Fitting tree metrics: Hierarchical clustering and Phylogeny

Nir Ailon[*]
Princeton University
Princeton, NJ
nailon@cs.princeton.edu

Moses Charikar[†]
Princeton University
Princeton, NJ
moses@cs.princeton.edu

Abstract

Given dissimilarity data on pairs of objects in a set, we study the problem of fitting a tree metric to this data so as to minimize additive error (i.e. some measure of the difference between the tree metric and the given data). This problem arises in constructing an M-level hierarchical clustering of objects (or an ultrametric on objects) so as to match the given dissimilarity data – a basic problem in statistics. Viewed in this way, the problem is a generalization of the correlation clustering problem (which corresponds to $M = 1$). We give a very simple randomized combinatorial algorithm for the M-level hierarchical clustering problem that achieves an approximation ratio of $M+2$. This is a generalization of a previous factor 3 algorithm for correlation clustering on complete graphs. The problem of fitting tree metrics also arises in phylogeny where the objective is to learn the evolution tree by fitting a tree to dissimilarity data on taxa. The quality of the fit is measured by taking the ℓ_p norm of the difference between the tree metric constructed and the given data. Previous results obtained a factor 3 approximation for finding the closest tree tree metric under the ℓ_∞ norm. No non-trivial approximation for general ℓ_p norms was known before. We present a novel LP formulation for this problem and obtain an $O((\log n \log\log n)^{1/p})$ approximation using this. En route, we obtain an $O((\log n \log\log n)^{1/p})$ approximation for the closest ultrametric under the ℓ_p norm. Our techniques are based on representing and viewing an ultrametric as a hierarchy of clusterings, and may be useful in other contexts.

1. Introduction

We consider the problem of finding a tree metric to fit dissimilarity data on pairs of objects from a given set X. A tree metric is defined by a weighted tree spanning X, with distances between a pair of objects determined by the sum of edge weights along the unique path in the tree connecting them. The main problem we consider is: *How well can we construct a tree metric to fit the given data ?* A special kind of tree metric is an ultrametric, where the underlying tree has a special structure: all elements of X are leaves of the tree and all leaves are at the same distance from the root. Ultrametrics naturally correspond to a hierarchy of clusterings of the data. Another question of great interest is: *How well can we construct an ultrametric to fit the given data ?*

Such problems referred to as *numerical taxonomy* arise naturally in numerous disciplines: in statistics – for clustering data into hierarchies, and in sciences such as linguistics and biology (see the survey [14]), where tree metrics represent evolutionary branching processes that give rise to the observed data. Consequently, such problems have been studied quite extensively. (see the paper of Agarwala etal [1] and the references therein).

When the given data can be realized exactly by a tree metric (or an ultrametric), it is well known that the underlying tree structure can be reconstructed. In fact, succinct necessary and sufficient conditions are known for checking whether a given dissimilarity function can be exactly realized thus, involving checking a certain criterion on all sets of 3 points (for ultrametrics) or 4 points (for tree metrics).

While the exact problem is well solved, finding the best fitting tree metric when none fits exactly is a much harder problem. In order to quantify the quality of the fit, we view a distance function on n objects as a vector with $\binom{n}{2}$ coordinates corresponding to pairwise distances. The fit between a given dissimilarity function D

[*] Partially supported by a Charlotte Elizabeth Procter Fellowship. Part of this work was done while visiting Microsoft Research.

[†] Supported by NSF ITR grant CCR-0205594, DOE Early Career Principal Investigator award DE-FG02-02ER25540, NSF CAREER award CCR-0237113, an Alfred P. Sloan Fellowship and a Howard B. Wentz Jr. Junior Faculty Award.

and a tree metric d_T is then measured by the ℓ_p norm $\|D - d_T\|_p$. The goal is to find a tree metric d_T so as to minimize this quantity.

1.1. Related Work

Farach etal [9] showed that under the ℓ_∞ norm, an optimal ultrametric can be computed in polynomial time. Unfortunately, these fitting problems are NP-Hard for various other norms of interest ℓ_1, ℓ_2 (for trees and ultrametrics) and ℓ_∞ (for tree metrics). In fact the ℓ_∞ problem (for tree metrics) and the ℓ_1 problem for tree metrics and ultrametrics are APX-Hard (see Wareham [17, 1]). The only non-trivial approximation result is the 3-approximation of Agarwala etal [1] for the closest tree metric under the ℓ_∞ norm. Their work makes an interesting connection between the closest tree metric and closest ultrametric problem. They show that an α-approximation for a restricted version of the closest ultrametric problem yields a 3α approximation for the closest tree metric problem for any ℓ_p norm. This is the basis for their 3 approximation for ℓ_∞ and we use this later in our results for ℓ_p. Recently, connections have been made between these results for the ℓ_∞ best ultrametric and some classical results in mathematics [3, 13]. Ma etal [15] considered the problem of finding the best ℓ_p fit by an ultrametric where distances in the ultrametric are no smaller that the given data. For this problem, they obtained an $O(n^{1/p})$ approximation. Recently, Dhamdhere [6] considered the problem of finding a line metric to minimize additive distortion from the given data (measured by the ℓ_1 norm) and obtained an $O(\log n)$ approximation. In fact, his motivation for considering this problem was to develop techniques that might be useful for finding the closest tree metric with distance measured by the ℓ_1 norm. Independently of our work, Harb, Kannan and McGregor [12] recently developed a factor $O(\min\{n^{1/p}, (k \log n)^{1/p}\})$ approximation for the closest tree metric under the ℓ_p norm where k is the number of distinct distances in the input. Of course, there is rich literature on metric embedding problems where the measure of interest is the *multiplicative* distortion. Several such problems have been studied in the context of approximating metric spaces via tree metrics (e.g. [8]). Researchers have also studied reconstruction of phylogenies under stochastic models of evolution (see Mossel etal [16] and the references therein).

1.2. Our Results

We make significant improvements to the state of the art for fitting ultrametrics and tree metrics to given data so as to minimize additive distortion according to the ℓ_p measure of fit. We present two main results.

First we consider the problem of fitting an ultrametric to dissimilarity data specified as integers in $\{1, \ldots, M+1\}$. This naturally corresponds to finding an M-level hierarchical clustering to best match the given data. In fact, the $M = 1$ problem is exactly the correlation clustering problem on complete graphs [4, 5]. This problem has received a lot of attention recently. We generalize the algorithm of [2] to obtain a simple randomized combinatorial algorithm for fitting an M-level hierarchical clustering with an approximation ratio of $M + 2$.[1] The algorithm is quite intuitive and proceeds by recursively modifying the given data so as to eventually produce an ultrametric. Even though the algorithm is completely combinatorial, the analysis proceeds by constructing a dual solution to a certain LP and the values in this dual solution are defined in terms of the probability distribution over the algorithm's actions. We describe these results in Section 2.

