Approximate classification via earthmover metrics

Aaron Archer* Jittat Fakcharoenphol† Chris Harrelson‡ Robert Krauthgamer§

Kunal Talwar¶ Éva Tardos‖

Abstract

Given a metric space (X, d), a natural distance measure on probability distributions over X is the *earthmover metric*. We use randomized rounding of earthmover metrics to devise new approximation algorithms for two well-known classification problems, namely, *metric labeling* and 0*-extension*.

Our first result is for the 0-extension problem. We show that if the terminal metric is decomposable with parameter α (e.g., planar metrics are decomposable with $\alpha = O(1)$), then the earthmover based linear program (for 0-extension) can be rounded to within an $O(\alpha)$ factor.

Our second result is an $O(\log n)$-approximation for metric labeling, using probabilistic tree embeddings in a way very different from the $O(\log k)$-approximation of Kleinberg and Tardos. (Here, n is the number of nodes, and k is the number of labels.) The key element is rounding the earthmover based linear program (for metric labeling) without increasing the solution's cost, when the input graph is a tree. This rounding method also provides an alternate proof to a result stated in Chekuri et al., that the earthmover based linear program is integral when the input graph is a tree.

Our simple and constructive rounding techniques contribute to the understanding of earthmover metrics and may be of independent interest.

*Operations Research Department, Cornell University, Ithaca, NY 14853. Email: aarcher@cs.cornell.edu. Supported by the Fannie and John Hertz Foundation and NSF grant CCR-0113371.

†Kasetsart University, Bangkok, Thailand. Work partially done while the author was a graduate student at UC Berkeley. Email: jtf@ku.ac.th

‡UC Berkeley. Supported in part by a GAANN fellowship and NSF grant CCR-0105533. Email: chrishtr@cs.berkeley.edu

§IBM Almaden Research Center, 650 Harry Road, San Jose CA 95120, USA. Part of this work was done while this author was with the International Computer Science Institute and with the Computer Science Division of U.C. Berkeley. Email: robi@almaden.ibm.com

¶UC Berkeley. Work partially done while the author was interning at Microsoft Research. Email: kunal@cs.berkeley.edu

‖Computer Science Department, Cornell University, Ithaca, NY 14853. Research supported in part by NSF grant CCR-0113371. Email: eva@cs.cornell.edu

1 Introduction

The *metric labeling problem* has been proposed by Kleinberg and Tardos [KT02] to model classification problems from several domains, ranging from categorizing web documents to machine vision, including image recovery and the stereo matching problem. Metric labeling models situations in which we wish to label nodes in a graph given some prior information about their true labels, and also information about whether each pair of nodes should have similar labels. See [KT02] for a motivation from the perspective of Markov random fields.

Formally, the input is an n-node graph $G = (V, E)$ with nonnegative edge weights $w(\cdot, \cdot)$, a set of labels $L = [k]$,[1] assignment costs $c(v, i)$ for all $v \in V, i \in L$, and a metric $d(\cdot, \cdot)$ on the set of labels. The goal is to find a labeling $f : V \to L$ that minimizes the total sum of two kinds of costs. For each node $v \in V$ we incur an *assignment cost* $c(v, f(v))$, which penalizes us for assigning v a label $f(v)$, according to the prior (un)likelihood of v having that label. For each edge $uv \in E$ we also incur a *separation cost* $w(u, v)d(f(u), f(v))$, which penalizes us for assigning dissimilar labels to adjacent nodes.

The 0*-extension* problem is a special case of metric labeling, in which k of the nodes are designated as *terminals*, whose labels are fixed in advance, and all assignment costs are zero. When, in addition, the metric on the labels is uniform, the problem further reduces to the *multiway cut* problem, which is NP-hard (and actually APX-hard [DJP+94]) even for just three terminals. Thus, 0-extension and metric labeling are also NP-hard.

Earthmover metric. In this paper, we consider a linear program (LP) for these problems based on a family of metrics, called earthmover metrics. Given a metric (X, d), the *earthmover metric* with respect to (X, d) is a natural extension of d to probability distributions over X. Consider mounds of earth sitting on points of X, and suppose that moving an ϵ amount of earth from point i to point j takes $\epsilon \cdot d(i, j)$ work.

[1] Throughout, $[k]$ denotes the set $\{1, \ldots, k\}$.

If x and y are probability distributions over X, define $d_{\mathrm{EM}}(x,y)$ to be the minimum amount of work necessary to move the mounds from the x configuration to the y configuration.

There are several important properties of the earthmover metric. First, if d is a metric, then so is d_{EM} (as the reader has already guessed from the name). Second, if x^i denotes the probability distribution (over X) that has all its mass concentrated on point $i \in X$ (i.e., $x_i = 1$ and $x_l = 0$ for all $l \neq i$), then $d_{\mathrm{EM}}(x^i, x^j) = d(i,j)$ for all $i, j \in X$. Hence, d_{EM} is indeed an extension of d. The distance $d_{\mathrm{EM}}(x,y)$ is given by solving a transportation problem. That is, it is the optimal value of the following linear program (LP), where each "flow" variable z_{ij} represents the amount of earth to be moved from label i to label j when going from configuration x to configuration y.

$$\begin{array}{lll} \text{Minimize} & \sum_{i,j \in X} d(i,j) z_{ij} & \\ \text{subject to} & \sum_j z_{ij} = x_i & \forall i \in X \\ & \sum_i z_{ij} = y_j & \forall j \in X \\ & z_{ij} \geq 0 & \forall i,j \end{array}$$

Note that in the special case where the metric d is uniform, the earthmover distance d_{EM} reduces to the ℓ_1 distance between the two distributions. Indeed, one of our original motivations for this work was to better understand the relationship between ℓ_1 metrics and earthmover metrics.

