Approximating the Best–Fit Tree Under L_p Norms

Boulos Harb*, Sampath Kannan**, and Andrew McGregor***

Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104, USA
{boulos,andrewm,kannan}@cis.upenn.edu

Abstract. We consider the problem of fitting an $n \times n$ distance matrix M by a tree metric T. We give a factor $O(\min\{n^{1/p}, (k \log n)^{1/p}\})$ approximation algorithm for finding the closest ultrametric T under the L_p norm, i.e. T minimizes $\|T, M\|_p$. Here, k is the number of distinct distances in M. Combined with the results of [1], our algorithms imply the same factor approximation for finding the closest *tree metric* under the same norm. In [1], Agarwala *et al.* present the first approximation algorithm for this problem under L_∞. Ma *et al.* [2] present approximation algorithms under the L_p norm when the original distances are not allowed to contract and the output is an ultrametric. This paper presents the first algorithms with performance guarantees under L_p ($p < \infty$) in the general setting.
We also consider the problem of finding an ultrametric T that minimizes L_{relative}: the sum of the factors by which each input distance is stretched. For the latter problem, we give a factor $O(\log^2 n)$ approximation.

1 Introduction

An *evolutionary tree* for a species set $\mathcal{S}$ is a rooted tree in which the leaves represent the species in $\mathcal{S}$, and the internal nodes represent ancestors. The goal of reconstructing the evolutionary tree is of fundamental scientific importance. Given the increasing availability of molecular sequence data for a diverse set of organisms and our understanding of evolution as a stochastic process, the natural formulation of the tree reconstruction problem is as a maximum likelihood problem – estimate parameters of the evolutionary process that are most likely to have generated the observed sequence data. Here, the parameters include not only rates of mutation on each branch of the tree, but also the topology of the tree itself. It is assumed (although this assumption is not always easy to meet) that the sequences observed at the leaves have been multiply aligned so that each position in a sequence has corresponding positions in the other sequences. It is

* This work was supported by NIH Training Grant T32HG00 46.
** This work was supported by NSF CCR98-20885 and NSF CCR01-05337.
*** This work was supported by NSF ITR 0205456.

also assumed for tractability, that each position evolves according to an independent identically distributed process. Even with these assumptions, estimating the most likely tree is a computationally difficult problem.

Recently, *approximately* most likely trees have been found for simple stochastic processes using *distance-based methods* as subroutines [3, 4].

For a distance-based method the input is an $n \times n$ distance matrix M where $M[i, j]$ is the observed distance between species i and j. Given such a matrix, the objective is to find an edge-weighted tree T with leaves labeled 1 through n which minimizes the L_p distance from M where various choices of p correspond to various norms. The tree T is said to *fit* M. When it is possible to define T so that $\|T, M\|_p = 0$, then the distance matrix is said to be *additive*. An $O(n^2)$ time algorithm for reconstructing trees from additive distances was given by Waterman *et al.* [5], who proved in addition that at most one tree can exist. However, real data is rarely additive and we need to solve the norm minimization problem above to find the best tree. Day [6] showed that the problem is NP-hard for $p = 1, 2$.

For the case of $p = \infty$, referred to as the L_∞ norm, [7] showed how the optimal ultrametric tree could be found efficiently and [1] showed how this could be used to find a tree T (not necessarily ultrametric) such that $\|T, M\|_p \leq 3\|T_{\mathrm{OPT}}, M\|_p$ where T_{OPT} is the optimal tree. The algorithm of [1] is the one that is used in [3] and [4] for approximate maximum likelihood reconstruction.

In this paper we explore approximation algorithms under other norms such as L_1 and L_2. We also consider a variant, L_{relative}, of the best-fit objective mentioned above where we seek to minimize the sum of the factors by which each input distance is stretched. The study of L_1 and L_2 norms is motivated by the fact that these are often better measures of fit than L_∞ and the idea that using these methods as subroutines may yield better maximum likelihood algorithms.