Secondly, we consider the problem of fitting ultrametrics and tree metrics to general dissimilarity data. For the problem of fitting an ultrametric, we introduce a novel LP formulation which arises from viewing an ultrametric as a hierarchy of clusterings (see Section 3). The closest ultrametric problem now becomes a hierarchy of correlation clustering problems which are dependent in a certain way. In Section 3.1, we show how to round the LP solution, consisting of a hierarchy of metrics, to obtain an $O((\log n \log\log n)^{1/p})$ approximation for the ℓ_p norm. (This follows from a Seymour-style analysis of the divide and conquer approach). The LP based method is fairly flexible, allowing imposing upper and lower bounds as well as equality constraints on certain pairs of distances in the final ultrametric obtained. This flexibility enables us to use the ultrametric result to obtain an $O((\log n \log\log n)^{1/p})$ approximation for fitting tree metrics in Section 3.3 via the results of Agarwala etal [1].

2. Hierarchical Clustering

Let X be a ground set of n elements, and let $M > 0$ be a constant integer. A level M hierarchical clustering of X is a rooted tree with the elements of X as leaves and a path of length exactly $M + 1$ from the root to any leaf. For $M = 1$ this is the standard definition of a clus-

[1] When viewed as an M level hierarchical clustering, it makes sense to use small values of M not exceeding $O(\log n)$. This has a precise information-theoretic motivation which we will not pursue in this version.

tering of X: the children of the root can be viewed as the clusters. For $M = 2$ we have a standard clustering, with the additional structure that every cluster is further partitioned into clusters. This nested clustering generalizes to any M.

For a level M hierarchical clustering $\mathcal{C}$ we define a distance function $d_{\mathcal{C}}$ between distinct pairs $i, j \in X$. The distance $d_{\mathcal{C}}(i,j)$ is the height of the subtree rooted by the lowest common ancestor of i and j. So if i, j share a parent, the distance is 1. If i, j do not share a parent but they share a grandparent, the distance is 2, and so on, where the maximal distance is $M+1$. Depending on the clustering application, this distance would measure the extent to which i and j are dissimilar. Note that this is exactly half the tree distance between i and j. Abusing notation, we denote by $d_{\mathcal{C}}$ the $\binom{n}{2}$ coordinate vector of distances.

Let G be the complete undirected graph on X, with an integer weight function $1 \leq D(i,j) \leq M+1$, representing dissimilarity between pairs of elements in X. We view D as an $\binom{n}{2}$-coordinate vector. The *generalized correlation clustering* problem is defined as finding a level M clustering of X, $\mathcal{C}$, minimizing $\|D - d_{\mathcal{C}}\|_1$.

For $M = 1$, this is exactly the *correlation clustering* problem on a complete graph. Here $D(i,j) = 1$ represents a $+$ edge between i and j, and $D(i,j) = 2$ indicates a $-$ edge.

Claim 1 (Ultrametric property) *The distance function $d_{\mathcal{C}}$ satisfies the following strong triangle inequality property: For any distinct i, j, k, $d_{\mathcal{C}}(i,j) \leq \max\{d_{\mathcal{C}}(i,k), d_{\mathcal{C}}(j,k)\}$.*

Note that the strong triangle inequality implies that $d_{\mathcal{C}}(i,j) = \max\{d_{\mathcal{C}}(i,k), d_{\mathcal{C}}(j,k)\}$ if $d_{\mathcal{C}}(i,k) \neq d_{\mathcal{C}}(j,k)$.

Claim 2 *Any ultrametric $d : X \times X \rightarrow \{1, \ldots, M+1\}$ is induced by some level M clustering $\mathcal{C}$.*

The proof of Claim 2 is by simple induction: It is easy to verify that the relation $R_M \subseteq X \times X$ of all i, j such that $d(i,j) \leq M$ is an equivalence relation. Using induction on M, we build $(M-1)$-level clusterings (trees) on the equivalence classes and connect them as children of a new root, thus obtaining a clustering $\mathcal{C}_d$. It is immediate to verify that $d = d_{\mathcal{C}_d}$. Note that this proof is constructive. Claims 1 and 2 thus give us a local characterization of the distance functions induced by level M hierarchical clusterings.

Algorithm HCLUST-PIVOT (Figure 1) builds a level-M hierarchical clustering represented by an ultrametric z so as to fit a given dissimilarity function D. Before running the algorithm we set[2] $z = D$ and the algorithm progressively mutates z, converting it into an ultrametric. The hierarchical clustering $\mathcal{C}_z$ can be easily derived from the vector z *after* the algorithm returns.

```
HCLUST-PIVOT($X$)

  if $X = \emptyset$ return
  pick random pivot $k \in X$.
  for $t = 1, \ldots, M+1$
      set $X_t = \{i \in X : z(k,i) = t\}$
  for all $1 \leq t_1 < t_2 \leq M+1$
      for all $i \in X_{t_1}, j \in X_{t_2}$
(1)       change $z(i,j) = t_2$
  for all $1 \leq t \leq M+1$
      for all distinct $i,j \in X_t$
(2)       change $z(i,j) = \min\{z(i,j), t\}$

  for $t = 2, \ldots, M+1$
      run HCLUST-PIVOT($X_t$)
```

Figure 1. Algorithm HCLUST-PIVOT**. Before calling, set $z = D$. After execution, z is solution.**

Theorem 1 *Algorithm* HCLUST-PIVOT *is an expected $2 + M$ approximation algorithm for generalized correlation clustering.*

The techniques we use in the proof are similar to the ones used in the proof of Theorem 1 in [2].