Earthmover based relaxation. The classification problems mentioned above require us to map every node of G to a label in L. A natural relaxation is to allow the nodes to be mapped into the earthmover metric that corresponds to the label metric (L,d). This relaxation can be expressed by an LP. We write this LP in terms of d_{EM}, since this is how we shall think about it. Namely, the configuration assigned to node $v \in V$, denoted by x_v, represents the vector $\{x_{vi}\}_{i \in L}$ (whose components are the actual variables in the LP). This further suppresses the $|L|^2$ flow variables necessary for representing d_{EM} on each edge of G.

Definition 1.1. (Earthmover LP) *The* earthmover linear program *for metric labeling is as follows.*

$$\begin{array}{lll} \text{Min.} & \sum_{v \in V, i \in L} c(v,i) x_{vi} + \sum_{uv \in E} w_{uv} d_{\mathrm{EM}}(x_u, x_v) & \\ \text{s.t.} & \sum_{i \in L} x_{vi} = 1 & \forall v \in V \\ & x_{vi} \geq 0 & \forall i \in L, v \in V \end{array}$$

Adding integrality constraints on the variables yields an exact formulation. For the 0-extension problem, we identify the labels with the k terminals, fixing $x_{ii} = 1$ for all $i \in L$. All assignment costs are zero, so the first term in the objective function is zero. Hence this LP relaxation also makes sense for 0-extension.

1.1 Preliminaries and related work

Definition 1.2. (Decomposable metric) *A metric space (X,d) is called* α-decomposable *if for every $\delta > 0$, there is a randomized algorithm that partitions X into* clusters *(subsets) $\{X_l\}_l$ such that*[2]

1. *Each X_l has diameter at most δ.*
2. *For every $x, y \in X$, the probability that x and y fall in different clusters is at most $\alpha \frac{d(x,y)}{\delta}$.*

We say that a graph (or family of graphs) is α-decomposable if the shortest path metric on the graph is α-decomposable. For example, it is easy to see that a line metric is 1-decomposable. It is well-known (see [AP90, LR99, LS93, Bar96]) that any graph on n vertices is $O(\log n)$-decomposable, and that for some graphs (expanders) this bound is the best possible. Klein, Plotkin and Rao [KPR93] showed that planar graphs are $O(1)$-decomposable, and that more generally, graphs excluding a fixed minor of size r are $O(r^3)$-decomposable. Fakcharoenphol and Talwar [FT03] improved the latter bound to $O(r^2)$. Charikar et al. [CCG+98] showed that ℓ_p^d-metrics are $d^{\max\{\frac{1}{p}, 1-\frac{1}{p}\}}$-decomposable for any $p \geq 1$, and that this bound is tight.

The 0-extension problem was posed by Karzanov [Kar98], who gave an LP relaxation that we will call the *semi-metric LP relaxation*. He showed that this linear program is integral when the terminal metric is a tree (actually, a larger class of bipartite graphs that contains trees; here, a tree metric cannot contain Steiner nodes). Calinescu, Karloff and Rabani [CKR01] gave an $O(\log k)$ approximation for 0-extension based on this LP relaxation. For the special case of *input graphs* that are planar, they gave an $O(1)$ approximation algorithm. In fact, their algorithm extends to α-decomposable input graphs, giving an $O(\alpha)$ approximation. Fakcharoenphol et al. [FHRT03] improved the general result to $O(\log k / \log\log k)$. On the other hand, the semi-metric relaxation is known to have an $\Omega(\sqrt{\log k})$ integrality gap [CKR01]. Finally, we note that Lee and

[2] A stronger definition of decomposability that is sometimes used (*padded* decomposition) requires that the distance from a vertex to its cluster's boundary is well distributed. Our results do not require this condition.

Naor [LN03] have recently and independently obtained a 0-extension result similar to ours, but using the semi-metric relaxation.

The semi-metric LP relaxation allows nodes to be mapped into an arbitrary metric containing the terminal metric.[3] Since the earthmover metric is a specific metric of this type (containing the terminal metric), the earthmover LP relaxation has an integrality gap no larger than that of the semi-metric relaxation. (Furthermore, all the above approximation algorithms still work.) However, it is still unknown whether the (worst-case) integrality gaps of the two relaxations are different. Interestingly, it is not known if the integrality gap of the earthmover LP relaxation is more than a constant, even for metric labeling.

The metric labeling problem has been studied for some time in the computer vision community. A polynomial-time algorithm for line metrics and a constant-factor approximation for uniform metrics were given by Boykov et al. [BVZ98, BVZ01]. Gupta and Tardos [GT00] gave a constant factor approximation on truncated line metrics. Kleinberg and Tardos [KT02] gave a 2-approximation algorithm for the case of a uniform metric on the labels, and combined this with Bartal's probabilistic tree embeddings [Bar98] to obtain an $O(\log k \log\log k)$-approximation for general metrics. Fakcharoenphol et al. [FRT03] recently improved Bartal's result, leading to an improved bound of $O(\log k)$.

The earthmover LP was proposed independently by Charikar [Cha00] and by Chekuri et al. [CKNZ01]. The first successful use of this LP was in [CKNZ01], where it was used to give matching or improved algorithms for certain classes of distance functions d (convex and truncated linear), which had been previously addressed using flow techniques and local search [BVZ98, BVZ01, GT00, IG98]. It is worth noting that this is the first LP relaxation that directly models the general metric labeling problem. Previously, LP relaxations were only known for various simple classes of metrics.