1.1 Our Results

We prove the following results:

- We can find an ultrametric tree whose L_p-error is within a factor of $O(\min\{n^{1/p}, (k \log n)^{1/p}\})$ of the optimum, where k is the number of distinct distances in the input matrix.
- We can find an ultrametric tree T whose L_{relative}-error is within a factor of $O(\log^2 n)$ of the optimum.

Our algorithms also solve the problem of finding *non-contracting* ultrametrics, i.e. when $T[i, j]$ is required to be at least $M[i, j]$ for all i, j. More generally, we can require that each output distance is lower bounded by some arbitrary positive value. This generalization allows us to also find *additive* metrics whose L_p-error is within a factor of $O(\min\{n^{1/p}, (k \log n)^{1/p}\})$ of the optimum by appealing to work in [1].

1.2 Related Work

Aside from the aforementioned L_∞ result given in [1], Ma *et al.* [2] present an $O(n^{1/p})$ approximation algorithm for finding non-contracting ultrametrics under $L_{p<\infty}$. Prior to our results, however, no algorithms with provable approximation guarantees existed for fitting distances by additive metrics under $L_{p<\infty}$ in the general setting.

Some of our results rely on the recent approximation algorithms for the problem of correlation clustering and related problems [8–11]. One of our algorithms can be viewed as performing a hierarchical version of correlation clustering.

Finally, we should mention some recent work that address special cases of our problem. In [12] an algorithm is given that finds a line-embedding of a metric whose L_1-error is $O(\log n)$ away from optimal. If the embedding is further restricted to be a non-contracting line-embedding, then [13] presents an algorithm whose approximation factor is constant.

2 Preliminaries

An *ultrametric* T on a set $[n]$ is a metric that satisfies the following *three-point condition*:

$$\forall x, y, z \in [n] \quad T[x, y] \leq \max\{T[x, z], T[z, y]\} \ .$$

That is, in an ultrametric, triangles are isosceles with the equal sides being longest. An ultrametric is a special kind of tree metric where the distance from the root to all points in $[n]$ (the leaves) is the same. Recall that a *tree metric* (equivalently an *additive* metric) A on $[n]$ is a metric that satisfies the *four-point condition*:

$$\forall w, x, y, z \in [n] \quad A[w, x] + A[y, z] \leq \max\{A[w, y] + A[x, z], A[w, z] + A[x, y]\} \ .$$

Given an $n \times n$ distance matrix M where $M[i, j]$ is the observed distance between objects i and j, our initial objective is to find an edge-weighted ultrametric T with leaves labeled 1 through n which minimizes the L_p distance from M, i.e. T minimizes

$$\|T, M\|_p = \sqrt[p]{\sum_{i,j} |T[i, j] - M[i, j]|^p} \ . \tag{1}$$

We will also look at finding an edge-weighted ultrametric T which minimizes the average stretch of the distances in M, i.e. T minimizes

$$\|T, M\|_{\text{relative}} = \sum_{i,j} \max\left\{\frac{T[i, j]}{M[i, j]}, \frac{M[i, j]}{T[i, j]}\right\} \tag{2}$$

The entry $T[i, j]$ is the distance between the leaves i and j, which is defined to be the sum of the edge weights on the path between i and j in T. We will also

refer to the *splitting distance* of an internal node v of T as the distance between two leaves whose least common ancestor is v. Because T is an ultrametric, the splitting distance of v is simply twice the height of v.

We will assume that the input distances in M are non-negative integers such that

- $M[x, y] = M[y, x]$; and,
- $M[x, y] = 0 \iff x = y$.

That is, we will not assume that the distances in M satisfy the triangle inequality. We denote the *distinct* distances in M by,

$$d_k > d_{k-1} > ... > d_2 > d_1 \ .$$

Relationship to Correlation Clustering. The problem of finding an optimal ultrametric T minimizing $\|T, M\|_1$ is closely related to the problem of correlation clustering introduced in [10]. We are interested in the minimization version of correlation clustering which is defined as follows: given a graph G whose edges are labeled "+" (similar) or "–" (dissimilar), cluster the vertices so as to minimize the number of pairs incorrectly classified with respect to the input labeling. That is, minimize the number of "–" edges within clusters plus the number of "+" edges between clusters. We will simply refer to this problem as *correlation clustering*. Note that the number of clusters is not specified in the input.