Proof: We first prove correctness, in other words, that the vector z after the execution of the algorithm is an ultrametric. Fix a triple $T = \{h, i, j\} \subseteq X$. There are two possible important events in the life of T. The first event is that one of its vertices (say, h) is chosen as pivot when the other two are input to the same recursive call. But then either line (1) or (2) will fix the value of $z(i,j)$ so that z has the ultrametric property on $\{h, i, j\}$. (In fact, the algorithm will change the value of $z(i,j)$ in a greedy way that minimizes the size of the change.) Now it is immediate to verify that the values of $z(h,i)$ and $z(h,j)$ are frozen until termination, and that no future change of $z(i,j)$ will violate the ultrametric property on t. Indeed, the only way $z(i,j)$ will change in recursive calls is if $z(h,i) = z(h,j) = l$, in which case $z(i,j)$ is set in line

[2] Think of z as a global variable

(2) to be $\leq l$, and in the recursion on X_l its value cannot climb above l (see Observation 1 below). The other event is that a fourth vertex $k \notin T$ was chosen as pivot when all four h, i, j, k are in the same recursive call, and the vertices of T are split between more than one recursive calls (in other words $z(k,h), z(k,i), z(k,j)$ are not all equal). It is not hard to verify that the work of lines (1) and (2) will enforce the ultrametric property on t. There are three subcases. First subcase: $z(k,h) < z(k,i) = l_1 < z(k,j) = l_2$. In this subcase, $z(i,j)$ and $z(h,j)$ are mutated and frozen as l_2 in line (1), $z(h,i)$ is mutated and frozen as l_1 in line (1). Second subcase: $z(k,h) = z(k,i) = l_1 < z(k,j) = l_2$. Then $z(h,j)$, $z(i,j)$ are frozen as l_2 in line (1), and $z(h,i)$ is mutated in line (2) to a value not exceeding l_1, above which it will not climb in the recursion (see Observation 1 below). Third subcase: $z(k,h) = l_1 < z(k,i) = z(k,j) = l_2$. Then $z(h,i)$ and $z(h,j)$ are frozen as l_2 and $z(i,j)$ is mutated to a value not exceeding l_2, above which it will not climb in the recursion (see Observation 1 below). In all three subcases triangle T now satisfies the strong triangle inequality and will not violate it in the recursion. We conclude the proof of correctness by stating the obvious claim that either the first or the second event occurs (exactly once) on all triples T.

We start the approximation factor guarantee proof with the following:

Observation 1 *For all distinct $i,j \in X$ the value of $z(i,j)$ can either increase or decrease during the execution of the algorithm, but not both.*

To see this, assume that the value of $z(i,j)$ increased during the execution. The only possible increase of z can occur in line (1), after which i and j are separated into two different recursion branches. Therefore, the value of $z(i,j)$ will not change from that point and on. Assume that the value of $z(i,j)$ was decreased to z' at some point. If the decrease occurred in line (1), then again, the value of $z(i,j)$ will never change again because of the splitting of i,j into two recursion branches. If it occurred in line (2), then the new value of $z(i,j) = z'$ will be maximal among all values of z in the recursive call corresponding to $X_{z'}$. Consequently, the value of $z(i,j)$ will not climb above z'.

Let $\mathcal{T}$ be the set of triples $\{i,j,k\}$ such that the largest D-value among the three values $D(i,j)$, $D(i,k)$, $D(j,k)$ is strictly larger than the second largest value. (We call such triples "violating triples"). For $T \in \mathcal{T}$ let $\lambda_1(T)$ denote the largest D-value of T, $\lambda_2(T)$ the second largest D-value of T, and $\lambda_3(T)$ the lowest z-value of T (arbitrarily breaking ties between $\lambda_2(T)$ and $\lambda_3(T)$). Let $L(T) = \lambda_1(t) - \lambda_2(t) > 0$. For an integer $1 \leq b < M+1$, let $\mathcal{T}_b \subseteq \mathcal{T}$ denote the set of $T \in \mathcal{T}$ such that $\lambda_2(T) \leq b$ and $\lambda_1(T) > b$. Clearly, any solution has to pay at least $L(T)$ on the edges of T (that is, $\|D_T - z_T\|_1 \geq L(T)$ for any solution z, where z_T is the restriction of a vector in $\binom{X}{2}$ to coordinates $\binom{T}{2}$). Also, if $T \in \mathcal{T}_b$, then in any solution the value of at least one edge $e \in \binom{T}{2}$ has to "cross" b. In other words, if it started $< b$ it will be $\geq b$ and vice-versa. The following two LPs capture this:

> *LP 1*: Minimize $\sum_{e \in \binom{X}{2}} \alpha_e$, s.t. $\sum_{e \in \binom{T}{2}} \alpha_e \geq L(T)$ for all $T \in \mathcal{T}$ and $\alpha_e \geq 0$ for all $e \in \binom{X}{2}$. In the corresponding IP, the variable α_e (w.r.t. a feasible ultrametric solution d) tell us the amount of change of the value of $e \in \binom{X}{2}$ (contribution to the ℓ_1 distance, which is $|D_e - d_e|$). Note that a solution to the corresponding IP does *not* necessarily encode a feasible ultrametric (there are missing constraints). The important point is that its optimal value is a lower bound to the optimal solution to the ultrametric problem.
>
> *LP 2*: Minimize $\sum_{e \in \binom{X}{2}} \sum_{b=1}^{M} \gamma_e^b$ s.t. $\sum_{e \in \binom{T}{2}} \gamma_e^b \geq 1$ for all $1 \leq b < M+1$, $T \in \mathcal{T}_b$, and $\gamma_e^b \geq 0$ for all $e \in \binom{X}{2}, 1 \leq b < M+1$. In the corresponding IP, the meaning of indicator variable γ_e^b (w.r.t. a feasible ultrametric problem solution d) is as follows. If $\gamma_e^b = 1$ then the value of e crosses b when changing from D_e to d_e (therefore contributing 1 to the objective function). Again, the corresponding IP formulation does *not* contain all the ultrametric constraints.

Fix any solution z^* (i.e. an ultrametric on V) with cost $c = \|D - z^*\|_1$. The following are the *duals* to LP 1 and 2. Therefore, any feasible solutions are lower bounds for c^*.

> *Dual LP 1*: Maximize $\sum_{T \in \mathcal{T}} \beta_T L(T)$, subject to $\sum_{T \in \mathcal{T}: e \subseteq t} \beta_T \leq 1$ for all $e \in \binom{X}{2}$ and $\beta_T \geq 0$ for all $T \in \mathcal{T}$.
>
> *Dual LP 2*: Maximize $\sum_{b=1}^{M} \sum_{T \in \mathcal{T}_b} \delta_T^b$, subject to $\sum_{T \in \mathcal{T}_b : e \subseteq T} \delta_T^b \leq 1$ for all $e \in \binom{X}{2}, 1 \leq b < M+1$, and $\delta_T^b \geq 0$ for all $1 \leq b < M+1, T \in \mathcal{T}_b$.

