Hierarchical Clustering of Data Streams: Scalable Algorithms and Approximation Guarantees

Anand Rajagopalan [1] **Fabio Vitale** [2] **Danny Vainstein** [3] **Gui Citovsky** [1] **Cecilia M. Procopiuc** [1] **Claudio Gentile** [1]

Abstract

We investigate the problem of hierarchically clustering data streams containing metric data in $\mathbb{R}^d$. We introduce a desirable invariance property for such algorithms, describe a general family of hyperplane-based methods enjoying this property, and analyze two scalable instances of this general family against recently popularized similarity/dissimilarity-based metrics for hierarchical clustering. We prove a number of new results related to the approximation ratios of these algorithms, improving in various ways over the literature on this subject. Finally, since our algorithms are principled but also very practical, we carry out an experimental comparison on both synthetic and real-world datasets showing competitive results against known baselines.

1. Introduction

Hierarchical clustering (HC) is a fundamental tool of data analysis by which data items are grouped together based on some notion of (semantic) similarity working at different levels of granularity. The goal of a HC algorithm operating on a dataset X is then to construct a tree whose leaves host the data items in X, and whose internal nodes encode subsets of X (that is, the clusters) at increasing levels of resolution from root to leaves. Applications are ubiquitous, ranging from phylogeny (e.g., (Eisen et al., 1998)) to data mining and information retrieval (e.g., (Manning et al., 2008)), to social network analysis (e.g., (Gilbert et al., 2011)), and beyond.

In practice, HC methods are often deployed in data-intensive applications, where massive datasets have to be hierarchically organized and connected to data-acquisition pipelines within highly dynamic (and typically non-stationary) environments. In these contexts, it is crucial to devise adaptive HC solutions that enable the handling of massive data streams in a robust and efficient manner. In HC for data streams, it is a common desideratum to have a fast way of updating the hierarchy with the newly acquired data without recomputing everything from scratch. Yet, at the same time, we would like to do so without exposing ourselves to unexpected temporal behaviors of the data stream that skew the hierarchy towards undesirable configurations.

Contributions. In this paper, we present the general algorithmic framework of hyperplane-based HC for data streams containing metric data. The (randomized) algorithms originating from this framework are purely geometric algorithms that can interchangeably be described as *batch* HC solutions (the dataset X is given up front in its entirety) or *dynamic* (aka sequential) HC solutions (the data points in X are disclosed one by one or in small batches). Crucially, within our framework, the two solutions turn out to be *statistically equivalent*, in that the statistical properties of the trees computed in the batch mode are the same as those for trees computed in the sequential mode. This means that the specific ordering of data by which the tree structure is grown does not affect the properties of the final tree, thereby giving our HC solutions a desirable robustness. We call this the *sequential property* of HC algorithms (see Section 2 for a formal definition). Moreover, the computed hierarchy is fully online in the sense that points are inserted as siblings of existing nodes, without changing the tree topology.

Quality measures: In order to evaluate the quality of our HC solutions, we follow the recent trend initiated by Dasgupta (2016), and further developed by a number of more recent works (e.g., (Charikar & Chatziafratis, 2017; Cohen-addad et al., 2019; Charikar et al., 2019b; Cohen-addad et al., 2019; Naumov et al., 2020; Alon et al., 2020; Vainstein et al., 2021)), who framed the HC problem as a combinatorial optimization problem over hierarchical structures against exogenous pairwise similarity/dissimilarity information on individual data points. In practice, as emphasized, e.g., by Charikar et al. (2019b); Naumov et al. (2020), data are often described by feature vectors, so that this pairwise information can be naturally delivered by the underlying

[1] Google Research, NY, USA [2] Lille University and INRIA Lille, France [3] Tel-Aviv University, Israel. Correspondence to: Anand Rajagopalan <anandbr@google.com>, Fabio Vitale <fabio.vitale@inria.fr>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

metric structure (e.g., ℓ_1 or ℓ_2) where data lies. In this paper we consider several quality measures: CKMM Revenue (Cohen-addad et al., 2019), MW Revenue (Moseley & Wang, 2017), Dasgupta Cost (Dasgupta, 2016) and MW Cost (a natural dissimilarity metric that, to our knowledge, has not been investigated before).

From the general hyperplane-based HC framework, we focus on two scalable algorithms: the **Random Cut Tree (RCT)** algorithm, originally proposed by Guha et al. (2016), and the **Uniform Radial Random Hyperplane** (**URRH**) algorithm, which is novel. We prove a number of new approximation guarantees for these two algorithms, including the following:

(i) For the CKMM Revenue, we prove that RCT (resp. URRH) has a *0.9-approximation ratio* when using the ℓ_1 (resp. ℓ_2) distances as dissimilarities, which improves on the 0.74 approximation ratio recently shown by Naumov et al. (2020) for a computationally more demanding algorithm;

(ii) For MW Revenue, when the similarity weights are defined through inverse ℓ_1-distances $1/||x-y||_1$ (resp. ℓ_2-distances) , we provide a *0.8-approximation ratio* for RCT (resp. URRH), while for the ℓ_2 Gaussian kernel similarity, URRH improves (Figure 2) on the approximation guarantee contained in (Charikar et al., 2019b);

(iii) When similarity weights are defined in terms of ℓ_2-distances in $\mathbb{R}^d$, we show that URRH achieves an approximation of $\frac{1}{3} + O(1/d^3)$ for the MW Revenue, yielding the first $> \frac{1}{3}$ approximation for non-constant d (in contrast to Charikar et al. (2019b) and Vainstein et al. (2021)).

(iv) For the MW Cost, we provide a *2-approximation ratio* for RCT (resp. URRH) in the case when dissimilarities are defined as ℓ_1-distances (resp. ℓ_2-distances).

We refer the reader to Table 2 in Section 4 for a summary of results on RCT as well as to Theorems 5.2 and 5.3 in Section 5 for the approximation guarantee of URRH.

Finally, we perform preliminary experiments on both synthetic and real-world datasets, where we compare RCT and URRH to known dynamic HC baselines. These experiments show that, in terms of approximation quality, our algorithms are on par with these baselines when the cluster separation in the data is moderate, tend to outperform the baselines in the presence of high level of noise (harder clustering instances), and vice versa for low noise levels.

Related work. Most of the existing hierarchical clustering solutions for streaming data are heuristics, e.g., Rodrigues et al. (2006); Loewenstein et al. (2008); Nguyen et al. (2014). The approach in Kobren et al. (2017) optimizes for a different quality measure, the so-called dendogram purity. In our experimental evaluation, we include three of the most popular previous approaches: BIRCH, PERCH and GRINCH.

Introduced by Zhang et al. (1996), BIRCH is a dynamic algorithm that maintains a hierarchical clustering tree such that every internal node contains the metadata corresponding to its subcluster (Clustering-Feature). PERCH (Kobren et al., 2017) is a dynamic clustering algorithm that performs rotations to enhance subtree purity and balance. GRINCH (Monath et al., 2019) is a dynamic clustering algorithm that employs two key operations, rotate and graft, which respectively handle local and global rearrangements.

Orthogonally, many theoretical results exist for the batch case, wherein the dataset is given up front in its entirety. This line of work may be divided into *general* instances and *metric-based* instances.

General weights. Paving the way, Dasgupta (2016) first framed the HC problem as an optimization problem. Currently the best known approximation to the Dasgupta Cost is achieved through iterative sparsest cut, yielding an approximation factor of $O(\sqrt{\log n})$ (Charikar & Chatziafratis, 2017; Cohen-addad et al., 2019). Furthermore, a constant approximation does not exist assuming the Small Set Expansion (SSE) Hypothesis (Charikar & Chatziafratis, 2017). Moseley & Wang (2017) introduced a maximization variant of the problem. Under this objective (MW Revenue), state of the art results include a 0.585 approximation factor (Alon et al., 2020). Dissimilarity information is considered in Cohen-addad et al. (2019) (CKMM Revenue). In this case, the best approximation is known to be 0.74 (Naumov et al., 2020). We note that both objectives (MW and CKMM) are APX-hard assuming the SSE Hypothesis.

