics of Operations Research*, 20(2):257–301. 28

[Räcke, 2008] Räcke, H. (2008). Optimal hierarchical decompositions for congestion minimization in networks. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 255–264. ACM. 27, 31

[Recht et al., 2012] Recht, B., Re, C., Tropp, J., and Bittorf, V. (2012). Factoring nonnegative matrices with linear programs. In *Advances in Neural Information Processing Systems*, pages 1214–1222. 2

[Schrijver, 1998] Schrijver, A. (1998). *Theory of linear and integer programming*. John Wiley & Sons. 12

[Sneath et al., 1973] Sneath, P. H., Sokal, R. R., et al. (1973). *Numerical taxonomy. The principles and practice of numerical classification.* 2

[Ward Jr, 1963] Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244. 24

[Young, 1995] Young, N. E. (1995). Randomized rounding without solving the linear program. In *SODA*, volume 95, pages 170–178. 28

[Zadeh and Ben-David, 2009] Zadeh, R. B. and Ben-David, S. (2009). A uniqueness theorem for clustering. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 639–646. AUAI Press. 2

LP and SDP hardness of finding the optimal hierarchical clustering

In this section we show that no polynomial sized Linear Program (LP) or Semidefinite Program (SDP) can approximate the hierarchical clustering problem with cost function (1) to within any constant factor. Note that the individual layer-t problem f-ILP-layer for $t = n/2$ is equivalent to the *minimum bisection problem* for which the best known true approximation is $O(\log n)$ due to [Räcke, 2008], while the best known bi-criteria approximation is $O\left(\sqrt{\log n}\right)$ due to [Arora et al., 2009] and improving these approximation factors is a major open problem. However it isn't clear if an improved approximation algorithm for hierarchical clustering under cost function (1) would imply an improved algorithm for every layer-t problem, which is why a constant factor inapproximability result is still interesting. Our main tool is to use the framework of [Braun et al., 2015] where a notion of reductions between different optimization problems was developed, relating the integrality gap of LP and SDP relaxations for the source problem to the integrality gaps of corresponding relaxations for the target problem. We briefly recall the exact notion of an optimization problem in the framework of [Braun et al., 2015].

Definition 36 (Optimization problem)**.** *[Braun et al., 2015] An* optimization problem *is a tuple* $\mathcal{P} = (\mathcal{S}, \mathfrak{I}, \mathrm{val})$ *consisting of a set* $\mathcal{S}$ *of* feasible solutions*, a set* $\mathfrak{I}$ *of* instances*, and a real-valued objective called* measure $\mathrm{val}\colon \mathfrak{I} \times \mathcal{S} \to \mathbb{R}$*. We shall use* $\mathrm{val}_{\mathcal{I}}(s)$ *for the objective value of a feasible solution* $s \in \mathcal{S}$ *for an instance* $\mathcal{I} \in \mathfrak{I}$*.*

Since we are interested in the integrality gaps of LP and SDP relaxations for an optimization problem $\mathcal{P} = (\mathcal{S}, \mathfrak{I}, \mathrm{val})$, we represent the approximation gap by two functions $C, S : \mathfrak{I} \to \mathbb{R}$ where C is the *completeness guarantee* while S is the *soundness guarantee*. Note that the ratio C/S represents the approximation factor for the problem $\mathcal{P}$. We recall below the formal definition of an LP relaxation of $\mathcal{P}$ that achieves a (C, S)-approximation guarantee. We assume without loss of generality that $\mathcal{P}$ is a maximization problem.