In fact, when G is complete, correlation clustering is equivalent to the problem of finding an optimal ultrametric under the L_1 norm when the input distances in M are restricted to 1 and 2. An edge (i, j) in the graph labeled "+" (resp. "–") is equivalent to the entry $M[i, j]$ being 1 (resp. 2). It is clear that an optimal ultrametric is an optimal clustering, and vice versa. Hence, the APX-hardness of finding an optimal ultrametric under the L_1 norm follows directly from [11, Theorem 11].

In [11], Charikar, Guruswami and Wirth give a factor $O(\log n)$ approximation to correlation clustering on general *weighted* graphs using linear programming. In an instance of correlation clustering that is weighted, each edge e has a weight w_e which can be either positive or negative. The objective is then to minimize

$$\sum_{e:w_e>0} (|w_e| \text{ if } e \text{ is split}) + \sum_{e:w_e<0} (|w_e| \text{ if } e \text{ is not split}) \ .$$

The bound for the LP relaxation is established via an application of the region growing procedure of Garg, Vazirani and Yannakakis [14]. We will state their theorem below for reference as our algorithm in section 3.1 uses their algorithm as a sub-procedure.

Theorem 1 ([11, Theorem 1]). *There is a polynomial time algorithm that achieves an $O(\log n)$ approximation for correlation clustering on general weighted graphs.*

3 Main Results

Both our algorithms take as input a set of splitting distances we call S that depends on the error norm. The distances in the constructed ultrametrics will be a subset of the given set S. The following lemma quantifies the affect of restricting the output distances to certain sets.

Lemma 1. *(a) There exists an ultrametric T with $T[i,j] \in \{d_1, d_2, \ldots, d_k\}$ for all i, j that is optimal under the L_1 norm.*

(b) There exists an ultrametric T with $T[i,j] \in \{d_1, d_2, \ldots, d_k\}$ for all i, j such that

$$\|T, M\|_p \leq 2\|T_{\mathrm{OPT}}, M\|_p \ ,$$

for $p \geq 2$.

(c) Assuming $d_k = O(poly(n))$, there exists an ultrametric T that uses $O(\log_{1+\epsilon} n)$ distances such that

$$\|T, M\|_{\text{relative}} \ \leq \ (1+\epsilon)\|T_{\mathrm{OPT}}, M\|_{\text{relative}} \ ,$$

where $\epsilon > 0$.

Proof. (a) Say an internal node v is *undesirable* if its distance $h(v)$ to any of its leaves satisfies $2h(v) \notin \{d_1, d_2, \ldots, d_k\}$. Suppose T_{OPT} is an optimal ultrametric with undesirable nodes. We will modify T_{OPT} so that it has one less undesirable node. Let v be the lowest undesirable node in T_{OPT} and let $d = 2h(v) \in (d_\ell, d_{\ell+1})$ for some $1 \leq \ell \leq k-1$. Define the following two multisets:

$D_\ell = \{M[a,b] : a, b$ are in different subtrees of v and $M[a,b] \leq d_\ell\}$,

$D_{\ell+1} = \{M[a,b] : a, b$ are in different subtrees of v and $M[a,b] \geq d_{\ell+1}\}$.

Then the contribution of the distances in $D_\ell \cup D_{\ell+1}$ to $\|T_{\mathrm{OPT}}, M\|_1$ is

$$\sum_{\alpha \in D_\ell} (d - \alpha) + \sum_{\beta \in D_{\ell+1}} (\beta - d) \ .$$

The expression above is linear in d. If its slope ≥ 0 then set $h(v) = d_\ell/2$, and if the slope < 0 then set $h(v) = \min\{d_{\ell+1}/2, h(v')\}$ where v' is the parent of v. Such a change can only improve the cost of the tree.