Metric-based weights. The MW objective has also been studied in connection to metric-based instances. Charikar et al. (2019b) considered the case where the similarity weights are defined through a non-increasing function $g : \mathcal{R} \rightarrow [0, 1]$ applied to pairwise distances defined via a metric. Vainstein et al. (2021) showed that if g admits certain "nice properties" and the metric has constant doubling dimension then there exists a $1 - \epsilon$ approximation for any constant $\epsilon > 0$. We note however, that this algorithm's running time is double exponentially dependent on $\frac{1}{\epsilon}$. Furthermore, in order to handle data streams the tree must be computed from scratch at each new insertion. Thus, the algorithm is impractical in many real-world applications, especially in dynamic settings.

We note that these objectives have been researched in many more flavours: Structural constraints (Chatziafratis et al., 2018), HC through hyperbolic embeddings (Chami et al., 2020), and many others (Wang & Moseley, 2020; Charikar et al., 2019a; Chatziafratis et al., 2020a).

Finally, it is worth mentioning that there has been work on

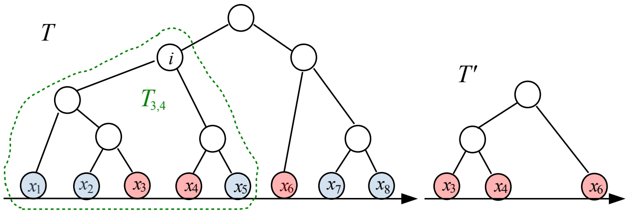

Figure 1: Left: A hierarchical clustering T on the set $X = \{x_1, \ldots, x_8\}$ made up of eight points laying on a line. Internal node i encodes the cluster $\{x_1, \ldots, x_5\} \subseteq X$. Tree $T_{1,4}$ is the subtree rooted at $\mathrm{lca}_T(x_1, x_4) = i$, with $|T_{1,4}| = 5$. Notice that $i = \mathrm{lca}_T(x_j, x_k)$ for $j = 1, 2, 3$, and $k = 4, 5$. Right: T' is the restriction of T to triplet $\{x_3, x_4, x_6\}$.

clustering of data streams when one is instead interested in optimizing flat clusters of the data. Most relevant to us is the work by Schmidt & Sohler (2019) on *hierarchical diameter k-clustering*. Here, the data are first hierarchically clustered and then each pointwise flat cluster (corresponding to a cut in the hierarchical clustering tree) is considered. The goal is to simultaneously minimize the median of each resulting flat clustering. Though operating in a dynamic setting, the resulting algorithms are of different flavor than ours due to the significant difference in objectives. Further work on flat clustering of data streams includes, e.g., Lin et al. (2010); Chen (2009).

2. Preliminaries and basic notation

In its standard formulation, in the HC problem we are given a set[1] of n items $X = \{x_1, \ldots, x_n\}$, and the goal is to construct a (binary) tree T whose leaves are the n items above so as to optimize some criterion. The tree encodes a clustering of X at different levels of granularity. Each internal node i of T can be naturally viewed as the cluster (that is, the subset of X) made up of all the leaves in the subtree rooted at i. Given leaves x_i and x_j of T, we denote by $T_{i,j}$ the subtree rooted at the lowest common ancestor $\mathrm{lca}_T(x_i, x_j)$ of x_i and x_j in T, while $|T_{i,j}|$ denotes the number of leaves in $T_{i,j}$. See Figure 1 (left) for a simple illustration.

Following the recent trend in the HC literature, we cast the problem as an optimization problem (e.g., (Dasgupta, 2016; Moseley & Wang, 2017; Wang & Wang, 2018; Cohen-addad et al., 2019; Alon et al., 2020; Charikar et al., 2019a;b; Chatziafratis et al., 2020b; Wang & Moseley, 2020; Naumov et al., 2020)), where an objective function is constructed that only depends on information about the pairwise *similarity* or pairwise *dissimilarity* over the points in X. Moreover, we assume the data are described by suitable feature vectors, so that the items in X lie within a suitably bounded subset of $\mathbb{R}^d$, for some input dimension $d \geq 1$, and the pairwise information is then a function of the feature vectors alone. This pairwise information may be encoded either through a *similarity* function $\mathrm{sim} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, for instance: $\mathrm{sim}(x_i, x_j) = x_i \cdot x_j$, the inner product between x_i and x_j; or $\mathrm{sim}(x_i, x_j) = \exp\left(-||x_i - x_j||_2^2/2\sigma^2\right)$, the Gaussian kernel between x_i and x_j at scale $\sigma > 0$; or $\mathrm{sim}(x_i, x_j) = D - ||x_i - x_j||$, where $||\cdot||$ is some norm over $\mathbb{R}^d$, and D is some notion of diameter of the set of points X; or through a *dissimilarity* function, $\mathrm{dissim} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, e.g., $\mathrm{dissim}(x_i, x_j) = ||x_i - x_j||$. A meaningful definition that connects $\mathrm{sim}(\cdot, \cdot)$ to $\mathrm{dissim}(\cdot, \cdot)$ is simply $\mathrm{sim}(x_i, x_j) = -\mathrm{dissim}(x_i, x_j)$.

Notation-wise, when the pairwise information represents similarity, we collectively denote it by a weight matrix $[w_{i,j}]_{i,j=1}^n$; when it represent dissimilarity we use instead matrix $[d_{i,j}]_{i,j=1}^n$. A number of objectives can then be defined, depending on whether we want to maximize similarity or minimize dissimilarity (see Table 1 for reference). The MW Revenue (Moseley & Wang, 2017) of tree T on similarity matrix $[w_{i,j}]$, denoted here as[2] $\mathrm{Rev}_S(T)$, is defined as

$$\mathrm{Rev}_S(T) = \sum_{i,j=1}^{n} w_{i,j}(n - |T_{i,j}|) ,$$

and the goal is to find a tree T such that the approximation ratio $\mathrm{Rev}_S(T)/\mathrm{Opt}_{\mathrm{Rev}_S}$ is as large as possible, where $\mathrm{Opt}_{\mathrm{Rev}_S} = \max_T \mathrm{Rev}_S(T)$ is the largest ("optimal") possible revenue any tree T can achieve on the given $[w_{i,j}]$. The other objectives (CKMM Revenue (Cohen-addad et al., 2019), Dasgupta Cost (Dasgupta, 2016), and MW Cost[3]) are defined analogously – please refer to Table 1 – and so are the corresponding optima and approximation ratios. For instance, when dealing with dissimilarity information and costs, we have $\mathrm{Opt}_{\mathrm{Cost}_D} = \min_T \mathrm{Cost}_D(T)$ and the goal is to find T so as to make the ratio $\mathrm{Cost}_D(T)/\mathrm{Opt}_{Cost_D}$ as small as possible.

Optimizing the above objectives exactly is know to be NP-hard (Dasgupta, 2016; Cohen-addad et al., 2019), hence the recent flurry of papers (e.g., (Dasgupta, 2016; Cohen-addad et al., 2019; Charikar et al., 2019a;b; Alon et al., 2020; Chatziafratis et al., 2020b; Naumov et al., 2020)) looking for fast approximation algorithms.

In this paper, we are specifically interested in HC algorithms having the *sequential property*, as defined next.

Definition 2.1. *Given a set of n items $X = \{x_1, \ldots, x_n\}$, denote by $T_i = A(\langle x_1, \ldots, x_i\rangle)$ the (random) output of a HC algorithm A having input $\{x_1, \ldots, x_i\}$, and let $T' = \mathrm{Ins}(T, x)$ denote a (randomized) insertion operation*

[1] This set may actually contain repeated items.