Definition 37 (LP formulation of an optimization problem)**.** *[Braun et al., 2015] Let* $\mathcal{P} = (\mathcal{S}, \mathfrak{I}, \mathrm{val})$ *be an optimization problem, and* $C, S : \mathfrak{I} \to \mathbb{R}$*. Then let* $\mathfrak{I}^S := \{\mathcal{I} \in \mathfrak{I} \mid \max \mathrm{val}_{\mathcal{I}} \leq S(\mathcal{I})\}$ *denote the set of* sound *instances, i.e., for which the soundness guarantee* S *is an upper bound on the maximum. A* (C, S)-approximate LP formulation *of* $\mathcal{P}$ *consists of a linear program* $Ax \leq b$ *with* $x \in \mathbb{R}^r$ *for some* r *and the following* realizations*:*

Feasible solutions *as vectors* $x^s \in \mathbb{R}^r$ *for every* $s \in \mathcal{S}$ *satisfying*

$$Ax^s \leq b \qquad \text{for all } s \in \mathcal{S}, \tag{36}$$

i.e., the system $Ax \leq b$ *is a relaxation of* $\mathrm{conv}\left(x^s \mid s \in \mathcal{S}\right)$*.*

Instances *as affine functions* $w_{\mathcal{I}}\colon \mathbb{R}^r \to \mathbb{R}$ *for all* $\mathcal{I} \in \mathfrak{I}^S$ *satisfying*

$$w_{\mathcal{I}}(x^s) = \mathrm{val}_{\mathcal{I}}(s) \qquad \text{for all } s \in \mathcal{S}, \tag{37}$$

i.e., the linearization $w_{\mathcal{I}}$ *of* $\mathrm{val}_{\mathcal{I}}$ *is required to be exact on all* x^s *with* $s \in \mathcal{S}$*.*

Achieving (C, S) **approximation guarantee** *by requiring*

$$\max\{w_{\mathcal{I}}(x) \mid Ax \leq b\} \leq C(\mathcal{I}) \qquad \text{for all } \mathcal{I} \in \mathfrak{I}^S, \tag{38}$$

The size *of the formulation is the number of inequalities in* $Ax \leq b$*. Finally, the* (C, S)*-approximate* LP formulation complexity $\mathrm{fc}_{\mathrm{LP}}(\mathcal{P}, C, S)$ *of* $\mathcal{P}$ *is the minimal size of all its LP formulations.*

One can similarly define a (C, S)-approximate SDP formulation for a problem $\mathcal{P}$ where instead of a LP, we now have a SDP relaxation $\mathcal{A}(X) = b$ with $X \in \mathbb{S}^r_+$ and where $\mathbb{S}^r_+$ denotes the space of $r \times r$ positive semidefinite matrices. The size of such an SDP formulation is measured by the dimension r and $\mathrm{fc}_{\mathrm{SDP}}(\mathcal{P}, C, S)$ is defined as the minimum size of an SDP formulation achieving (C, S)-approximation for problem $\mathcal{P}$. Below we recall the precise notion of a reduction between two problems as in [Braun et al., 2015].

Definition 38 (Reduction)**.** *[Braun et al., 2015] Let $\mathcal{P}_1 = (\mathcal{S}_1, \mathcal{I}_1, \mathrm{val})$ and $\mathcal{P}_2 = (\mathcal{S}_2, \mathcal{I}_2, \mathrm{val})$ be optimization problems with guarantees C_1, S_1 and C_2, S_2, respectively. Let $\tau_1 = +1$ if $\mathcal{P}_1$ is a maximization problem, and $\tau_1 = -1$ if $\mathcal{P}_1$ is a minimization problem. Similarly, let $\tau_2 = \pm 1$ depending on whether $\mathcal{P}_2$ is a maximization problem or a minimization problem.*
A reduction *from $\mathcal{P}_1$ to $\mathcal{P}_2$ respecting the guarantees consists of*

1. *two mappings: $*\colon \mathcal{I}_1 \to \mathcal{I}_2$ and $*\colon \mathcal{S}_1 \to \mathcal{S}_2$ translating instances and feasible solutions independently;*
2. *two nonnegative $\mathcal{I}_1 \times \mathcal{S}_1$ matrices M_1, M_2*

subject to the conditions

$$\tau_1 \left[C_1(\mathcal{I}_1) - \mathrm{val}_{\mathcal{I}_1}(s_1)\right] = \tau_2 \left[C_2(\mathcal{I}_1^*) - \mathrm{val}_{\mathcal{I}_1^*}(s_1^*)\right] M_1(\mathcal{I}_1, s_1) + M_2(\mathcal{I}_1, s_1) \qquad \text{(39-complete)}$$

$$\tau_2 \,\mathrm{OPT}\,(\mathcal{I}_1^*) \le \tau_2 S_2(\mathcal{I}_1^*) \qquad \text{if } \tau_1 \,\mathrm{OPT}\,(\mathcal{I}_1) \le \tau_1 S_1(\mathcal{I}_1). \qquad \text{(39-sound)}$$