Notice that throughout the execution of the algorithm no new violating triples are created. Fix a triple $T = \{i,j,k\} \in \mathcal{T}$. The triple will be charged if one of its vertices, say i, was chosen as pivot when the other two were in the same recursive call, and the value $z(j,k)$ was mutated. The amount of charge is the size of the change. Every unit of cost paid by the solution returned by the algorithm is charged to exactly one triple in $\mathcal{T}$. By Observation 1, the total cost of the solution returned by the algorithm is exactly the total amount of charge over all triples (the observation ensures that there are no cancellations). Every triple can be charged at most once. Not all triples $T \in \mathcal{T}$ are necessarily charged: some may be "fixed" during a choice of a pivot outside T. Additionally, note that if a triple $T \in \mathcal{T}$ is charged, the amount of charge is not necessarily $\lambda_1(T) - \lambda_2(T)$. For example, fix a triple $T = \{h, i, j\}$ with[3] $z(h,i) = \lambda_1(T) = 10$, $z(i,j) = \lambda_2(T) = 5$ and $z(j,h) = \lambda_3(T) = 1$. So $L(T) = 5$. Assume h, i, j are input to the same recursive call, and one of them is chosen as pivot. If i is chosen, then the value of $z(j,h)$ will be changed to 10, in which case the charge is 9. We will treat the first 4 units of charge (i.e. change $\lambda_3(T) \to \lambda_2(T)$) and the last 5 (i.e. the remaining climb up to $\lambda_1(T)$) separately (B-type charge and A-type charges, respectively). If h is chosen, then the value of $z(i,j)$ will change to 10, and the total charge will be 5 (only A-type charge). If j is chosen, then the value of $z(i,j)$ will change to 5, incurring an A-type cost of 5. One more event we must be aware of: If a vertex $k \notin T$ was chosen as pivot (when all of h, i, j, k were in the same recursive call), then the values on T might be mutated. The case is interesting only if at that moment $z(k,i) = z(k,j) = z(k,h) = l$, because otherwise the triple $T = (i,j,k)$ will be broken into at least two recursion branches and will not be charged. So assume this is the case. Then $\lambda_1(T)$ can decrease (and therefore $L(T)$ decreases). Note that $\lambda_2(T)$ can also decrease but in that case we will have $\lambda_1(T) = \lambda_2(T) = l$ (after the mutation), and T is no longer a violator.

By the above discussion, the sets $\mathcal{T}$ and $\mathcal{T}_b$ can decrease during the execution of the algorithm, and as long as $T \in \mathcal{T}$, its λ_3 and λ_2 values are fixed.

For integer $1 \le b < M + 1$ and $T \in \mathcal{T}$, let A_T^b denote the event that T was charged, and $T \in \mathcal{T}_b$ at that point. In other words, one of T's vertices was chosen as pivot when the other two were in the same recursive call, and also $\lambda_2(T) \le b < \lambda_1(T)$ at that point. This event captures an A-type charge. For $1 \le b < M$ let B_T^b denote the event that T was charged, $\lambda_3(T) \le b < \lambda_2(T)$, and the vertex not incident to the strictly lowest valued edge was the pivot (this captures a B-type charge). Let $p_T^b = \Pr[A_T^b]$ and $q_T^b = \Pr[B_T^b]$.

By Observation 1 and the above discussion, the total cost of the algorithm is therefore

$$\sum_{T\in\mathcal{T}} \sum_{b=1}^{M} \chi(A_T^b) + \sum_{T\in\mathcal{T}} \sum_{b=1}^{M-1} \chi(B_T^b),$$

where $\chi(\cdot)$ is the indicator variable for an event. The expected cost is

$$\sum_{T\in\mathcal{T}} \sum_{b=1}^{M} \Pr[A_T^b] + \sum_{T\in\mathcal{T}} \sum_{b=1}^{M-1} \Pr[B_T^b]$$
$$= \sum_{T\in\mathcal{T}} \sum_{b=1}^{M} p_T^b + \sum_{T\in\mathcal{T}} \sum_{b=1}^{M-1} q_T^b \ .$$

It is easy to see that for a given $e \in \binom{X}{2}$ and two distinct $T_1, T_2 \in \mathcal{T}$ such that $e \subseteq T_1, T_2$, the events "$A_{T_1}^b$ when $T_1 \setminus e$ is pivot" and "$A_{T_2}^b$ when $T_2 \setminus e$ is pivot" are mutually exclusive. Why? Because by Observation 1 we know that the value z_e can cross b only once. Conditioned on $A_{T_1}^b$, all three vertices of T_1 are equally likely to be the pivot (and the same for T_2). Therefore, summing up probabilities of pairwise disjoint events gives $\sum_{T\in\mathcal{T}:e\subseteq T} p_T^b/3 \le 1$ for all $e \in \binom{X}{2}$ and $1 \le b < M + 1$.

Using a similar argument, it is not hard to see that for a given $e \in \binom{X}{2}$ and $1 \le b < M$, the events $B_{T_1}^b$ and $B_{T_2}^b$ are mutually exclusive for distinct T_1, T_2 such that $e \subseteq T_1, T_2$. Therefore, $\sum_{T\in\mathcal{T}:e\subseteq T} q_T^b \le 1$ for all $e \in \binom{X}{2}, 1 \le b < M$ (we do not need to divide by 3 here because there is only one choice of pivot among the vertices of $T \supseteq e$ that causes B_T^b). Thus, $\sum_{T\in\mathcal{T}:e\subseteq T} \sum_{b=1}^{M-1} q_T^b/(M-1) \le 1$ for all $e \in \binom{X}{2}$. By setting $\delta_T^b = p_T^b/3$ we get a feasible solution to Dual LP 2, and therefore $\sum_{b=1}^{M} \sum_{T\in\mathcal{T}_b} p_T^b/3$ is a lower bound for c^* ($\mathcal{T}_b$ as defined *before* execution, as in the definition of Dual LP 2 - note that $p_T^b = 0$ for $T \notin \mathcal{T}_b$). Thus, $\sum_{b=1}^{M} \sum_{T\in\mathcal{T}} p_T^b$ is at most $3c^*$. Also, by setting $\beta_T = \sum_{b=1}^{M-1} q_T^b/(M-1)$ we get a feasible solution to Dual LP 1, and thus c^* is at least $\sum_{T\in\mathcal{T}} \beta_T L(T) \ge \sum_{T\in\mathcal{T}} \beta_T$ (note that $\beta_T = 0$ if T is not a violating triangle before execution) and we conclude that $\sum_{T\in\mathcal{T}} \sum_{b=1}^{M-1} q_T^b$ is at most $(M-1)c^*$. Therefore, the total expected approximation ratio is at most $3 + (M-1) = 2 + M$, as required.

3. A Linear Programming Approach

In this section, we describe our LP relaxation for the closest ultrametric. We first consider the closest ultrametric under the ℓ_1 norm and later generalize the ideas to general ℓ_p norms.

[3] The original definition of λ_s for $s = 1, 2, 3$ used the D-values, but their values are mutated with the z-values.