Interestingly, the history of the multiway cut problem parallels that of 0-extension in some ways. Dahlhaus et al. [DJP+94] introduced the problem, showed it is APX-hard, and gave a $2(1-1/k)$ approximation using a combinatorial *isolation heuristic*. It is known that a simple randomized algorithm based on the semi-metric LP relaxation matches this factor, and that the integrality gap for this LP is also $2(1-1/k)$ [Vaz01, p.155]. Calinescu et al. [CKR00] strengthened the LP by embedding the nodes of V in ℓ_1, using it to obtain a $(\frac{3}{2}-\frac{1}{k})$-approximation. Karger et al. [KKS+99] further improved these results to a bound approaching 1.38 as $k \to \infty$. The LP used for these algorithms is exactly the same as the earthmover LP when the metric on labels is uniform. It is plausible that the earthmover LP will be similarly used to improve the approximation bound for the general case of the 0-extension and metric labeling problems. We believe that our results make progress in this direction.

1.2 Our contributions We devise two approximation algorithms that are based on rounding the earthmover LP. Our techniques improve the understanding of earthmover metrics and may be of independent interest.

For 0-extension, we show in Section 2 that if the *terminal metric* is α-decomposable, then there is a simple rounding algorithm for the earthmover LP which approximates the LP solution by an integral solution within a factor of $O(\alpha)$. Previously, this approximation ratio was known to hold only under the more stringent requirement that the *input graph* is α-decomposable. Technically, we give a randomized rounding procedure such that the expected separation cost of every two vertices of G is no more than $O(\alpha)$ times larger than their earthmover distance (which is simply their contribution to the LP objective).

One case in which this is a significant improvement on the previous known approximations is the following. Suppose that there is a very large number of terminals, but they all lie in a low-dimensional normed space. In such cases, we know that α depends only on the dimension (e.g. $\alpha = \sqrt{d}$ for Euclidean space). Hence we can use our rounding algorithm to give an approximation which depends only on the dimension, and does not depend on the number of terminals (or the size of G) at all.

For metric labeling, we give in Section 3 an algorithm that achieves an $O(\log n)$-approximation. In many cases, we would expect n, the number of objects to be labeled, to far exceed k, the number of labels, and then our approximation guarantee is worse than the $O(\log k)$ currently known. However, there are applications where $k \gg n$. For instance, in recognizing the position of a human body from an image (see [FH00]), the objects are the handful of rigid moving parts, and each part needs to be labeled with a six-tuple representing position, rotation and scale.[4]

Our new algorithm for metric labeling uses the earthmover LP as follows. The LP solution defines

[3] Since several nodes may be mapped onto the same point in the containing metric, the induced distances between mapped nodes form only a semi-metric; hence the name. Our relaxation has the same property.

[4] The model used in [FH00] uses a different distance function for each edge, but is otherwise identical to our model.

an earthmover metric on V, which we probabilistically approximate with a tree, and then round. The crux is that our rounding method incurs no loss on the tree. Our use of tree approximations contrasts with that of [KT02], which instead approximates the label metric by a tree, then uses a specialized LP for trees. Our rounding method also proves that the earthmover LP has no integrality gap when the input graph is a tree. This phenomenon was previously mentioned by Chekuri et al. [CKNZ01] (a proof appears in their upcoming journal version), but our proof is very different.

Interestingly, this implies, for 0-extension on tree graphs G, a distinction between the integrality ratio of the earthmover LP and the semi-metric relaxations. The former is integral as a special case of the result above, while the latter has integrality ratio at least $2 - o(1)$ (e.g., if G is a star whose leaves are the terminals).

2 0-extension algorithm for decomposable terminal metrics

In this section we show that if the metric d on labels (terminals) is α-decomposable, then a solution to the 0-extension earthmover LP relaxation can be rounded, via a randomized algorithm, to an integral solution whose expected cost is no more than $O(\alpha)$ times the LP cost. Since, as remarked above, any k-point metric is $O(\log k)$-decomposable, we also get an $O(\log k)$ approximation to the 0-extension problem in general.[5] For better decomposable metrics, of course, the approximation ratio is better.

Let $T = [k]$ be the set of terminals and d be a metric on T. For each vertex u, let $x_u = \langle x_{u1}, x_{u2}, \ldots, x_{uk}\rangle$ be the distribution over terminals that is associated with u in the solution to the earthmover LP, and let d_{EM} be the induced earthmover metric. For a vertex $u \in V$, let $A_u = \min_{i\in T}\{d_{\mathrm{EM}}(x_u, i)\}$ be the distance from u to its closest terminal.

At a high level, the rounding algorithm consists of the following steps, each bringing us closer to an integral solution.

1. Break up V into groups $V_s = \{u : A_u \approx 2^s\}$, containing vertices u with approximately the same value of A_u. We will round each group separately.
2. *Truncate* the distribution x_u so that it is concentrated only on terminals "close" to u.
3. *Decompose* the terminal metric into clusters of diameter 2^s and choose a representative terminal from each cluster.
4. *Project* the distribution x_u onto representatives, so now all its mass is concentrated on representative terminals close to u.
5. *Round* each vertex in V_s to one of the representative terminals using the rounding algorithm of Kleinberg and Tardos [KT02] for metric labeling on uniform metrics.

Intuitively, this last step works since to the vertices in V_s, the metric on the representatives looks approximately uniform.

2.1 Formal rounding algorithm Assume without loss of generality that the smallest distance between terminals is 1, and denote the largest distance by Δ. Let Decomp(T, δ) be the decomposition algorithm implied by Definition 1.2.