The matrices M_1 and M_2 control the parameters of the reduction relating the integrality gap of relaxations for $\mathcal{P}_1$ to the integrality gap of corresponding relaxations for $\mathcal{P}_2$. For a matrix A, let $\mathrm{rk}_+ A$ and $\mathrm{rk}_{\mathrm{psd}} A$ denote the nonnegative rank and psd rank of A respectively. The following theorem is a restatement of Theorem 3.2 from [Braun et al., 2015] ignoring constants.

Theorem 39. *[Braun et al., 2015] Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be optimization problems with a reduction from $\mathcal{P}_1$ to $\mathcal{P}_2$ respecting the completeness guarantees C_1, C_2 and soundness guarantees S_1, S_2 of $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively. Then*

$$\mathrm{fc}_{\mathrm{LP}}(\mathcal{P}_1, C_1, S_1) \le \mathrm{rk}_+ M_2 + \mathrm{rk}_+ M_1 + \mathrm{rk}_+ M_1 \cdot \mathrm{fc}_{\mathrm{LP}}(\mathcal{P}_2, C_2, S_2), \qquad (40)$$

$$\mathrm{fc}_{\mathrm{SDP}}(\mathcal{P}_1, C_1, S_1) \le \mathrm{rk}_{\mathrm{psd}} M_2 + \mathrm{rk}_{\mathrm{psd}} M_1 + \mathrm{rk}_{\mathrm{psd}} M_1 \cdot \mathrm{fc}_{\mathrm{SDP}}(\mathcal{P}_2, C_2, S_2), \qquad (41)$$

where M_1 and M_2 are the matrices in the reduction as in Definition 38.

Therefore to obtain a lower bound for problem $\mathcal{P}_2$, it suffices to find a source problem $\mathcal{P}_1$ and matrices M_1 and M_2 of low nonnegative and psd rank, satisfying Definition 38. We will choose the source problem to be NAESAT* together with the NP-hardness reduction from [Dasgupta, 2016] with some minor modifications to fit this framework. We first define the optimization problem NAESAT*.

Definition 40 (NAESAT* as an optimization problem)**.** *An instance $\mathcal{I}$ of* NAESAT* *on n variables consists of two types clauses: m clauses with 3 literals and m' clauses with 2 variables. Every variable occurs exactly 3 times - once in every 3-clause and twice with opposite polarities in 2-clauses, so that $3m = n$ and $m' = n$. A solution s of* NAESAT* *consists of a $\{0, 1\}$ assignment to the n variables and satisfies a clause if there is both a 0 and a 1 literal in that clause. The objective value $\mathrm{val}_{\mathcal{I}}(s)$ counts the number of clauses of $\mathcal{I}$ that are satisfied by s.*

Similarly, we can cast the problem of obtaining a hierarchical clustering of minimum cost under cost function (1) as an optimization problem as in the language of Definition 36.

Definition 41 (HCLUST as an optimization problem)**.** *An instance $\mathcal{I}$ of* HCLUST *on n points consists of a similarity function $\kappa : [n] \times [n] \to \mathbb{R}$. A solution T of* HCLUST *is a hierarchical clustering on this set of n points. The objective value $\mathrm{val}_{\mathcal{I}}(T)$ is given by cost function (1).*

We are now ready to describe the reduction. We first describe the two mappings translating instances and solutions of NAESAT* to instances of HCLUST. The form of the two matrices M_1 and M_2 will become clear from this analysis. Note that a similarity function $\kappa : [n] \times [n] \to \mathbb{R}$ can be thought of as an edge-weighted complete graph on n nodes. For an edge weighted graph G on n nodes and a hierarchical clustering T on n points, we will denote by $\mathrm{cost}_G(T)$ the cost of T according to cost function (1) and where the similarity function is inherited from the weight of G.