It will be useful to restrict the solution to only include ultrametrics with distances in the set $\{D(i,j)|i,j \in X\}$. By Lemma 1(a) from [12], this does not change the value of the optimal solution. Let $\{D_1 < D_2 < \cdots < D_M\}$ denote the sorted set of values $\{D(i,j)\}$ where $M = O(n^2)$. Let $L_t = D_t - D_{t-1}$, for $1 < t \leq M$ and $L_1 = D_1$. Then an ultrametric with distances in the set $\{D_t\}$ can be viewed as an M-level tree where the edges at level t have length L_t as in Figure 2. (We number levels in increasing order from the leaves to the root.)

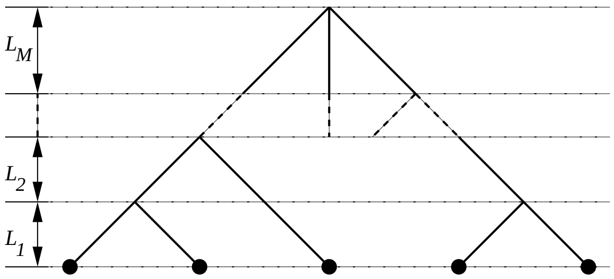

Figure 2. M-level tree with weighted levels.

Equivalently, such an ultrametric on X can be viewed as a hierarchy of M clusterings (i.e. partitions of X) with length L_t associated with the clustering at level t. Our LP relaxation for the closest ultrametric is based on this view. We model the clustering at level t by a $\{0,1\}$ distance function x_{ij}^t. We relax this to allow x_{ij}^t to be a $[0,1]$ distance function that satisfies triangle inequality. We impose the constraint that the clustering at level t is a refinement of that for $t+1$, by specifying that the distance function between two vertices i and j decreases as t increases. The constraints are summarized in Figure 3.

$$\begin{aligned} x_{ik}^t &\leq x_{ij}^t + x_{jk}^t \quad \forall t,i,j,k && (1)\\ x_{ij}^t &\geq x_{ij}^{t+1} \quad \forall t,i,j && (2)\\ 0 \leq x_{ij}^t &\leq 1 \quad \forall t,i,j && (3) \end{aligned}$$

Figure 3. The closest ultrametric LP constraints. The t's are integers from $\{1,\ldots,M\}$ and the i,j,k's are from X.

An integer solution $\hat{x}_{ij}^t$ to the LP is interpreted as an ultrametric U as follows: $U(i,j) = \sum_{t=1}^M L_t \hat{x}_{ij}^t$. For a fixed t, a $\{0,1\}$ solution $\hat{x}_{ij}^t$ corresponds to a clustering of X, because of the triangle inequality constraints (1). Further, the constraints (2) imply that the clustering at level t is a refinement of the clustering at level $t+1$. This leads to a weighted tree structure on X where all points in X are at the leaves of the tree and edges at level t of this tree have length L_t. Note that $\hat{x}_{ij}^t$ denotes whether i and j are separated at level t or not. The distance function $U(i,j) = \sum_{t=1}^M L_t \hat{x}_{ij}^t$ is (half) the shortest path distance in this weighted tree, and thus indeed an ultrametric. Modeling an ultrametric in this way as a hierarchy of clusters seems more useful than trying to work with an LP formulation that has variables corresponding to ultrametric distances directly.

In order to specify the LP objective function, we define constants D_{ij}^t where $D_{ij}^t = 1$ if $D(i,j) \geq D_t$ and $D_{ij}^t = 0$ if $D(i,j) < D_t$. The LP objective function is

$$\min \sum_{t=1}^{M} L_t (\sum_{ij:D_{ij}^t=0} x_{ij}^t + \sum_{ij:D_{ij}^t=1} (1 - x_{ij}^t)) \,. \qquad (4)$$

Note that $D(i,j) = \sum_{t=1}^M L_t D_{ij}^t$. Thus $|D(i,j) - U(i,j)| = \sum_{t=1}^M L_t |D_{ij}^t - \hat{x}_{ij}^t|$. This is the contribution of pair (i,j) to the objective function (see Figure 4).

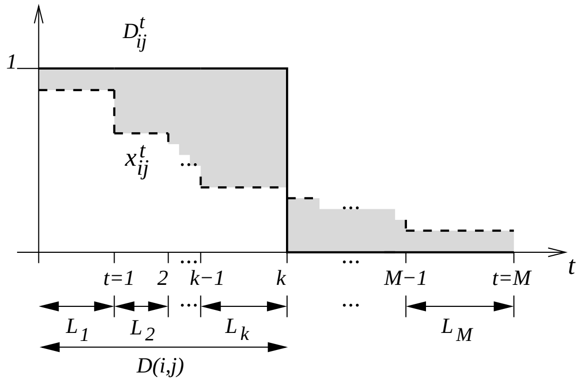

Figure 4. LP contribution from pair (i,j)

We note that the LP relaxation is quite flexible and easily allows incorporating additional constraints such as different weights on pairs (i,j), and allowed ranges for pairwise distances. This will be important later when we adapt the ideas for other ℓ_p norms and use the nearest ultrametric algorithm to solve the nearest tree metric problem.

3.1. Rounding algorithm for closest Ultrametric

In describing the algorithm, it is useful to keep in mind the equivalence between an ultrametric and a hierarchy of clusterings. Algorithm HIERARCHICAL-

CLUSTER (see Figure 5) constructs a hierarchical clustering in a top down fashion. One can also view the algorithm as constructing an integer solution by setting the LP variables to 0-1 in a recursive fashion. The algorithm constructs the clusterings at higher levels and then proceed to lower levels. At each stage, the algorithm works with a subset Z of points of the original point set, at a particular level $t \in \{1, M\}$. The algorithm partitions Z into clusters (possibly just one – leaving Z unchanged) and constructs a hierarchical clustering for each cluster starting at level $t-1$. It is intiated by calling HIERARCHICAL-CLUSTER(X, M). We begin with some definitions we will need in describing the algorithm and its analysis. We start by defining $\rho = LP/n$, where LP is the optimum value of the linear program. We also define the following variables:

Definition 1

$$\begin{aligned}
A^t_Z &= \rho|Z| + \sum_{s=1}^{t} L_s \sum_{ij \in Z : D^s_{ij}=0} x^s_{ij} \\
W^t_{ij} &= \sum_{s=1}^{t} L_s(1 - D^s_{ij}) \\
&= \max((\sum_{s=1}^{t} L_s) - D_{ij}, 0) \\
V^t_Z(c,r) &= \{i \in Z : x^t_{ci} \le r\} \\
\delta^t_Z(c,r) &= \{(i,j) | i \in V^t_Z(c,r), j \in Z \setminus V^t_Z(c,r)\} \\
W^t_Z(c,r) &= \sum_{(i,j) \in \delta^t_Z(c,r)} W^t_{ij} \\
A^t_Z(c,r) &= \rho|V^t_Z(c,r)| + \sum_{i,j \in V^t_Z(c,r)} \sum_{\substack{s \le t \\ D^s_{ij}=0}} L_s x^s_{ij} \\
&+ \sum_{\substack{i \in V^t_Z(c,r) \\ j \in Z \setminus V^t_Z(c,r)}} \sum_{\substack{s \le t \\ D^s_{ij}=0}} L_s(r - x^t_{ci})
\end{aligned}$$