1. $r_0 \leftarrow 0$.
2. For $s = 1, 2, \ldots, \lceil \log_2 \Delta \rceil$:
   - 2.1 Pick r_s uniformly at random in $[2^{s-1}, 2^s]$.
   - 2.2 (*Grouping*) Let $V_s = \{u \in V : r_{s-1} < A_u \le r_s\}$.
   - 2.3 Let $\delta = 2^s$.
   - 2.4 (*Truncation*) Pick uniformly at random $\gamma \in \{4\delta, \ldots, 7\delta\}$. For each $u \in v$, zero out x_{ui} whenever $d_{\mathrm{EM}}(x_u, i) > \gamma$ and rescale the remaining x_{ui}'s so that their sum is 1. Let x'_u be the resulting vector.
   - 2.5 Let $\{T_l\}_l$ be the clusters output by Decomp(T, δ).
   - 2.6 Pick an arbitrary terminal i_l in each cluster T_l, and make it the cluster's *representative*.
   - 2.7 (*Projection*) For $u \in V_s$, project the distributions x_u from the clusters onto the representatives:
   $$x''_{ui} \leftarrow \begin{cases} \sum_{i\in T_l} x'_{ui} & i = i_l \\ 0 & \text{otherwise} \end{cases}$$
   - 2.8 (*Rounding*) Apply on (V_s, x'') the rounding algorithm of [KT02] for metric labeling on the uniform metric.

2.2 Analysis Let f be the final assignment output by our algorithm, i.e. vertex u is assigned to terminal $f(u)$. For an edge uv, we will sometimes refer to $d_{\mathrm{EM}}(u, v) = d_{\mathrm{EM}}(x_u, x_v)$ as the *cost* of uv; the associated distribution vectors will be implicit from context. Also, since for terminals i, j, $d_{\mathrm{EM}}(i, j) = d(i, j)$, we will when convenient use the latter notation.

[5] We can in fact do slightly better; see the discussion at end of this Section.

Our analysis is edge by edge. We show that for every edge uv, the expected value of $d(f(u), f(v))$ is no more than $O(\alpha)$ times $d_{\mathrm{EM}}(x_u, x_v)$. Notice that the initial step of grouping breaks up the problem into several subproblems which we solve separately. Consequently, every edge can be either an *intragroup edge*, if its endpoints are in the same group V_s, or an *intergroup edge* that goes across different groups. Looking at the intragroup edges first, we show that each subsequent step of the algorithm does not increase the expected cost of any intragroup edge by too much. Intergroup edges are easier, and will be handled at the end.

Truncation. In the truncation step, we pick γ uniformly at random in $\{4\delta, \ldots, 7\delta\}$ and truncate each x_{ui} to be concentrated only on terminals within distance γ from u. We then rescale the x_{ui}'s to restore their sum to 1. We first show that we never need to rescale by too much.

LEMMA 2.1. *For any vertex u and any terminal i, $x'_{ui} \le \frac{3}{2} x_{ui}$.*

Proof. Let i^* be the terminal closest to u and consider the iteration in which $u \in V_s$. The grouping phase ensures that $d_{\mathrm{EM}}(u, i^*) \le r_s \le \delta$. By definition, $d_{\mathrm{EM}}(u, i^*) = \sum_{i \ne i^*} x_{ui} d_{\mathrm{EM}}(i^*, i) \ge \sum_{i: d_{\mathrm{EM}}(i^*, i) > 3\delta} 3\delta x_{ui}$. Thus $\sum_{i: d_{\mathrm{EM}}(i^*, i) > 3\delta} x_{ui} \le \frac{1}{3}$ and so $\sum_{i: d_{\mathrm{EM}}(i^*, i) \le 3\delta} x_{ui} \ge 2/3$.

Moreover, by the triangle inequality, $d_{\mathrm{EM}}(u, i) \le d_{\mathrm{EM}}(u, i^*) + d_{\mathrm{EM}}(i^*, i)$. Thus at least two-thirds of the probability mass in x_u is concentrated on terminals within distance 4δ from u. Since $\alpha \ge 4\delta$, the truncation removes no more than a third of the probability mass, and the scaling therefore requires multiplication by a factor no larger than $3/2$. $\square$

We now show that the transformation does not increase the cost of an edge uv by much.[6]

LEMMA 2.2. *For any intragroup edge uv,*

$$\mathbb{E}[d_{\mathrm{EM}}(x'_u, x'_v)] \le 16 \cdot d_{\mathrm{EM}}(x_u, x_v).$$

Proof. Recall that $d_{\mathrm{EM}}(x_u, x_v) = \sum_{ij} z_{ij} d(i, j)$, where the z_{ij} flow variables transform the distribution x_u to x_v. We will show how to modify the z_{ij}'s slightly so that they become a valid flow to transform x'_u to x'_v, and such that the expected cost of this new flow is not too large. Clearly the earthmover distance $d_{EM}(x'_u, x'_v)$, being the cost of the best flow, is no larger.

[6] We make no attempts to optimize the constants here or anywhere else in the paper.

To define the new flow z'_{ij}, we need to set up some terminology. Let T_u and T_v be the sets of terminals within distance γ from u and from v, respectively. Let $p_u = \sum_{i \in T_u} x_{ui}$ and $q_u = 1 - p_u$, and define p_v and q_v similarly. Without loss of generality, assume $p_u \ge p_v$. The original flow variables z naturally fall into four categories : f_{pp}, f_{pq}, f_{qp} and f_{qq}, where, for example, the flow f_{pq} goes from $i \in T_u$ to $j \notin T_v$. Notice that the new flow z'_{ij} transforms x'_u to x'_v, and is thus nonzero only on flow going from T_u to T_v.