[2] In these notations, we leave the dependence on $[w_{i,j}]$ or $[d_{i,j}]$ implicit, no ambiguity will arise.

[3] This is a natural dissimilarity metric, although it does not seem to have been investigated previously.

Objective	Type	Name	Symbol
$\max \sum_{i,j} w_{ij}(n - \lvert T_{ij}\rvert)$	Sim. Rev.	MW Rev.	Rev_S
$\max \sum_{i,j} d_{ij}\lvert T_{ij}\rvert$	Diss. Rev.	CKMM Rev.	Rev_D
$\min \sum_{i,j} w_{ij}\lvert T_{ij}\rvert$	Sim. Cost	Dasgupta Cost	Cost_S
$\min \sum_{i,j} d_{ij}(n - \lvert T_{ij}\rvert)$	Diss. Cost	MW Cost	Cost_D

Table 1: HC objectives. In the above, we abbreviated "similarity" by "Sim.", "Dissimilarity" by "Diss., and "Revenue" by "Rev". The type of objective refers to whether it deals with similarity or dissimilarity information and that the goal is to maximize (revenue) or minimize (cost).

that adds a new leaf x to tree T, producing in output the augmented tree T'. We say that A has the sequential property *w.r.t.* Ins *if, for all item sets X, and all $i = 1, \ldots, n$, the random variable T_i has the same distribution over trees as the random variable* $\mathrm{Ins}(T_{i-1}, x_i)$*, where T_0 is the empty tree.*

In other words, A has the sequential property if it admits an exogenous insertion procedure Ins such that building the tree incrementally by inserting one leaf after the other through Ins yields the same statistical properties as if the tree were constructed by A looking at all data in batch. Hence, if the tree constructed by A has approximation ratio r (in expectation over the internal randomization of A) then so does the tree incrementally constructed by Ins (in expectation over the internal randomization of Ins). Also, observe that this equivalence holds *independent* of the order in which Ins processes the n items.

In the next section (Section 3), we develop a general framework for *hyperplane-based* hierarchical clustering which encompasses a family of dynamic algorithms for that task. Then, in the two subsequent sections, we shall describe and analyze two members of this family. The first one (Random Cut Tree Algorithm, Section 4) operates with axis-aligned hyperplanes, and is suited to the ℓ_1-based objectives contained in Table 1, on a variety of definitions for $w_{i,j}$ and $d_{i,j}$. The second one (Uniform Radial Random Hyperplane Algorithm, Section 5) operates with general hyperplanes, and is suited to the analogous ℓ_2-based objectives (Theorem 5.2). In addition, it enjoys an unconditional approximation guarantee (Theorem 5.3) for the weights defined by case 1 of Table 1. For both algorithms, we show that they can (i) be implemented efficiently, (ii) cater to ℓ_1-based and ℓ_2-based geometry respectively, and (iii) enjoy good approximation guarantees.

3. Hyperplane-based hierarchical clustering

Let $\mathrm{Graff}_{d-1}(\mathbb{R}^d)$ be the manifold of all $(d-1)$-dimensional affine subspaces (that is, hyperplanes) of $\mathbb{R}^d$. Let μ be a nonnegative measure on $\mathrm{Graff}_{d-1}(\mathbb{R}^d)$ that is finite on compact sets. With such a μ, we associate a HC algorithm A_μ, as described next.

Objective	Metric (L1)	Approx.	Random
1. MW Rev*	L1-similarity	0.73	5/9
2. Dasgupta Cost*	L1-similarity	2	∞
3. CKMM Rev	L1-distance	0.90	2/3
4. MW Cost	L1-distance	2	∞
5. MW Rev	Inverse distance	0.80	1/3
6. Dasgupta Cost	Inverse distance	1.5	∞
7. MW Rev	Gauss. Kernel	Figure 2	$\frac{1+2\delta}{3}$
8. MW Rev	Abs. Exp. Kernel	Figure 2	$\frac{1+2\delta}{3}$

Table 2: RCT approximation guarantees for different objectives and metrics. All metrics are $L1$-based. Only the first two cases (*) require Assumption 4.4. The last two cases assume that the weights are in $[\delta, 1]$, for some $\delta \in (0, 1]$. The last column is the approximation achieved by the baseline RANDOM that returns a binary tree on the leaves, which is chosen uniformly at random.

Given finite $X \subset \mathbb{R}^d$ as input, denote by $\mathrm{Conv}(X)$ the convex hull of X. Then the set $\mathcal{H}_X$ of hyperplanes of $\mathrm{Graff}_{d-1}(\mathbb{R}^d)$ that intersect $\mathrm{Conv}(X)$ is compact, and hence $\mu(\mathcal{H}_X) < \infty$. Let $\mu_X = \mu/\mu(\mathcal{H}_X)$ be the probability measure induced on $\mathcal{H}_X$ by restricting to $\mathcal{H}_X$ and normalizing to 1. On input X, algorithm A_μ:

1. Samples a random hyperplane $H_X \sim \mu_X$;
2. Partitions X into Y and Z according to H_X (points lying on H_X can be arbitrarily assigned to either Y or Z).
3. Recurses on Y and Z using the probability measures μ_Y and μ_Z, respectively.

Applying the above until we arrive at singleton sets, we construct a (random) binary tree T with leaves the points in X, based on the partitions induced by the sampled hyperplanes. Such T is the output of A_μ on input X. The key observation now is that for any set of points Y' with $\mathrm{Conv}(Y) \subseteq \mathrm{Conv}(Y')$, we have $\mathcal{H}_Y \subseteq \mathcal{H}_{Y'}$, and thus $\mu_Y = \mu_{Y'}|_Y$, the probability measure of $\mu_{Y'}$ conditioned on Y. Thus we may rephrase Step 3 above as rejection-sampling from $\mu_{Y'}$ conditioned on the sampled hyperplane intersecting $\mathrm{Conv}(Y)$ (resp. $\mathrm{Conv}(Z)$).

If we have a way of sampling efficiently from the hyperplane probability measures, the main property of algorithm A_μ is that it leads to a natural algorithm for HC with the sequential property.[4]

Theorem 3.1. *Let μ be a nonnegative measure on* $\mathrm{Graff}_{d-1}(\mathbb{R}^d)$ *which is finite on compact sets, and suppose there is an efficient way to sample from μ_X for all finite sets X. Then, there is an efficient insertion operation* Ins_μ *such that A_μ has the sequential property w.r.t.* Ins_μ.

In particular, recall that this means that the (random) tree generated by Ins_μ is independent of the order in which Ins processes the inserted items. The general pseudocode for

[4] All proofs are given in the supplementary material.

Ins_μ is given in Appendix A. In the following sections, we specify particular measures μ from which hyperplanes can be efficiently sampled and which additionally give rise to HC algorithms having the sequential property, and exhibiting good approximation ratios for the metrics of Section 2. The associated insertion operations are presented in the corresponding sections of the appendix.

Remark 3.2. *It is important to stress that the above algorithm, as well as its by-products in later sections, do not take as input the pairwise information encoded by $[w_{i,j}]$ or $[d_{i,j}]$. These algorithms are purely geometric algorithms that will exhibit strong approximation properties, provided the pairwise information we use at evaluation time to compute the metrics in Table 1 is reasonably aligned with the geometry these algorithms rely upon. Further examples of this sort are the Projected Random Cut algorithm in (Charikar et al., 2019b), as well as the dynamic algorithms we compare to in our experimental investigation (Section 6).*

4. Random Cut Tree approximation

In this section we discuss a special case of hyperplane-based clustering known as the Random Cut Tree (RCT) which has been introduced by Guha et al. (2016) in the context of anomaly detection. We provide approximation results for related similarity and dissimilarity objectives (from Table 1). In the case of dissimilarity objectives, we use the distances themselves as the dissimilarity measure.