Mapping instances: Given an instance $\mathcal{I}$ of NAESAT* we map it to the instance $\mathcal{I}^*$ of HCLUST. Let us first define a weighted graph G on $2n$ nodes and $6m + 2m' + n$ edges. For every variable x_i there are nodes in G corresponding to x_i and $\overline{x}_i$; thus G has a total of $2n$ vertices. There are edges of weight $W = 2nm + 1$ connecting x_i and $\overline{x}_i$ for every i. For every 3-clause with literals l_i, l_j, l_k we add edges of weight 1 between $\{l_i, l_j, l_k\}$ and between $\left\{\overline{l}_i, \overline{l}_j, \overline{l}_k\right\}$. For every 2-clause with literals l_i, l_j we add edges of weight 2 between $\{l_i, l_j\}$ and between $\left\{\overline{l}_i, \overline{l}_j\right\}$. This completes the description of the graph G. Instance $\mathcal{I}^*$ will be the complement of the graph G where the complement is taken with respect to a weighted K_{2n} with every edge having weight W. Note that the cost of T with respect to this weighted K_{2n} is $\left(\frac{8n^3-2n}{3}\right) W$ (see Theorem 3 of [Dasgupta, 2016]) so that by definition, we have

$$\mathrm{val}_{\mathcal{I}^*}(T) = \left(\frac{8n^3 - 2n}{3}\right) W - \mathrm{cost}_G(T) = \frac{16n^5 + 20n^3 - 6n}{9} - \mathrm{cost}_G(T).$$

Mapping solutions: Let the $2n$ points of instance $\mathcal{I}^*$ be denoted by V. Given a solution s we map it to the following hierarchical clustering T^* on V. The first level consists of two nodes $V^+ := \{l_i \mid s(l_i) = 1\}$ and $V^- := \{l_i \mid s(l_i) = 0\}$. The second layer splits V^+ and V^- completely into leaves. Note that this mapping of solutions is independent of the instances.

The following lemma outlines the relationship between $\mathrm{val}_{\mathcal{I}}(s)$ and $\mathrm{val}_{\mathcal{I}^*}(T^*)$ so that we may use Theorem 39 for appropriate choices of the reduction paramters.

Lemma 42. *Let $\mathcal{I}$ be an instance of* NAESAT* *and let $\mathcal{I}^*$ be the instance of* HCLUST *it is mapped to, as in the above description. Then we have the following*

$$\frac{16n^5 + 20n^3 - 6n}{9} - \mathrm{val}_{\mathcal{I}^*}(T^*) = \frac{4n^4}{3} + 8n^2 + 4n\,\mathrm{val}_{\mathcal{I}}(s).$$

Proof. It suffices to show that $\mathrm{cost}_G(T^*) = \frac{4n^4}{3} + 8n^2 + 4n\,\mathrm{val}_{\mathcal{I}}(s)$. We split the $\mathrm{cost}_G(T^*)$ according to the contribution of different classes of edges of G. We have the following types of edges:

1. **Edges connecting x_i and $\overline{x}_i$**: Each such edge is split at the first level and so the number of leaves of the subtree of its lca is $2n$. There are n such edges each of weight W and so contribute $2n^2W$ to the cost.

2. **Edges from satisfied 3-clauses**: Since this clause is satisfied, two of its edges are split at the first level while the third edge gets split at the second level. Let the number of satisfied 3-clauses be $\widetilde{m}$, then there are $2\widetilde{m}$ triangles in G. The total contribution by these edges is $2(2\widetilde{m} * 2n + \widetilde{m} * n) = 10n\widetilde{m}$.