To create the new flow, first scale up the first kind of flow f_{pp} by a factor of $1/p_u$. Set the flows in f_{qp} and f_{qq} to zero. Finally, scale up the flows in f_{pq} by $1/p_u$ and re-route them greedily to terminals in T_v. It is easy to verify that the total flow leaving T_u is 1, the total flow leaving $V \setminus T_u$ is 0, the total flow entering $V \setminus T_v$ is 0, and by conservation of flow, the total flow entering T_v is 1. Furthermore, the total flow leaving every $i \in T_u$ is at most x'_{ui}, and, since $p_u \ge p_v$, the last step can be done so that the total flow entering every $i \in T_v$ is at most x'_{vi}. Thus we get a feasible flow.

Notice that by Lemma 2.1, $p_u, p_v \ge 2/3$, and hence the scaling only increases any cost by a small factor. The flows in f_{pq} are rerouted to terminals further away. Thus these flows contribute to an increase in cost, and this cost needs to be bounded.

We first show that there isn't too much such flow. Consider any flow variable z_{ij}. For this flow to be in f_{pq} or f_{qp} (note that we now cannot assume that $p_u \ge p_v$), it must be the case that exactly one of the two events $d_{\mathrm{EM}}(u, i) \le \gamma$ and $d_{\mathrm{EM}}(v, j) \le \gamma$ happen. In other words, γ must fall between $d_{\mathrm{EM}}(u, i)$ and $d_{\mathrm{EM}}(v, j)$. The probability of that happening is clearly at most $|d_{\mathrm{EM}}(v, j) - d_{\mathrm{EM}}(u, i)|/3\delta$. Further, by the triangle inequality, $|d_{\mathrm{EM}}(v, j) - d_{\mathrm{EM}}(u, i)| \le d_{\mathrm{EM}}(u, v) + d_{\mathrm{EM}}(i, j)$. Thus flow z_{ij} falls in the bad categories with probability at most $\min\{1, (d_{\mathrm{EM}}(u, v) + d_{\mathrm{EM}}(i, j))/3\delta\}$.

On the other hand, if a flow in f_{pq} is rerouted, it now goes from $i \in T_u$ to some $j' \in T_v$ (or is distributed amongst more than one such j'). The distance traveled by the flow is $d_{\mathrm{EM}}(i, j') \le d_{\mathrm{EM}}(i, u) + d_{\mathrm{EM}}(u, v) + d_{\mathrm{EM}}(v, j') \le d_{\mathrm{EM}}(u, v) + 14\delta$.

Thus the expected cost of the rerouted flow (ignoring the scaling) is

$$\begin{aligned}
&\sum_{i,j} z_{ij} \cdot \Pr[z_{ij} \text{ is rerouted}] \cdot (\text{cost of rerouting}) \\
&\quad \le \sum_{i,j} z_{ij} \cdot \min\{1, (d_{\mathrm{EM}}(u, v) + d_{\mathrm{EM}}(i, j))/3\delta\} \\
&\quad\quad \cdot (d_{\mathrm{EM}}(u, v) + 14\delta) \\
&\quad \le \sum_{i,j} z_{ij} \cdot d_{\mathrm{EM}}(u, v) + \frac{14}{3} \cdot \sum_{i,j} z_{ij}(d_{\mathrm{EM}}(i, j) + d_{\mathrm{EM}}(u, v))
\end{aligned}$$

$$\leq \quad d_{\mathrm{EM}}(u,v) + \frac{14}{3} d_{\mathrm{EM}}(u,v) + \frac{14}{3} d_{\mathrm{EM}}(u,v)$$
$$= \quad \frac{29}{3} d_{\mathrm{EM}}(u,v),$$

where in the second to last step, we use the fact that $\sum_{i,j} z_{ij} d_{\mathrm{EM}}(i,j)$ is exactly $d_{\mathrm{EM}}(x_u, x_v)$.

Since the scaling is by a factor no larger than 3/2 and $(1 + \frac{29}{3}) \cdot \frac{3}{2} = 16$, the lemma follows. $\square$

Clustering and projection. Intuitively, we would like the clustering and projection steps to convert the metric on the host space T to an approximately uniform metric. On this metric, the earthmover distance is almost the same as the ℓ_1 metric, and we can the proceed using the Kleinberg-Tardos rounding.

We now show that clustering of terminals and the projection does just this. More precisely, we relate the ℓ_1 distance between x''_u and x''_v with the earthmover distance $d_{\mathrm{EM}}(x'_u, x'_v)$.

Lemma 2.3. *For any $u, v \in V_s$,*

$$\mathbb{E}[||x''_u - x''_v||_1] \quad \leq \quad \frac{\alpha}{2^s} \cdot d_{\mathrm{EM}}(x'_u, x'_v)$$

Proof. Let the flow values z'_{ij} be such that $d_{\mathrm{EM}}(x'_u, x'_v) = \sum_{i,j} d(i,j) z'_{ij}$. We will make use of those values below. Let $\rho(i,j)$ be 1 if terminals i and j are separated by the clustering step, and 0 otherwise. Let ρ_{EM} be the earthmover distance w.r.t. ρ. Now we have that

$$\mathbb{E}[||x''_u - x''_v||_1] \quad = \quad \mathbb{E}[\rho_{EM}(x''_u, x''_v)] \tag{2.1}$$
$$\leq \quad \mathbb{E}[\sum_{i,j} \rho(i,j) z'_{ij}] \tag{2.2}$$
$$\leq \quad \sum_{i,j} \frac{\alpha}{2^s} d(i,j) z'_{ij} \tag{2.3}$$
$$= \quad \frac{\alpha}{2^s} d_{EM}(x'_u, x'_v).$$

The first equality (2.1) holds because ρ is a uniform metric, for which the earthmover metric is ℓ_1. The following inequality (2.2) holds because one (not necessarily optimal) way to route flow from x''_u to x''_v is to take the flow induced by z'_{ij}, i.e., the flow from one cluster representative to another equals the total flow that z'_{ij} has routed from all the nodes in the first cluster to all the nodes in the second cluster; in this case, flow originating from z'_{ij} is associated with cost $\rho(i,j)$. The last inequality (2.3) is due to the requirements of Definition 1.2. $\square$

Also note that since the clusters are of diameter δ, the distribution x_u for any terminal u is still concentrated on terminals within distance 8δ of u.