An RCT (batch algorithm) $T(X)$ on item set $X \subseteq \mathbb{R}^d$ is a tree-valued random variable generated as follows:

- Draw random index $I \in [d]$ with probability $\mathbb{P}[I = i] = \frac{l_i}{\sum_{i=1}^d l_i}$, where
$$l_i = \max_{x\in X}(x)_i - \min_{x\in X}(x)_i \ ,$$
with $(x)_i$ denoting the i-th component of vector x. Hence the above probability is proportional to the side lengths of the (axis-parallel minimum) bounding box of X;
- Draw threshold θ uniformly at random in the interval $[\min_{x\in X} x_I, \max_{x\in X} x_I]$;
- Let $X_1 = \{x \,|\, x \in X, (x)_I \leq \theta\}$ and $X_2 = X \backslash X_1$ correspond to the left and right subtrees of the root of $T(X)$, and recurse on X_1 and X_2, until $T(X)$ is a (singleton) leaf.

We have the following characterization of RCT:

Fact 4.1. *Fix dimension d, and let $H_{i,v} = \{x \in \mathbb{R}^d \,|\, x_i = v\}$, where x_i is the i-th component of vector x. Let then $\mathcal{H} = \{H_{i,v} \,|\, i \in [d], v \in \mathbb{R}\}$ be the set of axis-parallel hyperplanes. For $\mathcal{H}' \subset \mathcal{H}$, define μ_{RCT} by $\mu_{\mathrm{RCT}}(\mathcal{H}') = \sum_{i=1}^d \mu_L(\{v \in \mathbb{R} \,|\, H_{i,v} \in \mathcal{H}'\})$, where L is the standard Lebesgue measure on $\mathbb{R}$. Then $A_{\mu_{\mathrm{RCT}}}$ (resp. $\mathrm{Ins}_{\mu_{\mathrm{RCT}}}$) is the offline (resp. dynamic) RCT algorithm.*

In (Guha et al., 2016), it is shown (Theorem 3 therein) that an RCT can be maintained over a set of points X that is dynamically updated with streaming data in sub-linear update time and $O(dn)$ space. The pseudocode for the insertion operation (adapted from (Guha et al., 2016)) is given in Appendix B.

The analysis of RCT with respect to the HC objectives in Table 1 rests on an important restriction property that this algorithm enjoys.

Definition 4.2. *Given tree T on the set of leaves X, and $R \subseteq X$, the* restriction *of T to R is the tree obtained by deleting the leaves of T in $X \setminus R$ (along with their edges), and contracting edges to obtain a binary tree whose leaves are identified with R. In particular, if R is a triplet $R = \{x_i, x_j, x_k\}$, the restriction of T to R when $\mathrm{lca}_T(x_i, x_j)$ is a descendant of $\mathrm{lca}_T(x_i, x_k)$ is the tree where x_i, x_j are siblings, and x_k is a sibling of their parent (and similarly for the other cases). See Figure 1 (right) for an illustration.*

Lemma 4.3. *Let $X \subseteq \mathbb{R}^d$ be a set of items. For any $R \subseteq X$, the restriction of the RCT $T(X)$ (that is, the output of RCT on input X) to subset R has the same distribution as $T(R)$.*

In fact all algorithms from the family A_μ enjoy this property (see the supplementary material for a proof). We will use this result in the particular case of R being a generic triplet $\{x_i, x_j, x_k\}$.

RCT as characterized in Fact 4.1 can be seen as naturally operating in an ℓ_1 geometry. We now introduce a necessary assumption in order to obtain competitive approximation guarantees for RCT in the case of similarity-based objectives (MW Revenue and Dasgupta Cost) for the ℓ_1 similarity measure $w_{i,j} = D - d_{i,j}$, where $d_{i,j} = ||x_i - x_j||_1$ and $D = \max_{i,j} d_{i,j}$. As we shall see momentarily, this assumption will not be required by dissimilarity-based objectives (CKMM Revenue and MW Cost).

Assumption 4.4. *We assume $\binom{n}{3}^{-1} \sum_{i<j<k}(d_{i,j} + d_{i,k} + d_{j,k})/2 \leq D = \max_{i,j} d_{i,j}$. Observe that we have $\max_{i,j,k}(d_{i,j} + d_{i,k} + d_{j,k})/2 \leq \frac{3D}{2}$ always, so this also follows under the modified similarity $w_{i,j} = \frac{3}{2}D - d_{i,j}$. The weights are now in the range $[\frac{D}{2}, \frac{3D}{2}]$, and in this case,* RANDOM *gives a baseline revenue approximation of $\frac{3D/2+D/2+D/2}{3(3D/2)} = \frac{5}{9}$.*

The reason for Assumption 4.4 is the following. RCT is a geometric algorithm whose cuts of triplets $\{x_i, x_j, x_k\}$ depend on the distances $d_{i,j}$, $d_{i,k}$, and $d_{j,k}$. Allowing the similarity weights $w_{i,j}$ to have ratios substantially different from the corresponding ratios of the $d_{i,j}$'s can lead to adversarial situations, as we illustrate next.

Example 4.5. *Let $V \subset \mathbb{R}^3$ consist of the points $x_1 = (1+\epsilon, 0, 0)$, $x_2 = (0,1,0)$, and $x_3 = (0,0,1)$. We have $D = d_{1,2} = d_{1,3} = 2+\epsilon$, $d_{2,3} = 2$, $w_{1,2} = w_{1,3} = 0$, and $w_{2,3} = \epsilon$. Thus $\mathrm{Rev}_{\mathrm{Opt}_S} = \epsilon$. On the other hand, being based on the ℓ_1 geometry,* RCT *makes the cuts with approximately equal probabilities, which leads to an approximation ratio of $1/3 + O(\epsilon)$, that is, very close to the trivial approximation ratio of 1/3 achieved on Rev_S by a* random *binary tree (Moseley & Wang, 2017).*

Theorem 4.6. RCT *satisfies the approximation guarantees for the combination of objectives and metrics listed in Table 2. In detail, for each combination of revenue (resp. cost) objective* Obj*, metric m, and approximation factor α in Table 2, we have the approximation guarantee that for all $X \subseteq \mathbb{R}^d$ endowed with metric m, $\mathbb{E}[\mathrm{Obj}(\mathrm{RCT}(X))] \geq \alpha \mathrm{Opt}_{\mathrm{Obj}}(X)$ (resp. $\leq$), where the expectation is over the internal randomization of* RCT*, and X satisfies Assumption 4.4 in the first two cases.*

While Theorem 4.6 covers a diverse range of objectives and metrics, the proof technique is similar. We sketch the main idea in the case of MW Revenue with ℓ_1-similarity.

The following length-proportional cut property of the RCT algorithm is a main ingredient of our approximation results.

Lemma 4.7. *Given input X and a cut H_X sampled from μ_X, the probability $p_{i,j}$ that x_i and x_j are split by H is proportional to their ℓ_1 distance $d_{i,j}$.*

A consequence of Lemma 4.7 which we need for the proof of Theorem 4.6 is the following lemma.

Lemma 4.8. *Fix a triplet $\{x_i, x_j, x_k\}$ of X. Then the probability, $p_{i,j|k}$, that* RCT *$T(X)$ separates x_i and x_j from x_k is given by*

$$p_{i,j|k} = \frac{d_{i,k} + d_{j,k} - d_{i,j}}{d_{i,j} + d_{i,k} + d_{j,k}} \, ,$$

and similarly for $p_{i,k|j}$ and $p_{j,k|i}$.

We use below the cyclic sum notation $\sum_{\mathrm{cyc}} f(i,j,k) = f(i,j,k) + f(j,k,i) + f(k,i,j)$. For a tree T, and a triplet of leaves i,j,k, we write $ij|k$ to mean that $\mathrm{lca}_T(i,j)$ is a descendant of $\mathrm{lca}_T(i,k)$.