(b) For $p \geq 2$, let T_{OPT} be an optimal ultrametric with undesirable nodes. We will transform T_{OPT} to an ultrametric T with no undesirable nodes such that $\|T, M\|_p \leq 2\|T_{\mathrm{OPT}}, M\|_p$. Let,

$$\|T_{\mathrm{OPT}}, M\|_p^p \ = \ \sum_u g_u(2h(u)) \ ,$$

where the sum is over the internal nodes of T_{OPT} and $g_u(x)$ is the cost of setting the splitting distance of node u to x. Again, let v be the lowest undesirable node and define D_ℓ and $D_{\ell+1}$ as above. Fix $d = 2h(v) \in (d_\ell, d_{\ell+1})$. We claim that $\min\{g_v(d_\ell), g_v(d_{\ell+1})\} \leq 2^p g_v(d)$.

If $d \leq (d_\ell + d_{\ell+1})/2$, then we can set $h(v) = d_\ell/2$ since for all $\alpha \in D_\ell$, $d_\ell - \alpha \leq d - \alpha$ and for all $\beta \in D_{\ell+1}$, $\beta - d_\ell \leq 2(\beta - d)$. Otherwise, we set $h(v) = d_{\ell+1}/2$. We are assuming *w.l.o.g.* that v has no parent in the region $(d_\ell, d_{\ell+1})$ since if such a parent v' exists, $h(v')$ will also be set to $d_{\ell+1}/2$.

(c) Let $D(T_{\mathrm{OPT}})$ be the set of distances in an optimal ultrametric that minimizes $\|T, M\|_{\mathrm{relative}}$. Group the distances in $D(T_{\mathrm{OPT}})$ geometrically, i.e. for some $\epsilon > 0$, group the distances into the following buckets:

$$[1, 1+\epsilon], \ \left(1+\epsilon, (1+\epsilon)^2\right], \ldots, \ \left((1+\epsilon)^{s-1}, (1+\epsilon)^s\right] \ .$$

Let t be the largest distance in $D(T_{\mathrm{OPT}})$. Clearly, $t \leq d_k = O(\mathrm{poly}(n))$. Hence, the number of buckets $s = \log_{1+\epsilon} t = O(\log_{1+\epsilon} n)$. Now consider an ultrametric T' that sets $T'[i,j] = (1+\epsilon)^\ell$ if the optimal $T[i,j] \in ((1+\epsilon)^{\ell-1}, (1+\epsilon)^\ell]$.

$$\begin{aligned}
\|T', M\|_{\mathrm{relative}} &= \sum_{i,j} \max\left\{\frac{T'[i,j]}{M[i,j]}, \frac{M[i,j]}{T'[i,j]}\right\} \\
&\leq \sum_{i,j} \max\left\{\frac{(1+\epsilon)T[i,j]}{M[i,j]}, \frac{M[i,j]}{T[i,j]}\right\} \\
&\leq (1+\epsilon)\|T_{\mathrm{OPT}}, M\|_{\mathrm{relative}} \ .
\end{aligned}$$

For ease of notation, we adopt the following conventions. Let $G = (V, E)$ be the graph representing M in the natural way. For an edge $e = (i, j)$ denote its input distance $M[i, j]$ by m_e and its output distance $T[i, j]$ by t_e. As described in section 2, w_e will code for the label and the weight $|w_e|$ on the edge passed to the correlation clustering algorithm. The lower bound on e, λ_e, is the minimum value e can contract, i.e. $t_e \geq \lambda_e$.

Supplying our algorithm with an edge lower bounds matrix Λ allows us, for example, to solve non-contracting versions of the objective functions we seek to minimize where for all e, $t_e \geq m_e$ by simply setting $\Lambda = M$. We will also use these lower bounds in section 4 when constructing general additive metrics under L_p norms.

In the following two subsections we present algorithms for our problem. The first algorithm is suitable if the number of distinct distances, k, in M is small. Otherwise, the second algorithm is more suitable.

3.1 Algorithm 1

Our algorithm takes as input a set of *splitting distances* S. Each distance in the constructed tree will belong to this set. Let $|S| = \kappa$ and number the splitting distances in ascending order $s_1 < s_2 < \ldots < s_\kappa$. The algorithm considers the splitting distances in descending order, and when considering s_l it may set some distances $T[i, j] = s_l$. If a distance of the tree is not set at this point, it will later be set to $\leq s_{l-1}$. The decision of which distances to set to s_l and which distances to set to $\leq s_{l-1}$ will be made using correlation clustering. See Fig. 1 for the description of the algorithm.