The ρ-terms in the definition of the A-variables (A^t_Z and $A^t_Z(c,r)$) are used to simplify our analysis. Without the term, it is easy to see that the A-variables are simply portions of the LP cost. Our main argument will show how to charge the algorithm cost (captured by the W-variables) to the A-variables. By our definitions, and since $\sum_{s=1}^t L_s \sum_{ij \in Z : D^s_{ij}=0} x^s_{ij} \le LP$ and $|Z| \le n$, we have

Observation 2 *(i) $A^t_Z/\rho \le 2n$, (ii) $A^t_Z(c,r) \ge \rho$, and (iii) $A^M_X \le 2LP$,*

We now show that the algorithm outputs a valid solution.

```
HIERARCHICAL-CLUSTER($Z, t$)
(1) call CLUSTER-PARTITION($Z$,$t$)
        obtaining partition of $Z$:
        $Z = Z_1 \cup \ldots \cup Z_m$
(2) for $i, j \in Z_l$, $1 \le l \le m$
        set $\hat{x}^t_{ij} = 0$
(3) for $1 \le l < l' \le m, 1 \le s \le t, i \in Z_l, j \in Z_{l'}$
        set $\hat{x}^s_{ij} = 1$
(4) for $l = 1, \ldots, m$
        call HIERARCHICAL-CLUSTER($Z_l$,$t-1$)
(5) return

– – – – – – – – – – – – – – – – – – – – – – – – –
CLUSTER-PARTITION($Z, t$)
(1) set $m = 1$
(2) if $\forall i, j \in Z : (x^t_{ij} \le 2/3 \vee D^t_{ij} = 0)$ then
        set $Z_m = Z$
        return $Z_1, \ldots, Z_m$
(3) pick $ij \in Z$ s.t. $x^t_{ij} > 2/3$ & $D^t_{ij} = 0$
(4) if $A^t_Z(i, 1/3) \le A^t_Z/2$ then
        $c = i$
    else
        $c = j$
(5) pick $r \in [0, 1/3]$ s.t.
          $W^t_Z(c,r) \le O(\log\log n) A^t_Z(c,r)$ ×
                              $\ln(A^t_Z/A^t_Z(c,r))$
(6) set $Z_m = V^t_Z(c,r)$, $Z = Z \setminus Z_m$
(7) set $m = m + 1$
(8) goto (2)
```

Figure 5. Algorithm HIERARCHICAL-CLUSTER and procedure CLUSTER-PARTITION. Note that the precise expression of the $O(\log\log n)$ in line (5) of CLUSTER-PARTITION is $\ln\ln(A^t_z/\rho) - \ln\ln 2$.

Lemma 1 *The solution $\hat{x}^t_{ij}$ produced by Algorithm HIERARCHICAL-CLUSTER is a valid integer solution to the LP (1)-(3).*

Proof: It is easy to see that each $\hat{x}^t_{ij}$ is set exactly once in the algorithm. We show that the values of $\hat{x}^t_{ij}$ satisfy the constraints of the LP. Since the values are 0-1, the only possible violation of inequality (2) is $\hat{x}^{t+1}_{ij} = 1$ and $\hat{x}^t_{ij} = 0$. Step (3) ensures that this does not happen. Also, the only possible violation of inequality (1) is $\hat{x}^t_{ik} = 1, \hat{x}^t_{ij} = 0$ and $\hat{x}^t_{jk} = 0$. Consider the recursive call of the algorithm where $\hat{x}^t_{ik}$ was set to 1. Note that

$i \in Z_l, j \in Z_{l'}, l \neq l'$. $\hat{x}^t_{ij} = 0$ implies that $j \in Z_l$ and $\hat{x}^t_{jk} = 0$ implies that $j \in Z_{l'}$, giving a contradiction. □

Lemma 2 *In Step (4) of algorithm* CLUSTER-PARTITION, $A^t_Z(c, 1/3) \leq A^t_Z/2$.

Proof: We claim that $A^t_Z(i, 1/3) + A^t_Z(j, 1/3) \leq A^t_Z$. It is easy to see this, by verifying that for every pair $i, j \in Z$, the total contribution to the LHS is at most the contribution to the RHS. The choice of c in Step (4) now ensures that the lemma holds. □

Lemma 3 *In Step (5) of algorithm* CLUSTER-PARTITION, *there exists* $r \in [0, 1/3]$ *such that* $W^t_Z(c, r) \leq (\ln\ln(A^t_Z/\rho) - \ln\ln 2) A^t_Z(c, r) \ln(A^t_Z/A^t_Z(c, r))$.

Proof: Note that $A^t_Z(c, r)$ is a nondecreasing function of r. It has at most n linear pieces with possible discontinuities for values of $r = x^t_{ci}$ for $i \in Z$. Let $R = \{r : A^t_Z(c, r)$ is differentiable at $r\}$. For all $r \in R$,

$$\frac{dA^t_Z(c, r)}{dr} = W^t_Z(c, r)$$

Assume for contradiction, that the statement of the lemma is false. Then, for all $r \in [0, 1/3] \cap R$,

$$\frac{dA^t_Z(c, r)}{dr} > (\ln\ln(A^t_Z/\rho) - \ln\ln 2) \times A^t_Z(c, r) \ln(A^t_Z/A^t_Z(c, r)) \ .$$

Therefore,

$$\frac{1}{A^t_Z(c, r) \ln(A^t_Z/A^t_Z(c, r))} \frac{dA^t_Z(c, r)}{dr} > (\ln\ln(A^t_Z/\rho) - \ln\ln 2)$$

$$-\frac{d \ln\ln(A^t_Z/A^t_Z(c, r))}{dr} > (\ln\ln(A^t_Z/\rho) - \ln\ln 2) \ ,$$

giving by integration over $[0, 1/3]$,

$$\ln\ln(A^t_Z/A^t_Z(c, 0)) - \ln\ln(A^t_Z/A^t_Z(c, 1/3)) > (\ln\ln(A^t_Z/\rho) - \ln\ln 2) \ .$$

By Observation 2 (ii), $\ln\ln(A^t_Z/A^t_Z(c, 0)) \leq \ln\ln(A^t_Z/\rho)$. Also, $A^t_Z(c, 1/3) \leq A^t_Z/2$ (by the choice of c in Step (4)). Therefore, the maximum possible value of the LHS in the last inequality derived above is $\ln\ln(A^t_Z/\rho) - \ln\ln 2$. This gives a contradiction. □

Theorem 2 *The cost of the solution* $\hat{x}^t_{ij}$ *produced by the algorithm* HIERARCHICAL-CLUSTER *is within a factor of* $O(\log n \log\log n)$ *times the value of the LP solution.*

Proof: We will bound the cost of the solution produced by separately bounding the contribution of $\hat{x}^t_{ij}$ set to 0 and $\hat{x}^t_{ij}$ set to 1. Note that whenever we set $\hat{x}^t_{ij} = 0$, either $D^t_{ij} = 0$ or $x^t_{ij} \leq 2/3$. In the former case, the contribution of $\hat{x}^t_{ij}$ to the solution produced is 0. In the latter case, the contribution to the solution produced is at most 3 times the contribution of x^t_{ij} to the LP solution.