Proof of Theorem 4.6– sketch. Fix input $X = \{x_1, \ldots, x_n\}$. Given a tree T on X, note that we can rewrite the MW Revenue $\sum_{i,j} w_{i,j}(n - |T_{i,j}|)$ as the triplet-wise sum $\sum_{i<j<k} \mathrm{Rev}_{i,j,k}(T)$, where

$$\mathrm{Rev}_{i,j,k}(T) = \begin{cases} w_{i,j} & \text{if } ij|k \text{ in } T \\ w_{i,k} & \text{if } ik|j \text{ in } T \\ w_{j,k} & \text{if } jk|i \text{ in } T \, . \end{cases} \tag{1}$$

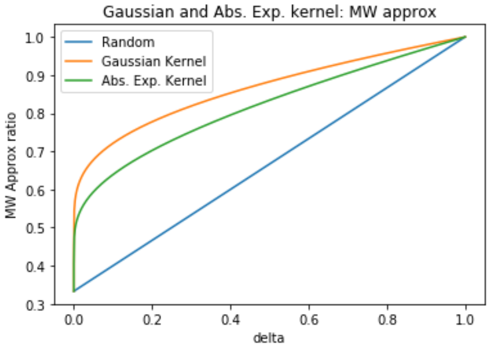

Figure 2: Approximation ratio for absolute exponential kernel and Gaussian kernel with weights $w_{i,j} \in [\delta, 1]$. RCT (resp. URRH) satisfy these guarantees with the ℓ_1 (resp. ℓ_2) distances in these kernels.

Now, from Lemma 4.8, for tree $T_{\mathrm{RCT}}(X)$ computed by RCT on X we can write

$$\mathbb{E}[\mathrm{Rev}_{i,j,k}(T(X))] = \sum_{\mathrm{cyc}} \frac{d_{i,k} + d_{j,k} - d_{i,j}}{d_{i,j} + d_{i,k} + d_{j,k}} (D - d_{i,j}) \, .$$

On the other hand, we also have the upper bound $\mathrm{Rev}_{i,j,k}(T_{\mathrm{Opt}}(X)) \leq \max\{D - d_{i,j}, D - d_{i,k}, D - d_{j,k}\}$, where $T_{\mathrm{Opt}}(X)$ is the optimal tree on X. It now remains to use these expressions to prove the required approximation bound, which reduces to a tractable optimization problem. □

We have shown that RCT enjoys good approximation guarantees for ℓ_1-based measures. In the following section, we introduce a new algorithm URRH that matches RCT's approximation guarantees for ℓ_2-based similarity measures. To conclude this section, we introduce a simple extension of RCT that achieves weaker guarantees than URRH, but is of theoretical interest. Namely, we define the Projected Random Cut Tree (PRCT) algorithm as follows:

Definition 4.9. *Given input X and projection dimension k,* PRCT(k) *applies* RCT *to $(Px_i)_{i=1}^n$, where P is a $k \times d$ Gaussian projection matrix with i.i.d. entries.*

We remark that when $k = 1$, PRCT reduces to the projected random cut algorithm from (Charikar et al., 2019b).

We have the following guarantees for PRCT:

Theorem 4.10. *Fix $\epsilon, \delta > 0$. Consider a revenue case from Table 2 (cases 1, 3, 5, 7, 8) but with the corresponding ℓ_2 metric. Let α be the corresponding approximation guarantee of* RCT *under ℓ_1 metric. Then there exists an absolute constant c such that with probability $1-\delta$,* PRCT *with $k = c\log(n/\delta)/\epsilon^2$ achieves an expected approximation of $\alpha - \epsilon$ in the ℓ_2 metric.*

Theorem 4.11. *Fix $\epsilon, \delta > 0$. Consider a cost case from Table 2 (cases 2, 4, 6) but with the corresponding ℓ_2 metric. Let α be the corresponding approximation guarantee of* RCT *under ℓ_1 metric and assume that the weights lie in the range $[\gamma, 1]$ for arbitrarily small but positive γ. Then there exists an absolute constant c such that with probability $1-\delta$,* PRCT *with $k = c\log(n/\delta)/\epsilon^2$ achieves an expected approximation of $\alpha + \epsilon$ in the ℓ_2 metric.*

5. Uniform radial random hyperplane approximation

The second hyperplane-based HC algorithm we present is the URRH (Uniform Radial Random Hyperplane) Algorithm.

At each recursive step, URRH takes as input a subset $C \subseteq X$ of the input items X, and *any* $(d-1)$-sphere $\mathcal{S}(C)$ containing all points of C. The algorithm randomly cuts $\mathcal{S}(C)$ to split C into C' and C''. Whenever the cut makes either C' or C'' empty, the hyperplane is rejected, and a new cut is drawn until $C', C'' \neq \emptyset$. Finally, as in RCT, the URRH algorithm recurses on C' and C'' until the input becomes a singleton.

The details (pseudocode) of URRH are given in Appendix C. Below we give an idea of the key steps.

Each random hyperplane is selected through a two-step process: (i) A direction in $\mathbb{R}^d$ is selected by choosing a unit vector p uniformly at random, and (ii) a hyperplane orthogonal to p is selected among those intersecting $\mathcal{S}(C)$. More precisely, at each recursive step, URRH operates as follows:

- Direction p is selected uniformly at random from $\mathcal{S}^{d-1}$, the unit $(d-1)$-sphere;
- Let $\mathcal{S}(C)$ be *any* $(d-1)$-sphere containing all items in C (e.g., $\mathcal{S}(C)$ is the *circumsphere* of $\mathrm{Conv}(C)$).[5] Let r and c be the radius and the center of $\mathcal{S}(C)$. Hyperplane $H_{p,b}(\mathcal{S}(C)) := \{x \in \mathbb{R}^d \,:\, x \cdot p = b\}$ is generated by drawing b uniformly at random from the interval $[c \cdot p - r,\, c \cdot p + r]$.
- If $H_{p,b}(\mathcal{S}(C))$ cuts C, i.e., it splits C into C' and C'' such that $C', C'' \neq \emptyset$, then we recurse on C' and C'', otherwise we *reject* $H_{p,b}(\mathcal{S}(C))$ and generate it again (by re-drawing p and b)).

From the above, a clear computational trade-off emerges between calculating a $(d-1)$-sphere $\mathcal{S}(C)$ having small radius, and the number of hyperplanes that get rejected. As mentioned in Appendix C, there are several strategies to resolve this trade-off. For now, we just anticipate that, for any $d \in \mathbb{N}$ and any C, the probability that $H_{p,b}(\mathcal{S}(C))$ is *not* rejected is at least $\frac{c}{\sqrt{d}}$ for a constant c, and decreases linearly in the radius of the sphere $\mathcal{S}$ currently used by URRH (see the formal statement in Lemma C.5 in Appendix C).

We have the following characterization of URRH as a member of the general hyperplane-based family.

Fact 5.1. *Fix dimension d, let $H_{u,v} = \{x \in \mathbb{R}^d \mid x \cdot u = v\}$, and $\mathcal{H} = \{H_{u,v} \mid u \in \mathcal{S}^{d-1}, v \in \mathbb{R}\}$ be the set of all hyperplanes in $\mathbb{R}^d$. Define $\mu_{\text{URRH}}(\mathcal{H}') = \int_{u \in \mathcal{S}^{d-1}} \mu_L(\{v \in \mathbb{R} \mid H_{u,v} \in \mathcal{H}'\}) d\nu$ for $\mathcal{H}' \subset \mathcal{H}$, where μ_L is the Lebesgue measure on $\mathbb{R}$ and ν is the uniform measure on $\mathcal{S}^{d-1}$. Then $A_{\mu_{\text{URRH}}}$ (resp. $\mathrm{Ins}_{\mu_{\text{URRH}}}$) is the offline (resp. dynamic)* URRH *algorithm.*

In fact, URRH satisfies Lemma 4.7 with $d_{i,j}$ now representing ℓ_2 distances. The same proof machinery for Theorem 4.6 thus applies to URRH for the case that all measures are ℓ_2-based, and we have the following result.