Algorithm *Correlation-Clustering-Splitting*(G, S, Λ)
($*$ Uses correlation clustering to decide how to split $*$)
1. Let all edges be "unset"
2. **for** $l = \kappa$ to 1:
3. **do** Do correlation clustering on the graph induced by the unset edges with weights:
-If $m_e \geq s_l$ and $\lambda_e < s_l$ then,
$w_e = -(f(m_e, s_{l-1}) - f(m_e, s_l))$
-If $\lambda_e = s_l$ then $w_e = -\infty$
-If $m_e = s_i < s_l$ then $w_e = f(s_i, s_l)$
4. **for** For each unset edge e split between different clusters:
5. **do** $t_e \leftarrow s_l$ and mark e as "set"

Fig. 1. Algorithm 1 (The function f is defined in Thm. 2)

Theorem 2. *Algorithm 1 can be used to find an ultrametric T such that any one of the following holds:*

1. $\|T, M\|_p \leq O((k \log n)^{1/p}) \|T_{\mathrm{OPT}}, M\|_p$ *if* $S = \{d_1, \ldots, d_k\}$ *and* $f(m_e, t_e) = |m_e - t_e|^p$.
2. $\|T, M\|_{\mathrm{relative}} \leq O(\log^2 n) \|T_{\mathrm{OPT}}, M\|_{\mathrm{relative}}$ *if* $S = \{(1+\epsilon)^i : 0 \leq i \leq \log_{1+\epsilon} d_k\}$ *and* $f(m_e, t_e) = \max\{\frac{t_e}{m_e}, \frac{m_e}{t_e}\}$.

Proof. Our algorithm produces an ultrametric T where the splitting distance of each node is restricted to be from the set S, i.e. $t_e \in S$ for all e. The proof below shows that the algorithm gives a $O(|S| \log n)$–approximation to $\sum_e f(m_e, t'_e)$ where T' is the optimal ultrametric satisfying $t'_e \in S$ for all e. The results in the theorem will then follow by appealing to Lemma 1.

Consider the correlation clustering instance performed in iteration l of the algorithm. Let $\mathrm{cost}_{\mathrm{OPT}}(l)$ be the optimal value for this instance and let $\mathrm{cost}(l)$ be the cost of our solution.

Claim 1: $\sum_{1 \leq l \leq \kappa} \mathrm{cost}(l) = \sum_e f(m_e, t_e)$.

Consider each edge e in turn. Let $t_e = s_l$. If $s_l > m_e$, then in the lth iteration we pay $f(m_e, s_l)$ for this edge. If $s_l < m_e = s_{l'}$, then in each iteration i, $l' \geq i > l$, we pay $f(s_{l'}, s_{i-1}) - f(s_{l'}, s_i)$; hence, in total we pay $f(s_{l'}, s_l) = f(m_e, t_e)$.

Claim 2: $\mathrm{cost}_{\mathrm{OPT}}(l) \leq \sum_e f(m_e, t'_e)$

Consider the following solution to the correlation clustering problem at iteration l induced by T': for all unset edges e if $t'_e \geq s_l$ we split e and if $t'_e < s_l$ we don't split e. We claim that the cost of this solution for the correlation clustering problem is less that $\sum_e f(m_e, t'_e)$. Consider each edge e in turn.

– $t'_e < s_l$ and $m_e < s_l$: Not splitting this edge contributes nothing to the correlation clustering objective.

– $t'_e \geq s_l$ and $m_e < s_l$: Splitting this edge contributes $f(s_l, m_e)$ to the correlation clustering objective but contributes $f(t'_e, m_e) \geq f(s_l, m_e)$ to $\sum_e f(m_e, t'_e)$.
– $t'_e < s_l$ and $m_e \geq s_l$: Not splitting this edge contributes $f(m_e, s_{l-1}) - f(m_e, s_l)$ to the correlation clustering objective but contributes $f(m_e, t'_e) \geq f(m_e, s_{l-1}) \geq f(m_e, s_{l-1}) - f(m_e, s_l)$ to $\sum_e f(m_e, t'_e)$.
– $t'_e \geq s_l$ and $m_e \geq s_l$: Splitting this edge contributes nothing to the correlation clustering objective.