3. **Edges from unsatisfied 3-clauses**: Note that such triangles do not get split at the first level since they belong entirely to either V^+ or V^-. The number of such triangles are $2m - 2\widetilde{m}$. Thus the total contribution to the cost due to these edges is $3 * 2 * (m - \widetilde{m}) = 6nm - 6n\widetilde{m}$.

4. **Edges from satisfied 2-clauses**: Such edges are split at the first level since they are satisfied. Let $\widetilde{m}'$ denote the number of satisfied 2-clauses, then the number of such edges is $2\widetilde{m}'$. They contribute $2 * 2\widetilde{m}' * 2n = 8n\widetilde{m}'$ to the total cost.

5. **Edges from unsatisfied 2-clauses**: These edges get split at the second level. Since there are $2m' - 2\widetilde{m}'$ such edges, they contribute a total of $2 * 2(m' - \widetilde{m}') * n = 4nm' - 4n\widetilde{m}'$.

Thus the total cost of T^* w.r.t G is $2n^2W + 6nm + 4n\widetilde{m} + 4nm' + 4n\widetilde{m}' = 2n^2W + 6n^2 + 4n\,\mathrm{val}_{\mathcal{I}}(s)$. The claim of the lemma follows by plugging in the expression for W and using the fact that $\mathrm{val}_{\mathcal{I}^*}(T^*) = \frac{16n^5+20n^3-6n}{9} - \mathrm{cost}_G(T^*)$. □

Let the completeness guarantee for NAESAT* be $C_1(\mathcal{I})$ and the completeness guarantee for hierarchical clustering be $C_2(\mathcal{I}^*)$. Then we have the following expressions for M_1, M_2, and C_2 (see [Braun et al., 2015]):

$$C_2(\mathcal{I}^*) = \frac{16n^5 - 12n^4 + 20n^3 - 72n^2 - 6n}{9} \tag{42}$$

$$M_1(\mathcal{I}, s) = \frac{1}{4n} \tag{43}$$

$$M_2(\mathcal{I}, s) = C_1(\mathcal{I}) \tag{44}$$

It is easy to see that M_1 and M_2 both have $O(1)$ nonnegative and psd ranks. The soundness guarantee follows by choosing

$$S_2(\mathcal{I}^*) = \frac{16n^5 - 12n^4 + 20n^3 - 120n^2 - 6n}{9}$$

and observing that if there is a tree T that costs at most $S_2(\mathcal{I}^*)$ on $\mathcal{I}^*$, then $\mathrm{cost}_G(T) \geq 10nm + 8nm' + 2n^2W$. The rest of the argument is similar to [Dasgupta, 2016], i.e., note that such a tree T must necessarily split the opposite polarity vertices, since otherwise its cost in G is at most $2n(6m + 4m' + nW) - W < 10nm + 8nm' + 2n^2W$. This split clearly must leave at least one edge per triangle unsplit; however it must cut all the other edges since othewise its cost again falls below $10nm + 8nm' + 2n^2W$. Thus it is actually "not all equal satisfiable", and so clearly $OPT(\mathcal{I}) \geq S_1(\mathcal{I})$, where S_1 is the soundness guarantee for NAESAT*. Note that completeness and soundness parameters $C_1(\mathcal{I})$ and $S_1(\mathcal{I})$ for NAESAT* can be chosen to be a constant due to the well known reduction of 3-SAT to NAESAT and the reduction from NAESAT to NAESAT* as in [Dasgupta, 2016], so that by the results of [Chan et al., 2013, Lee et al., 2015] we have

$$\mathrm{fc}_{\mathrm{LP}}\left(\text{NAESAT*}, C_1, S_1\right) \geq n^{\Omega(\log n/\log\log n)},$$
$$\mathrm{fc}_{\mathrm{SDP}}\left(\text{NAESAT*}, C_1, S_1\right) \geq n^{\Omega(\log n/\log\log n)}.$$

Thus applying Theorem 39 and choosing n to be large enough, it follows that no polynomial sized LP or SDP can give a constant factor approximation for the problem HCLUST.