Theorem 5.2. URRH *satisfies the same approximation guarantees as* RCT *given in Theorem 4.6, with ℓ_2-based measures under the equivalent ℓ_2 analog of Assumption 4.4 for the first two cases.*

We also have the following unconditional approximation ratio guarantees, for the case of Rev_S and similarity weights defined as $w_{i,j} := D - d_{i,j}$, where $d_{i,j}$ is the Euclidean distance between x_i and x_j, and $D = \max_{1 \leq i < j \leq n} d_{i,j}$ is the maximal distance over all pairs of points in X. The result states that the MW Revenue of URRH is strictly larger than the trivial $\frac{1}{3}$ approximation ratio[6] for any input dimension $d > 3$, whenever n is not too small w.r.t. to d.

Theorem 5.3. *Given any input set $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$, with $d > 3$, the approximation ratio $\frac{\mathbb{E}[\mathrm{Rev}_S(\text{URRH}(X))]}{\mathrm{Opt}_{\mathrm{Rev}_S}}$ is lower bounded by $\frac{1}{3} + g(d,n)$, where $g(d,n)$ is a function of d and n such that $g(d,n) > 0$ for all $n > \frac{605}{116}d \approx 5.22d$. In particular, if $n \geq \left(9 + \frac{38}{d-3.98}\right)d$ and $d > 3$, we have*

$$\mathbb{E}[\mathrm{Rev}_S(\text{URRH}(X))] \geq \left(\frac{1}{3} + \frac{1}{31d^3}\right)\mathrm{Opt}_{\mathrm{Rev}_S}.$$

In the above, the expectation is over the internal randomization of URRH.

Notice that condition $d > 3$ does not really limit the scope of Theorem 5.3, since one can always pad the input vectors with dummy components that do not alter pairwise distances, so as to force $d \geq 4$. Moreover, it is also worth observing that for all $d < 8$, the requirement $n \geq \left(9 + \frac{38}{d-3.98}\right)d$

[5] That is, the smallest sphere enclosing all the points of $\mathrm{Conv}(X)$.

[6] Recall that approximation ratio 1/3 can be trivially achieved in expectation by a randomly generated tree (Moseley & Wang, 2017).

becomes less stringent if one pads the input so as to force $d = 8$. This implies that $148 = \left\lceil \left(9 + \frac{38}{8-3.98}\right) 8 \right\rceil$ input points are always sufficient when $d \leq 8$. Finally, the special case $d = 1$ can be treated separately (see Theorem D.1 in the appendix) obtaining an expected MW revenue larger than the one of Theorem 5.3.

As mentioned in Fact 5.1, URRH has the sequential property (Definition 2.1). The pseudocode of the insertion procedure is detailed in Algorithm 5 in Appendix C.

6. Experiments

In this section, we demonstrate experimentally that RCT and URRH perform competitively compared to other well-known dynamic HC algorithms. In particular, we compare to BIRCH (Zhang et al., 1996), PERCH (Kobren et al., 2017), and GRINCH (Monath et al., 2019). Additionally, we compare all algorithms to the RANDOM baseline that builds a tree at random, and to PROJECTED RANDOM CUT (Charikar et al., 2019b) on a line, which can be made dynamic by applying RCT to the projected dataset. The objectives considered are MW Revenue, MW Cost, CKMM Revenue and Dasgupta Cost.

We evaluate these algorithms on both synthetic and real-world datasets. For synthetic datasets, we evaluate the performance of these algorithms in both noisy and well-separated settings. In particular, we draw 10K examples from standard Gaussians in $\mathbb{R}^2$. In the noisy setting, we draw from one Gaussian, and in the well-separated setting, we draw from two Gaussians with centers separated by four standard deviations in one direction. We denote these datasets by OneG, resp. TwoG. For real-world datasets, we compare the algorithms on the following data of varying scale: MNIST, ALOI (Geusebroek et al. (2005)), and ILSVRC12 (Deng et al. (2009)) trained with ResNet34 architecture. We note that when considering our (four) objectives, the resulting trees must be binary. All algorithms other than BIRCH output binary trees and thus do not need to be modified. In order to handle BIRCH, we follow the methodology of (Naumov et al., 2020) and simply assign the value of a random partitioning to all data point triplets that share the same lca in the tree. For an extended explanation see (Naumov et al., 2020), Appendix B.1 therein. For hardware, we used machines with a maximum of 125GB of RAM and 16 CPUs.

Methodology. For each of these experiments, we randomly permute the datasets and stream each one in the preprocessed order consistently across algorithms. We evaluate each of the aforementioned measures on the produced hierarchies, as follows: We sample 10K triplets $\mathcal{T}'$ from each dataset, then compute the measures restricted to these triplets. For the MW Revenue, this is $\sum_{(i,j,k)\in\mathcal{T}'} \mathrm{Rev}_{i,j,k}(T)$ (see Equation 1). We also report the measures for RANDOM: $\sum_{(i,j,k)\in\mathcal{T}'}(w_{i,j} + w_{j,k} + w_{i,k})/3$ and an upper (resp. lower) bound for the optimal revenue (resp. cost): $\sum_{(i,j,k)\in\mathcal{T}'} \max$ (resp. $\min)(w_{i,j}, w_{j,k}, w_{i,k})$ (see (Naumov et al., 2020)).[7] Finally, for RCT, URRH and PROJECTED RANDOM CUT, the output trees are non-deterministic, so we report the average over 10 different runs.

Table 3 compares MW Revenue using RBF kernel similarity $\mathrm{RBF}(x,y) = e^{-\|x-y\|_2^2/2\sigma^2}$ across all algorithms. Note that this is a function of the ℓ_2 distance, which we have chosen for uniform comparison across all algorithms. We choose σ as the mean ℓ_2 distance between pairs of points. This is to ensure a reasonable distribution of similarity weights. We defer the results for MW Cost, CKMM Revenue and Dasgupta Cost to the appendix, but note that they show similar trends. The following conclusions can be drawn:

(1) RCT and URRH achieve the highest revenue for OneG, a noisy setting in which there is no obvious way to split the data into two clusters at the root level. By contrast, they are outperformed by BIRCH, PERCH and GRINCH on TwoG, where the two clusters are well separated. We believe this can be explained by the fact that the baseline approaches rely heavily on clusters and/or nearest neighbor information to build the trees. On the other hand, RCT and URRH split the data by random cuts that are less sensitive to local data densities. Thus, the baselines take advantage of well-separated datasets, while RCT and URRH are more robust on noisy data.

(2) BIRCH, PERCH and GRINCH perform reasonably well for these objectives even though they are not explicitly designed to do so. We offer two reasons for this. First, in many of our experiments, RANDOM turned out to perform reasonably (and sometimes surprisingly) well; a similar phenomenon has been experimentally observed in (Naumov et al., 2020). When this happens, there is not much room for improvement across the various algorithms. Second, in order to ensure a fair comparison, in our experiments each dataset was randomly shuffled. This does not affect RCT and URRH, but it might have potentially eliminated unfavorable orderings of data for BIRCH, PERCH and GRINCH, thereby giving these competitors some advantage.

(3) RCT and URRH perform competitively compared to all other algorithms on the real-world datasets we tested, where clusters are moderately well-separated. Unlike BIRCH, PERCH and GRINCH, the practical relevance of RCT and URRH is complemented by their approximation

[7] These metrics are computed by sampling triplets, since exact computation would be unwieldy. The extra variance generated in the results turns out to be negligible.