Summing over all edges, the contributions to both objective functions gives the second claim.

Combining the above claims with Thm. 1, the tree we construct has the following property,

$$\sum_e f(t_e, m_e) = \sum_{1 \leq l \leq \kappa} \text{cost}(l) \;\leq\; O(\kappa \log n) \sum_e f(m_e, t'_e) \;\; .$$

The theorem follows.

3.2 Algorithm 2

Our second algorithm also takes as input a set of *splitting distances* S and, as before, each distance in the constructed tree belongs to this set. However while the approximation guarantee of the first algorithm depended on $|S|$, the approximation guarantee of the second algorithm depends only on n. At each step the first algorithm decided whether or not to place internal nodes at height s_l, and, if it did, how to partition the nodes below. In our second algorithm, at each step we instead decide the height at which we should place the next internal node and its partition. See Fig. 2 for the description of the algorithm. The first call to the algorithm sets $s_{l^*} = s_\kappa$.

Theorem 3. *Algorithm 2 can be used to find an ultrametric T such that any one of the following holds:*

1. $\|T, M\|_1 \leq n\|T_{\text{OPT}}, M\|_1$ *if* $S = \{d_1, \ldots, d_k\}$.
2. *For* $p \geq 2$, $\|T, M\|_p \leq 2n^{1/p}\|T_{\text{OPT}}, M\|_p$ *if* $S = \{d_1, \ldots, d_k\}$.

Proof. Our algorithm produces a ultrametric T where the splitting distance of each node is restricted to be from the set S, i.e. $t_e \in S$ for all e. The proof below shows that the algorithm gives an n–approximation to $\|T', M\|_p^p$ where T' is the optimal ultrametric satisfying $t'_e \in S$ for all e. The results in the theorem will then follow by appealing to Lemma 1.

Claim 1: The sum of Min-Split-Cost over all recursive calls of *Min-Cut-Splitting* equals $\|T, M\|_p^p$.

Consider an edge $e = (i, j)$ and let v be the lowest common ancestor of i and j in T. If $m_e \leq t_e$ then we paid $(t_e - m_e)^p$ for this edge in the Cut-Cost when splitting at v. If $m_e > t_e$, consider the internal nodes on the path from root to

Algorithm *Min-Cut-Splitting*(G, S, s_{l^*}, Λ)
($*$ Uses min cuts to work out splits $*$)
1. $l \leftarrow l^*+1$
2. Min-Split-Cost$\leftarrow \infty$
3. **repeat**
4. $l \leftarrow l-1$
5. Push-Down-Cost $\leftarrow \sum_e (\max\{0, m_e - s_l\})^p - (\max\{0, m_e - s_{l^*}\})^p$
6. **if** there exists an edge $e=(s,t)$ such that $\lambda_e = s_l$
7. **then** Find min-(s,t) cut C in G with edge weights
$w_e = (\max\{0, s_l - m_e\})^p$
8. **else** Find min-cut C in G with edge weights
$w_e = (\max\{0, s_l - m_e\})^p$
9. Cut-Cost $\leftarrow$ the cost of the cut
10. **if** Cut-Cost+Push-Down-Cost $\leq$ Min-Split-Cost
11. **then** Best-Cut$\leftarrow C$
12. Best-Splitting-Point$\leftarrow s_l$
13. Min-Split-Cost $\leftarrow$ Cut-Cost+Push-Down-Cost
14. **until** $l=0$ or there exists an edge e with $\lambda_e = s_l$
15. **for** all edges e in Best-Cut:
16. **do** $t_e \leftarrow$ Best-Splitting-Point
17. **for** each connected component of $G' \in (V, E \setminus$ Best-Cut):
18. **do** *Min-Cut-Splitting*$(G', S$,Best-Splitting-Point,$\Lambda)$

Fig. 2. Algorithm 2

v that have splitting distances $\leq m_e$, $m_e \geq s_{i_1} > s_{i_2} > \ldots s_{i_j} = t_e$. We paid a total of

$$(m_e - s_{i_2})^p + [(m_e - s_{i_3})^p - (m_e - s_{i_2})^p] + \ldots + \left[(m_e - s_{i_j})^p - (m_e - s_{i_{j-1}})^p\right] = (m_e - t_e)^p$$

for this edge as Push-Down-Costs.