Kleinberg-Tardos rounding. We now go over the rounding algorithm given by Kleinberg and Tardos [KT02] for rounding the ℓ_1 linear program they used for metric labeling when the terminal metric is uniform. Given a set of terminals $1, \ldots, k$, a set of vertices V and distribution vectors x''_u for every $u \in V$, their algorithm assigns each vertex $u \in V$ to some terminal.

The algorithm is as follows:

1. While there is an unassigned vertex do
   1.1 Pick a terminal i uniformly at random from T.
   1.2 Pick a value c uniformly at random from $[0, 1]$.
   1.3 Assign to i all $u \in V$ such that $x''_{ui} \geq c$.

It is not difficult to show that the algorithm terminates in expected polynomial time. Kleinberg and Tardos show that the algorithm has the following nice properties. We repeat the proof here for completeness.

Lemma 2.4. *The algorithm KT has the following properties:*

- *For any vertex $u \in V$, the probability that u gets assigned to i is exactly x''_{ui}.*
- *For a pair of vertices $u, v \in V$, the probability that u and v get assigned to different terminals is at most $\|x''_u - x''_v\|_1$.*

Proof. In any iteration, conditioned on u being unassigned, the probability that it gets assigned to i is exactly $\frac{1}{|T|} x''_{ui}$. Thus the first property follows. Moreover, it follows that each vertex has a probability $\frac{1}{|T|}$ of being assigned to some terminal in each iteration.

Now let us look at a pair of vertices u and v. The probability that exactly one of u and v is assigned to some terminal in a particular iteration is exactly $\frac{1}{|T|} \sum_i |x''_{ui} - x''_{vi}| = \frac{1}{|T|} \|x''_u - x''_v\|_1$.

Thus, the probability that u and v are assigned to different terminals can be upper bounded by

$$\frac{\Pr[\text{exactly one of } u \text{ and } v \text{ assigned in an iteration}]}{\Pr[\text{at least one of } u \text{ and } v \text{ assigned in an iteration}]}$$

Plugging in the values of these probabilities, the second property follows. $\square$

Intragroup summary. Since the Kleinberg-Tardos algorithm assigns a vertex u to a terminal i with nonzero x''_{ui}, our truncation step ensures that any vertex u is assigned to a terminal within distance 8δ from u.

When vertices u and v are assigned to different terminals $f(u)$ and $f(v)$, the distance $d(f(u), f(v))$ is bounded by $d_{\mathrm{EM}}(f(u), u) + d_{\mathrm{EM}}(u, v) + d_{\mathrm{EM}}(v, f(v))$, where $d_{\mathrm{EM}}(f(u), u) \leq 8\delta$ and $d_{\mathrm{EM}}(f(v), v) \leq 8\delta$. Moreover, the probability that u and v are assigned different terminals is at most $\|x''_u - x''_v\|_1$, by Lemma 2.4. Thus, using Lemmas 2.2 and 2.3, the expected value of $d_{\mathrm{EM}}(f(u), f(v))$ is at most $\min\{1, 16(\alpha/\delta) \cdot d_{\mathrm{EM}}(u, v)\} \cdot (d_{\mathrm{EM}}(u, v) + 16\delta) = O(\alpha) \cdot d_{\mathrm{EM}}(u, v)$. Thus we have shown the following.

Lemma 2.5. *For every $u, v \in V_s$,*

$$\mathbb{E}[d(f(u), f(v))] = O(\alpha) \cdot d_{\mathrm{EM}}(x_u, x_v).$$

Intergroup edges. We now argue about the intergroup edges.

Lemma 2.6. *For any edge $uv \in E$, the contribution to the expected cost of uv due to intergroup cutting is $O(d_{\mathrm{EM}}(x_u, x_v))$.*

Proof. As argued above, each vertex $u \in V_s$ is assigned to a terminal closer than $8 \cdot 2^s$. Let $uv \in E$ be an edge that goes from $u \in V_s$ to $v \in V_{s'}$ where $s' \leq s$. Then $d(f(u), f(v))$ is at most $d_{\mathrm{EM}}(f(u), u) + d_{\mathrm{EM}}(u, v) + d_{\mathrm{EM}}(v, f(v))$. Since $d_{\mathrm{EM}}(u, f(u)) \leq 2^{s+3}$ and $d_{\mathrm{EM}}(f(v), v) \leq 2^{s+3}$, it follows that $d(f(u), f(v)) \leq d_{\mathrm{EM}}(u, v) + 2^{s+4}$.