Note that if $\hat{x}^s_{ij}$ is set to 1 for $s \in \{1, \ldots, t\}$ in Step (3) of algorithm HIERARCHICAL-CLUSTER, w.l.o.g. $i \in Z_l, j \in Z_{l'}, l < l'$ in the partition of Z produced by procedure CLUSTER-PARTITION. The total contribution to the solution is W^t_{ij}. We bound this contribution by considering the condition ensured in Step (5) of CLUSTER-PARTITION when $i \in Z_l$ is separated from j:

$$W^t_Z(c, r) \leq O(\log\log n) A^t_Z(c, r) \ln(A^t_Z/A^t_Z(c, r)) \ .$$

Note that W^t_{ij} is included in the $W^t_Z(c, r)$ term in the LHS, by definition. This inequality ensures that the cost $W^t_Z(c, r)$ can be charged to $A^t_Z(c, r)$, which is a portion of A^M_X. In fact, each unit of $A^t_Z(c, r)$ is charged $O(\log\log n) \ln(A^t_Z/A^t_Z(c, r))$ times.

Now we show that every piece of A^M_X is charged to an extent of at most $O(\log n \log\log n)$. In virtue of Observation 2 (iii), this would imply the statement of the theorem. Consider a pair i, j and s with $D^s_{ij} = 0$. The contribution to A^M_X is x^s_{ij} (multiplied by L_s). Each time this piece of A^M_X is charged, i, j are in some set Z_l that is partitioned at some level t_l by growing a ball of radius r_l around center c_l. The charge to this piece of the LP is $O(\log\log n) \ln(A^{t_l}_{Z_l}/A^{t_l}_{Z_l}(c_l, r_l))$. Either i and j are separated by this partitioning, or i and j are in the same partition as c_l, in which case the LP contribution could be charged further. Suppose that there are a total of q such charges, for $l = 1 \ldots q$. The total charge to this piece of the LP solution is at most

$$O(\log\log n) \sum_{l=1}^{q} \ln(A^{t_l}_{Z_l}/A^{t_l}_{Z_l}(c_l, r_l)) = O(\log\log n) \sum_{l=1}^{q} (\ln A^{t_l}_{Z_l} - \ln A^{t_l}_{Z_l}(c_l, r_l)) \ .$$

Note that $A^{t_{l+1}}_{Z_{l+1}} \leq A^{t_l}_{Z_l}(c_l, r_l)$ (since the LHS includes fractional contributions of edges cut by Z_{l+1} and the algorithm might skip a few levels before partitioning again). Therefore, the total charge to a piece of LP is at most

$$O(\log\log n)(\ln A^{q_1}_{Z_1} - \ln A^{t_q}_{Z_q}(c_q, r_q)) \ .$$

By Observation 2 (i) and (ii), the last expression is at most $O(\log n \log\log n)$, as required. □

Corollary 3 *Algorithm* HIERARCHICAL-CLUSTER *yields an $O(\log n \log\log n)$ approximation for the problem of finding the closest ultrametric under the ℓ_1 norm.*

We remark that our LP based method can be modified to incorporate additional constraints on the pairwise distances. In particular, we can upper and lower bound certain pairwise distances by constants and hence also specify that certain distances should be equal to constants. This is done by constraining the appropriate x^t_{ij} variables in the LP to be either 0 or 1. Note that the rounding procedure has the property that $x^t_{ij} = 0 \Rightarrow \hat{x}^t_{ij} = 0$ and $x^t_{ij} = 1 \Rightarrow \hat{x}^t_{ij} = 1$.

3.2. Other Objective Functions

The LP rounding algorithm HIERARCHICAL-CLUSTER is analyzed in Section 3.1 for the problem of minimizing the ℓ_1 distance of an ultrametric to a target distance function D. We show in this section that the techniques can be extended to any ℓ_p norm for $p > 1$.

More precisely, consider the problem of minimizing $||d - D||_p = (\sum_{ij}(d(i,j) - D(i,j))^p)^{1/p}$ over all ultrametrics d. As in the ℓ_1 case, we will restrict the solution to include only ultrametrics d with values in the set $\{D(i,j)|i,j \in X\}$. By Lemma 1(b) from [12], this increases the value of the optimal solution by a multiplicative factor of at most 2. In fact, we will consider the objective function $||d - D||_p^p$. For this objective, the optimal value is increased by a factor of at most 2^p.

Our LP relaxation uses the same variables and constraints as before (in Figure 3) with a different objective function. Recall that $\{D_1 < D_2 < \cdots < D_M\}$ are the sorted set of values $\{D(i,j)\}$. In addition, define $D_0 = 0$. In order to write down the objective function, we define constants $\mathcal{Q}^t_{ij}$ and $\mathcal{R}^t_{ij}$ as follows (see Figure 6): $\mathcal{Q}^t_{ij} = \max(|D_t - D(i,j)|^p - |D_{t-1} - D(i,j)|^p, 0)$, and $\mathcal{R}^t_{ij} = \max(|D_{t-1} - D(i,j)|^p - |D_t - D(i,j)|^p, 0)$. We now write the LP objective function for minimizing $||d - D||_p^p$ as follows:

$$\min \sum_{i,j\in X} \sum_{t=1}^{M} \left(x^t_{ij}\mathcal{Q}^t_{ij} + (1 - x^t_{ij})\mathcal{R}^t_{ij} \right) \ , \qquad (5)$$

Note that the sums in the definition of $\mathcal{Q}^t_{ij}$and $\mathcal{R}^t_{ij}$ are telescoping. It is easy to verify that for an integer feasible solution $x^t_{ij} \in \{0,1\}$ to the closest metric LP, the objective function (5) is exactly $\sum_{ij}(d_x(i,j) - D(i,j))^p$, where d_x is the ultrametric induced by the feasible solution x (that is, $d_x(i,j) = D_t$ for the maximal t such that $x^t_{ij} = 1$). Also it is not hard to change the A and W-variables in Definition 1 (and hence also CLUSTER-PARTITION) in a suitable way so that the analysis leading to Corollary 3 will lead to