	MNIST	ILSVRC12	ALOI	OneG	TwoG
RCT	0.93±0.01	0.94±0.0	0.91±0.01	0.9±0.01	0.9±0.06
URRH	0.93±0.0	0.94±0.0	0.9±0.01	0.9±0.01	0.9±0.03
BIRCH	0.93	0.94	0.91	0.87	0.98
PERCH	0.92	0.94	0.91	0.87	0.90
GRINCH	0.93	0.93	0.89	0.88	0.97
PROJECTED RANDOM CUT	0.92±0.0	0.94±0.0	0.88±0.01	0.87±0.0	0.86±0.07
RANDOM	0.92	0.93	0.85	0.74	0.71
UPPER BOUND	1.0	1.0	1.0	1.0	1.0

Table 3: MW Revenue approximation factors using RBF kernel similarity; ↑ is better. Each revenue is shown as a percentage of the corresponding upper bound for that dataset.

guarantees.

7. Conclusions and ongoing activity

We have introduced the general framework of hyperplane-based HC for data streams in metric spaces. We have given a general family of algorithms indexed by a sampling probability over hyperplanes. Each algorithm in this family admits two formulations, batch and sequential, whose (statistical) equivalence ensures a desirable robustness to data arrival order. We have studied two fast HC algorithms originating from this general family, and provided a number of approximation guarantees w.r.t. known objective functions, some of which improve on the available literature on HC. In addition, the algorithms are simple to implement, requiring only the selection of a splitting hyperplane in each node. New points are inserted as siblings of existing nodes, without the need to perform other changes in the tree structure.

We have run initial experiments on synthetic and real-world metric data, where the trend that seems to emerge is that our randomized algorithms are on par with celebrated dynamic HC baselines in the presence of moderate noise levels, tend to outperform these baselines with higher noise rate and be outperformed in the opposite case of clear cluster separation.

An interesting research direction is to generalize the family of hyperplane-based HC to richer separation classes, which would give us higher flexibility, while still retaining the crucial benefits of the dynamic solutions.

We conclude by mentioning a couple of additional results we obtained for the MW Revenue maximization problem on one-dimensional data, with weights $w_{i,j} := D - d_{i,j}$, and $n \to \infty$. We proved (see Appendix D) that the approximation ratio of RCT is at least 0.8303. We also developed two *very* fast deterministic algorithms for the batch setting, achieving approximation ratios of $\frac{3}{4}$ and $\frac{1}{2}$, respectively. Interestingly enough, the latter is always obtained by simply building a caterpillar tree, and has also a $\frac{3}{4}$-approximation ratio for the CKMM Revenue.

References

Ailon, N. and Chazelle, B. The fast johnson-lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, 39(1), 2009.

Alon, N., Azar, Y., and Vainstein, D. Hierarchical clustering: A 0.585 revenue approximation. In *Proceedings of Thirty Third Conference on Learning Theory*, volume 125, pp. 153–162. PKLR, 2020.

Chami, I., Gu, A., Chatziafratis, V., and Ré, C. From trees to continuous embeddings and back: Hyperbolic hierarchical clustering. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

Charikar, M. and Chatziafratis, V. Approximate hierarchical clustering via sparsest cut and spreading metrics. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA)*, pp. 841–854, 2017.

Charikar, M., Chatziafratis, V., and Niazadeh, R. Hierarchical clustering better than average-linkage. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2291–2304. Society for Industrial and Applied Mathematics, 2019a.

Charikar, M., Chatziafratis, V., Niazadeh, R., and Yaroslavtsev, G. Hierarchical clustering for euclidean data. In *Proceedings of Machine Learning Research*, volume 89, pp. 2721–2730. PMLR, 2019b.

Chatziafratis, V., Niazadeh, R., and Charikar, M. Hierarchical clustering with structural constraints. In *Proceedings of the 35th International Conference on Machine Learning, (ICML 2018)*, pp. 773–782, 2018.

Chatziafratis, V., Gupta, N., and Lee, E. Inapproximability for local correlation clustering and dissimilarity hierarchical clustering. *CoRR*, abs/2010.01459, 2020a.

Chatziafratis, V., Yaroslavtsev, G., Lee, E., Makarychev, K., Ahmadian, S., Epasto, A., and Mahdian, M. Bisect and conquer: Hierarchical clustering via max-uncut bisection. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108, pp. 3121–3132. PMLR, 2020b.

Chen, K. On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications. *SIAM J. Comput.*, 39(3):923–947, 2009. doi: 10.1137/070699007. URL https://doi.org/10.1137/070699007.

Cohen-addad, V., Kanade, V., Mallmann-trenn, F., and Mathieu, C. Hierarchical clustering: Objective functions and algorithms. *J. ACM*, 66:4, 2019.

Dasgupta, S. A cost function for similarity-based hierarchical clustering. In *Proceedings of the 48th annual ACM symposium on Theory of Computing*, pp. 118127, 2016.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. Cluster analysis and display of genomewide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, 1998.

Geusebroek, J.-M., Burghouts, G. J., and Smeulders, A. W. The amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112, 2005.

Gilbert, F., Simonetto, P., Zaidi, F., Jourdan, F., and Bourqui, R. Communities and hierarchical structures in dynamic social networks: Analysis and visualization. *Social Network Analysis and Mining*, 1(2), 2011.

Guha, S., Mishra, N., Roy, G., and Schrijvers, O. Robust random cut forest based anomaly detection on streams. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pp. 2712–2721. PMLR, 2016.

Kobren, A., Monath, N., Krishnamurthy, A., and McCallum, A. A hierarchical algorithm for extreme clustering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 255–264. Association for Computing Machinery, 2017.

Li, S. Concise formulas for the area and volume of a hyperspherical cap. *Asian Journal of Mathematics and Statistics*, 4(1):66–70, 2011.

Lin, G., Nagarajan, C., Rajaraman, R., and Williamson, D. P. A general approach for incremental approximation and hierarchical clustering. *SIAM J. Comput.*, 39(8):3633–3669, 2010. doi: 10.1137/070698257. URL https://doi.org/10.1137/070698257.

Loewenstein, Y., Portugaly, E., Fromer, M., and Linial, M. Efficient algorithms for accurate hierarchical clustering of huge datasets: tackling the entire protein space. In *Proceedings 16th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pp. 41–49, 2008.

Manning, C. D., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

Monath, N., Kobren, A., Krishnamurthy, A., Glass, M. R., and McCallum, A. Scalable hierarchical clustering with tree grafting. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1438–1448. Association for Computing Machinery, 2019.

Moseley, B. and Wang, J. Approximation bounds for hierarchical clustering: Average linkage, bisecting k-means, and local search. In *Advances in Neural Information Processing Systems*, volume 30, pp. 30943103. Curran Associates, Inc., 2017.

Naumov, S., Yaroslavtsev, G., and Avdiukhin, D. Objective-based hierarchical clustering of deep embedding vectors. In *Proc. AAAI*, 2020.

Nguyen, T., Schmidt, B., and Kwoh, C. K. Sparsehc: A memory-efficient online hierarchical clustering algorithm. In Abramson, D., Lees, M., Krzhizhanovskaya, V. V., Dongarra, J. J., and Sloot, P. M. A. (eds.), *Proceedings of the International Conference on Computational Science, ICCS 2014*, volume 29 of *Procedia Computer Science*, pp. 8–19. Elsevier, 2014.

Rodrigues, P. P., Gama, J., and Pedroso, J. P. ODAC: hierarchical clustering of time series data streams. In Ghosh, J., Lambert, D., Skillicorn, D. B., and Srivastava, J. (eds.), *Proceedings of the Sixth SIAM International Conference on Data Mining*, pp. 499–503. SIAM, 2006.