Moreover, note that uv is an intergroup edge only if $A_u \leq r_{s-1} < A_v$, which happens with probability at most $\min\{1, (A_v - A_u)/2^{s-2}\} \leq \min\{1, d_{\mathrm{EM}}(u, v)/2^{s-2}\}$. Thus the expected contribution to the length of an edge uv due to the intergroup cutting is $O(d_{\mathrm{EM}}(u, v))$. $\square$

We note that the proof of lemma 2.6 does not use any property of the earthmover distance, and this grouping can be applied to the semi-metric relaxation as well. This leads to a much simpler proof of the Calinescu et al. [CKR01] result on decomposable graphs. We omit the details here.

Combining lemmas 2.5 and 2.6, we get

Theorem 2.1. *The algorithm described above gives an $O(\alpha)$ approximation algorithm to the 0-extension problem when the terminal metric is α-decomposable.*

We note that each of the random steps of the algorithm can be easily derandomized using standard techniques, leading to a deterministic algorithm with the same performance guarantee. We omit the details from this extended abstract.

Discussion. Recall that it is always the case that $\alpha = O(\log k)$, so that our algorithm has an $O(\log k)$ worst-case bound. However, the clustering and projection steps above could in fact have been done at any scale larger than 2^s; the analysis for the intragroup edges still goes through. Thus, if the terminal metric has a better decomposition at some large scale, the intragroup edges do not overpay in expectation. (The bound on the distance of a vertex from its assigned terminal, however, goes up, increasing the expected cost of intergroup edges.)

To take advantage of this, we pick a random $i \in [0, \frac{1}{2}\log\log k]$, and do the decomposition at scale 2^{s+i} instead. In expectation, this can be shown to be a worst-case $O(\log k/\log\log k)$-decomposition, and the intergroup edges are also fine in expectation. Thus we can improve the result for general graphs from $O(\log k)$ to $O(\log k/\log\log k)$, matching the result in [FHRT03].

It can be shown that the techniques of [CKR01] imply an $O(\beta)$ approximation to the problem when the terminal metric satisfies $|B(x, 2r)|/|B(x, r)| \leq 2^\beta$ for every x and r. (Here $B(x, r)$ is the set of points within radius r from x in the metric.) Since any such metric has a β-decomposition, their results follow from ours. However, there are metrics where β is much larger than α (e.g. d-dimensional Euclidean spaces have $\beta = d$ but have $\sqrt{d}$-decompositions) and thus our results are stronger than those implied by previous techniques.

3 Metric labeling via tree-rounding

In this section we use the earthmover LP to obtain an $O(\log n)$-approximation for the metric labeling problem. The basic idea of our algorithm is to use the LP to define an embedding of the nodes of V into the earthmover metric space defined from the label metric (L, d). We then approximate the metric space (V, d_{EM}) with a tree, and round in a coordinated way based on the tree. The crux of this method is that we can round on a tree while preserving both separation costs along tree edges and all assignment costs. Hence we start off by showing how to do this.

Coordinated rounding. Given two distributions x and y over labels, we show how to randomly round x and y to labels in a way that obeys the specified distributions, but is coordinated so that the expected distance between the resulting labels is $d_{\mathrm{EM}}(x, y)$. For $i \in [k+1]$, define the *breakpoint* $B_i = \sum_{\ell=1}^{i-1} x_i$. Identify label i with the interval $I_i = [B_i, B_{i+1})$, generate a uniform random variable $U_x \in [0, 1)$, and round x to the label into whose interval U_x falls. We now use the flow variables z_{ij} to determine how to round y. Further subdivide I_i into intervals $I_{ij} = [B_{ij}, B_{i(j+1)})$, where $B_{ij} = B_i + \sum_{\ell=1}^{j-1} z_{ij}, j \in [k+1]$. If U_x lands in the subinterval I_{ij}, we round x to i (since $U_x \in I_{ij} \subseteq I_i$) and y to j. We also generate a $[0, 1)$ random variable U_y as a function of U_x (for use later in further propagating the rounding). Break $[0,1)$ into intervals I'_j, and partition each I'_j into subintervals I'_{ij} by the z_{ij} flow values, analogously to what we did for x. If $U_x \in I_{ij}$, then set $U_y = B'_{ij} + (U_x - B_{ij})$. That is, U_y is defined so that it lands in the same spot in interval I'_{ij} as U_x did in interval I_{ij}. This process leads easily to the following proposition.

PROPOSITION 3.1. *In the rounding scheme above, x and y are each rounded to labels with probabilities matching their given distributions, and the expected distance between these labels is $d_{\mathrm{EM}}(x,y)$. Moreover, U_y is distributed uniformly in $[0,1)$.*

This rounding technique easily extends to trees, where each node is assigned a distribution over labels. Fix an arbitrary root node r, and generate a uniform $[0,1)$ random variable U_r. Round r according to U_r using the procedure described above, and propagate this rounding to all of r's children, thereby generating an associated random rounding variable U_c for each child c. In this way, we propagate the rounding all the way down the tree.

COROLLARY 3.1. *If the input graph is a tree, the earthmover LP has an integrality gap of* 1.

We note that the metric labeling problem is polynomial time solvable on trees using dynamic programming. The above corollary then provides an alternative approach. More importantly, this rounding technique leads to a new approach for the general metric labeling problem.

Our algorithm. A probabilistic tree approximation of a metric d with distortion c is a probability distribution over a collection of trees $\mathcal{T}$ such that for all $u, v \in V$, we have $\mathbb{E}_{T\in\mathcal{T}}[d_T(u,v)] \leq cd(u,v)$, while for every $T \in \mathcal{T}$ we have $d_T(u,v) \geq d(u,v)$. In general, the trees may contain Steiner points not in the original metric space. Our algorithm uses the following two results.