Claim 2: The Min-Split-Cost of each call is at most $\|T', M\|_p^p$

Consider a call *Min-Cut-Splitting*$(\widehat{G} = (\widehat{V}, \widehat{E}), \cdot, s_l, \cdot)$. If there exists an $e \in \widehat{E}$ such that $t'_e \geq s_l$, then $\{e \in \widehat{E} : t'_e \geq s_l\}$ contains at least one cut of which let C be the cut of minimum weight. For edges $e \in C$ the cost of cutting e is $(\max\{0, s_l - m_e\})^p \leq |t'_e - m_e|^p$. Hence the Cut-Cost is $\leq \|T', M\|_p^p$. The Push-Down-Cost is 0 since we are cutting in the first iteration of the loop; therefore,

$$\text{Min-Split-Cost} \leq \|T', M\|_p^p \ .$$

If all $e \in \widehat{E}$ satisfy $t'_e < s_l$ then let the splitting point be $s_{l'} = \max_{e \in \widehat{E}}\{t'_e\}$. The Push-Down-Cost is then at most

$$\sum_{e \in \widehat{E}} (\max\{0, m_e - s_{l'}\})^p \leq \sum_{e \in \widehat{E}: m_e > t'_e} (m_e - t'_e)^p \ .$$

Now the set of edges $\{e \in \widehat{E} : t'_e = s_{l'}\}$ contains at least one cut and, as before, choosing the minimum weight cut, call it C, results in the Cut-Cost being equal to $\sum_{e \in C} (\max\{0, s_{l'} - m_e\})^p = \sum_{e \in C: t'_e > m_e} (t'_e - m_e)^p$. Hence,

$$\text{Min-Spilt-Cost} \leq \sum_{e \in \widehat{E}: m_e > t'_e} (m_e - t'_e)^p + \sum_{e \in C: t'_e > m_e} (t'_e - m_e)^p \leq \|T', M\|_p^p \ .$$

The number of recursive calls of *Min-Cut-Splitting* is $n-1$ because each call fixes an internal node of the tree being constructed and the tree has n leaves.

Therefore, $\|T, M\|_p^p \leq (n-1)\|T', M\|_p^p$ and the theorem follows. Note that while a slightly better analysis gives that $\|T, M\|_p^p \leq D\|T', M\|_p^p$ where D is the depth of the recursion tree, D can be as much as $n-1$.

4 Extension to Additive Trees

In this section, we will generalize our results to approximating the input matrix M by general additive metrics under any L_p norm. Our generalization depends on the following theorem from [1],

Theorem 4 (see [1, Theorem 6.2]). *If $\mathcal{G}(M)$ is an algorithm which achieves an α-approximation to the optimal a-restricted ultrametric under the L_p norm, then there is an algorithm $\mathcal{F}(M)$ which achieves a 3α-approximation to the optimal additive metric under the same norm.*

We will show how our algorithms from section 3 can be used to produce a-restricted ultrametrics. We start with the definition of an a-restricted ultrametric from [1].

Definition 1. *For a point a, an ultrametric T^a is a-*restricted *with respect to a distance matrix M if*
(1) $T^a[a,i] = 2\mu_a$ *for all* $i \neq a$,
(2) $2\mu_a \ \geq \ T^a[i,j] \ \geq \ 2\,(\mu_a - \min\{M[a,i], M[a,j]\})$ *for all* i, j
where $\mu_a = \max_i M[a,i]$.