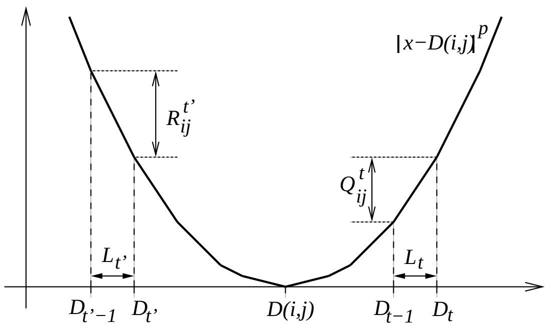

Figure 6. Illustration of Q^t_{ij} and $R^{t'}_{ij}$.

Corollary 4 *Algorithm* HIERARCHICAL-CLUSTER *yields an $O(2^p \log n \log\log n)$ approximation for the problem of finding the ultrametric d minimizing $||d - D||_p^p$ and hence an $O((\log n \log\log n)^{1/p})$ approximation for the problem of minimizing $||d - D||_p$.*

We omit the required technical changes from this abstract.

3.3. Closest tree metric

Agarwala etal [1] show that the closest tree metric problem can be solved via a reduction to the closest a-restricted ultrametric problem ($a \in X$). This is a variant of the closest ultrametric problem, which restricts the space of permissible ultrametrics to those that satisfy some additional distance properties expressed in terms of distances from point $a \in X$. We first introduce some notation to explain the additional restrictions needed. Consider $a \in X$. Let $m_a = \max_i\{D(a,i)\}$. Let $l_i = m_a - D(a,i)$. Consider the *centroid* metric C^a where $C^a(i,j) = l_i + l_j = 2m_a - D(a,i) - D(a,j)$. Now consider the distance function $D + C^a$. Note that the distance from a to any point i in this new distance function is $2m_a$. An ultrametric U is said to be a-restricted (with respect to distance function D) if it satisfies the following constraints:

$$\begin{array}{lrcl} \text{for all } i,j & 2\max(l_i,l_j) \le U(i,j) & \le & 2m_a \\ & U(a,i) & = & 2m_a \end{array}$$

The variant of the closest ultrametric problem we need to solve is the following: Given distance function D, $a \in$

X, find an a-restricted ultrametric U so as to minimize $||U - (D + C^a)||_p$.

Note that the additional constraints imposed on the ultrametric are simply upper and lower bounds for certain pairs of distances and equality constraints for certain pairs of distances. In fact our LP based method for the nearest ultrametric can be easily modified to give an $O(\log n \log \log n)$ approximation for this variant. (We need to include the values $\{l_i\}$ in the set of allowed distances in writing down our LP relaxation.) Combined with the reduction of Agarwala etal [1], this implies the following result.

Theorem 5 *There is a polynomial time algorithm to obtain an $O((\log n \log \log n)^{1/p})$ approximation for the problem of finding the closest tree metric under the ℓ_p norm.*

4. Conclusion

It would be interesting to obtain a combinatorial $poly \log(n)$ approximation for the closest ultrametric/tree metric problem we consider. Determining whether an $O(1)$ approximation can be obtained is a fascinating question. The LP formulation used in our work could eventually lead to such a result. It would be interesting to look at the problem of aggregating a given set of hierarchical clusterings into a single representative one. A natural formulation of such a question is to ask for a single hierarchical clustering that minimizes the sum of distances from a given set of hierarchical clusterings, akin to the formulation of aggregation problems in other settings [7, 10, 11]. A 2-approximation for this problem is trivial (by picking the best of the given clusterings). Going beyond factor 2 requires some nontrivial combining of clusterings. For aggregating clusterings (i.e. the $M = 1$ case), Ailon etal [2] recently obtained better algorithms and it is natural to ask whether such results can be obtained for hierarchical clusterings as well. The LP formulation we use seems to be a valuable tool for representing distributions on trees and may have applications to other problems involving tree metrics.

References

[1] R. Agarwala, V. Bafna, M. Farach, M. Paterson, and M. Thorup. On the approximability of numerical taxonomy (fitting distances by tree metrics). *SIAM Journal on Computing*, 28(3):1073–1085, 1999.

[2] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: Ranking and clustering. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, 2005.

[3] F. Ardila. Subdominant matroid ultrametrics. *Annals of Combinatorics*, 8(4):379–389, 2005.

[4] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning Journal (Special Issue on Theoretical Advances in Data Clustering)*, 56(1–3):89–113, 2004. Extended abstract appeared in FOCS 2002, pages 238–247.

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

[6] K. Dhamdhere. Approximating additive distortion of embeddings into line metrics. In *Proceedings of 7th APPROX and 8th RANDOM*, volume 3122. Springer-Verlag, 2004.

[7] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the Tenth International Conference on the World Wide Web (WWW10)*, pages 613–622, Hong Kong, 2001.

[8] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC)*, pages 448–455, 2005.

[9] M. Farach, S. Kannan, and T. Warnow. A robust model for finding optimal evolutionary trees. *Algorithmica, Special Issue on Computational Biology*, pages 155–179, 1995.

[10] V. Filkov and S. Skiena. Integrating microarray data by consensus clustering. In *Proceedings of International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 418–425, Sacramento, 2003.

[11] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. In *Proceedings of the 21st International Conference on Data Engineering (ICDE)*, Tokyo, 2005. To appear.

[12] B. Harb, S. Kannan, and A. McGregor. Approximating the best-fit tree under l_p norms. In *Proceedings of 8th APPROX and 8th RANDOM*. Springer-Verlag, 2005.

[13] B. Holland, K. Huber, J. Koolen, V. Moulton, and J. Weyer-Menkhoff. Delta-additive and delta-ultra-additive maps, gromov's trees, and the farris transform. *Discrete Applied Mathematics*, pages 51–73, 2005.

[14] J. Kim and T. Warnow. Tutorial on phylogenetic tree estimation, 2004. Originally presented at ISMB 1999.

[15] B. Ma, L. Wang, and L. Zhang. Fitting distances by tree metrics with increment error. *Journal of Combinatorial Optimization*, 3(2-3):213–225, 1999.

[16] E. Mossel and S. Roch. Learning nonsingular phylogenies and hidden markov models. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, 2005.

[17] H. T. Wareham. On the complexity of inferring evolutionary trees. Technical Report Technical Report 9301, Memorial University of New Foundland, 1993.