THEOREM 3.1. (FAKCHAROENPHOL ET AL. [FRT03]) *For every n-point metric d there is a probabilistic tree approximation with distortion $O(\log n)$ from which we can sample in polynomial time.*

THEOREM 3.2. (GUPTA [GUP01]) *For every tree $T = (V', E, w)$ and a set of required vertices $V \subseteq V'$, there exists a tree $T^* = (V, E^*, w^*)$ such that for all $u, v \in V$, $d_T(u,v) \leq d_{T^*}(u,v) \leq 8d_T(u,v)$.*

Here is our algorithm:

1. Solve the earthmover LP.
2. Define the metric d_{LP} on V by $d_{LP}(u,v) = d_{\mathrm{EM}}(x_u, x_v)$. Use Theorems 3.1 and 3.2 to generate a random tree T spanning just V that probabilistically approximates d_{LP} with dilation $O(\log n)$.
3. Compute the earthmover distances along each edge of T to obtain the flow variables. Select one node arbitrarily as the root, randomly round it, and propagate the results to the rest of the tree.

Let A and S denote the assignment and separation costs of our solution, while A_{LP} and S_{LP} denote the corresponding costs in the optimal LP solution.

THEOREM 3.3. *Our algorithm achieves $\mathbb{E}[A] = A_{LP}$ and $\mathbb{E}[S] \leq O(\log n)S_{LP}$. Thus, it provides $O(\log n)$-approximation for metric labeling.*

Proof. The LP solution provides a lower bound on the cost of OPT. By Proposition 3.1, the nodes of the tree are each rounded to labels according to the distribution specified by the LP, so the assignment cost is preserved in expectation. The separation cost is preserved in expectation along each tree edge, so by the triangle inequality, $\mathbb{E}[d(f(u), f(v))] \leq d_T(u,v)$ for all $u, v \in V$ (here the expectation is over the randomized rounding). But $\mathbb{E}_T[d_T(u,v)] \leq O(\log n)d_{\mathrm{EM}}(x_u, x_v)$, so summing over the edges of G yields the desired bound. $\square$

We note that this algorithm can also be easily derandomized. We omit the details.

Acknowledgment We would like to thank Eran Halperin and Satish Rao for several helpful discussions.

References

[AP90] B. Awerbuch and D. Peleg. Sparse partitions. In *31st Annual IEEE Symposium on Foundations of Computer Science*, pages 503–513, 1990.

[Bar96] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science*, pages 184–193. IEEE, 1996.

[Bar98] Y. Bartal. On approximating arbitrary metrics by tree metrics. In *30th Annual ACM Symposium on Theory of Computing*, pages 161–168. ACM, 1998.

[BVZ98] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *7th IEEE International Conference on Computer Vision*, pages 377–384, 1998.

[BVZ01] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 23(11):1222–1239, Nov. 2001.

[CCG+98] M. Charikar, C. Chekuri, A. Goel, S. Guha, and S. Plotkin. Approximating a finite metric by a small number of tree metrics. In *39th Annual Symposium on Foundations of Computer Science*, pages 379–388, 1998.

[Cha00] M. Charikar. Personal communication, January 2000.

[CKNZ01] C. Chekuri, S. Khanna, J. Naor, and L. Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. In *12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 109–118, 2001.

[CKR00] G. Calinescu, H. J. Karloff, and Y. Rabani. An improved approximation algorithm for multiway cut. *JCSS*, 60(3):564–574, 2000.

[CKR01] G. Calinescu, H. Karloff, and Y. Rabani. Approximation algorithms for the 0-extension problem. In *12th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2001.

[DJP+94] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM J. Comput.*, 23(4):864–894, 1994.

[FH00] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient matching of pictorial structures. In *IEEE Computer Vision and Pattern Recognition Conference*, pages 66–73, 2000.

[FHRT03] J. Fakcharoenphol, C. Harrelson, S. Rao, and K. Talwar. An improved approximation algorithm for the 0-extension problem. In *14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 257–265, 2003.

[FRT03] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *35th Annual ACM Symposium on Theory of Computing*, pages 448–455, 2003.

[FT03] J. Fakcharoenphol and K. Talwar. Improved decompositions of graphs with forbidden minors. In *6th International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 36–46, 2003.

[GT00] A. Gupta and É. Tardos. A constant factor approximation algorithm for a class of classification algorithms. In *32nd Annual ACM Symposium on Theory of Computing*, pages 652–658, 2000.

[Gup01] A. Gupta. Steiner points in tree metrics don't (really) help. In *12th Annual ACM-SIAMSymposium on Discrete Algorithms*, pages 220–227, 2001.

[IG98] H. Ishikawa and D. Geiger. Segmentation by grouping junctions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

[Kar98] A. Karzanov. Minimum 0-extensions of graph metrics. *European Journal of Combinatorics*, 19(1):71–101, 1998.

[KKS+99] D. R. Karger, P. Klein, C. Stein, M. Thorup, and N. E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. In *31st Annual ACM Symposium on Theory of Computing*, pages 668–678, 1999.

[KPR93] P. Klein, S. A. Plotkin, and S. Rao. Excluded minors, network decomposition, and multicommodity flow. In *25th Annual ACM Symposium on Theory of Computing*, pages 682–690, May 1993.

[KT02] J. M. Kleinberg and É. Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. *Journal of the Association for Computing Machinery*, 49:616–639, 2002.

[LN03] J. R. Lee and A. Naor. Metric decomposition, smooth measures, and clustering, 2003. Submitted.

[LR99] F. T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *JACM*, 46(6):787–832, 1999.

[LS93] N. Linial and M. Saks. Low diameter graph decompositions. *Combinatorica*, 13(4):441–454, 1993.

[Vaz01] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, New York, 2001.