The definition of an a-restricted ultrametric immediately implies a procedure for approximating the distance $\|T^a_{\text{OPT}}, M\|_p$ between an optimal T^a_{OPT} and M. For a point a, let M^a be the matrix M with row a and column a deleted. And let Λ^a be the $n-1 \times n-1$ edge lower bounds matrix where

$$\Lambda^a[i,j] = 2(\mu_a - \min\{M[a,i], M[a,j]\}) \ ,$$

for all $i, j \in [n] \setminus \{a\}$, $i \neq j$. Given G^a, the graph representing M^a, and Λ^a our algorithms now find an a-restricted ultrametric T^a such that

$$\|T^a, M\|_p \;\leq\; O(\min\{n^{1/p}, (k \log n)^{1/p}\})\; \|T^a_{\mathrm{OPT}}, M\|_p \;\;.$$

Appealing to Thm. 4, we have a $O(\min\{n^{1/p}, (k \log n)^{1/p}\})$-approximation to the optimal additive metric under L_p.

5 Conclusions and Further Work

In this paper we have looked at embedding metrics into additive trees and ultrametrics. We have presented two algorithms, one suitable when the number of distinct distances in the metric is small, and one suitable when the number of distinct distances is large. Both algorithms are intrinsically greedy; they construct trees in a top-down fashion, establishing each internal node in turn by considering the immediate cost of the split it defines. Using these algorithms we provide the first approximation guarantees for this problem; however, there is scope for improving those guarantees.

Addendum: We recently learned that, independent of our work, Ailon and Charikar [15] have obtained improved results. They use ideas similar to those in our work.

References

1. Agarwala, R., Bafna, V., Farach, M., Paterson, M., Thorup, M.: On the approximability of numerical taxonomy (fitting distances by tree metrics). SIAM J. Comput. **28** (1999) 1073–1085
2. Ma, B., Wang, L., Zhang, L.: Fitting distances by tree metrics with increment error. J. Comb. Optim. **3** (1999) 213–225
3. Farach, M., Kannan, S.: Efficient algorithms for inverting evolution. Journal of the ACM **46** (1999) 437–450
4. Cryan, M., Goldberg, L., Goldberg, P.: Evolutionary trees can be learned in polynomial time in the two state general markov model. SIAM J. Comput **31** (2001) 375 – 397
5. Waterman, M., Smith, T., Singh, M., Beyer, W.: Additive evolutionary trees. J. Theoretical Biology **64** (1977) 199–213
6. Day, W.: Computational complexity of inferring phylogenies from dissimilarity matrices. Bulletin of Mathematical Biology **49** (1987) 461–467
7. Farach, M., Kannan, S., Warnow, T.: A robust model for finding optimal evolutionary trees. Algorithmica **13** (1995) 155–179
8. Emanuel, D., Fiat, A.: Correlation clustering - minimizing disagreements on arbitrary weighted graphs. In Battista, G.D., Zwick, U., eds.: ESA. Volume 2832 of Lecture Notes in Computer Science., Springer (2003) 208–220
9. Demaine, E.D., Immorlica, N.: Correlation clustering with partial information. In Arora, S., Jansen, K., Rolim, J.D.P., Sahai, A., eds.: RANDOM-APPROX. Volume 2764 of Lecture Notes in Computer Science., Springer (2003) 1–13

10. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. In: Proc. of the 43rd IEEE Annual Symposium on Foundations of Computer Science. (2002) 238
11. Charikar, M., Guruswami, V., Wirth, A.: Clustering with qualitative information. In: Proc. of the 44th IEEE Annual Symposium on Foundations of Computer Science. (2003) 524
12. Dhamdhere, K.: Approximating additive distortion of embeddings into line metrics. In Jansen, K., Khanna, S., Rolim, J.D.P., Ron, D., eds.: APPROX-RANDOM. Volume 3122 of Lecture Notes in Computer Science., Springer (2004) 96–104
13. Dhamdhere, K., Gupta, A., Ravi, R.: Approximation algorithms for minimizing average distortion. In Diekert, V., Habib, M., eds.: STACS. Volume 2996 of Lecture Notes in Computer Science., Springer (2004) 234–245
14. Garg, N., Vazirani, V.V., Yannakakis, M.: Approximate max-flow min-(multi)cut theorems and their applications. SIAM J. Comput. **25** (1996) 235–251
15. Ailon, N., Charikar, M.: Personal comunication (2005)