Schmidt, M. and Sohler, C. Fully dynamic hierarchical diameter k-clustering and k-center. *CoRR*, abs/1908.02645, 2019. URL http://arxiv.org/abs/1908.02645.

Vainstein, D., Chatziafratis, V., Citovsky, G., Rajagopalan, A., Mahdian, M., and Azar, Y. Hierarchical clustering via sketches and hierarchical correlation clustering. *CoRR*, abs/2101.10639, 2021.

Wang, D. and Wang, Y. An improved cost function for hierarchical cluster trees. In *ArXiv, abs/1812.02715*, 2018.

Wang, Y. and Moseley, B. An objective for hierarchical clustering in euclidean space and its connection to bisecting k-means. In *Proc. AAAI*, pp. 6307–6314, 2020.

Zhang, T., Ramakrishnan, R., and Livny, M. Birch: An efficient data clustering method for very large databases. In *SIGMOD*, volume 25(2), pp. 103–114, 1996.

A. Supplementary material for Section 3

The pseudocode for Ins_μ is presented in Algorithm 1. For all algorithms, we use the notation $T \wedge T'$ to denote the tree with T (resp. T') as its left (resp. right) sub-tree, and **Node**(x) to denote a node containing point x.

Algorithm 1 Insert operation Ins_μ for the general dynamic algorithm of Section 3

```
Input ▹ Tree $T(X)$ on points $X = \{x_1, x_2, \ldots, x_n\} \subseteq \mathbb{R}^d$, and new point $x_{n+1}$
Output ▹ Tree $T(X \cup \{x_{n+1}\})$ on $X \cup \{x_{n+1}\}$

  • /* If the new point is the first one added to $T$, create the root of $T$ and return */
    If $X = \emptyset$ then return Node$(x_{n+1})$;

  • /* Draw a hyperplane $H$ from distribution $\mu_{X\cup\{x_{n+1}\}}$ */
    Draw $H := \{x \in \mathbb{R}^d : w \cdot x = b, \|w\| = 1\} \sim \mu_{X\cup\{x_{n+1}\}}$;

  • /* If $H$ separates $X$ and $x_{n+1}$, add $x_{n+1}$ and return */
    If $\forall i \leq n \ \mathrm{sgn}(w \cdot x_i - b) \neq \mathrm{sgn}(w \cdot x_{n+1} - b)$ then return $T(X) \wedge$ Node$(x_{n+1})$;
    /* Else let $H'$ be the hyperplane previously generated in $T$ to split $X$ into the two
    subsets $X^+$ and $X^-$ of $X$ */
    Else $H' \leftarrow$ hyperplane $\{x \in \mathbb{R}^d : w' \cdot x = b', \|w'\| = 1\}$ cutting $T(X)$;
    $X^- \leftarrow \{x \in X : w' \cdot x - b' \leq 0\}$;
    $X^+ \leftarrow \{X \in X : w' \cdot x - b' > 0\}$;

  • /* Recurs on the subset obtained by cutting $X$ with $H'$ where all points are on the same
    side w.r.t $x_{n+1}$ */
    If $w' \cdot x_{n+1} - b' \leq 0$ then return $\mathrm{Ins}_\mu(T(X^-), x_{n+1}) \wedge T(X^+)$;
    Else return $T(X^-) \wedge \mathrm{Ins}_\mu(T(X^+), x_{n+1})$;
```

We restate an observation made in Section 3 regarding the restriction of hyperplane measures.

Lemma A.1 (Restriction invariance property)**.** *Fix a hyperplane distribution μ and consider two finite sets $X, Y \subset \mathbb{R}^d$ with $X \subset Y$. Recall from Section 3 the notation μ_X to mean the probability measure on hyperplanes intersecting* $\mathrm{Conv}(X)$*. Then $(\mu_Y)|_X = \mu_X$, where $(\mu_Y)|_X$ is the restriction of μ_Y to hyperplanes intersecting* $\mathrm{Conv}(X)$*. In words, a hyperplane chosen from μ_Y conditioned on intersecting* $\mathrm{Conv}(X)$ *is equal in distribution to a hyperplane chosen from μ_X.*

Proof. This follows immediately from the definitions of μ_Y and μ_X, and the fact that the hyperplanes intersecting $\mathrm{Conv}(X)$ are a subset of those intersecting $\mathrm{Conv}(Y)$. □

Theorem 3.1. *Let μ be a nonnegative measure on* $\mathrm{Graff}_{d-1}(\mathbb{R}^d)$ *which is finite on compact sets, and suppose there is an efficient way to sample from μ_X for all finite sets X. Then, there is an efficient insertion operation* Ins_μ *such that A_μ has the sequential property w.r.t.* Ins_μ*.*

Proof. We wish to show that, for all finite $X \subset \mathbb{R}^d$, and $x \in \mathbb{R}^d$, we have $\mathrm{Ins}_\mu(A_\mu(X), x) \overset{d}{=} A_\mu(X \cup \{x\})$, where $\overset{d}{=}$ denotes equality in distribution, and A_μ is the recursive algorithm defined in Section 3. We prove the claim by induction on the size of X. For $|X| = 1$, both $\mathrm{Ins}_\mu(\emptyset, x)$ and $A_\mu(x)$ return a singleton node with point x. By the induction hypothesis, we will assume that the claim is true for all sets X with $|X| \leq k - 1$, and consider $|X| = k$. Let $T'(X, x) = \mathrm{Ins}_\mu(A_\mu(X), x)$ and $T(X \cup \{x\}) = A_\mu(X \cup \{x\})$. We first show that the root cuts H of T and H' of T' are equal in distribution. By Algorithm 1, H' is chosen from $\mu_{X\cup\{x\}}$ if H' separates X from x. Otherwise, H' is left as the root cut of $A_\mu(X)$ which is distributed as μ_X. However, by Lemma A.1, $\mu_{X\cup\{x\}}|_X = \mu_X$. Thus $H' \overset{d}{=} H$. In the event that H' and H split X from x, the subtree of T' containing X is distributed as $T(X)$ and hence $T' \overset{d}{=} T$. Now consider the event that H' does not split X from x, and that (without loss of generality) x is inserted into the subtree X' of X, with X'' unchanged. In this case, by Algorithm 1, $T'(X' \cup \{x\}) = \mathrm{Ins}_\mu(A_\mu(X'), x)$ and by the induction hypothesis, the latter is distributed as $A_\mu(X' \cup \{x\})$. Since X'' is untouched, we also have $T'(X'') \overset{d}{=} T(X'')$. Putting the above cases together completes the proof. □

B. Supplementary material for Section 4

B.1. Proofs of Fact 4.1 and Lemmas 4.3, 4.7, and 4.8

Fact 4.1. *Fix dimension d, and let $H_{i,v} = \{x \in \mathbb{R}^d \,|\, x_i = v\}$, where x_i is the i-th component of vector x. Let then $\mathcal{H} = \{H_{i,v} \,|\, i \in [d], v \in \mathbb{R}\}$ be the set of axis-parallel hyperplanes. For $\mathcal{H}' \subset \mathcal{H}$, define μ_{RCT} by $\mu_{\text{RCT}}(\mathcal{H}') = \sum_{i=1}^{d} \mu_L(\{v \in \mathbb{R} \,|\, H_{i,v} \in \mathcal{H}'\})$, where L is the standard Lebesgue measure on $\mathbb{R}$. Then $A_{\mu_{\text{RCT}}}$ (resp. $\text{Ins}_{\mu_{\text{RCT}}}$) is the offline (resp. dynamic) RCT algorithm.*

Proof. We recall the definition of an RCT (batch algorithm) from (Guha et al., 2016):

Definition B.1 (Batch RCT Algorithm)**.** *A random cut tree (RCT) $T(X)$ on item set $X \subseteq \mathbb{R}^d$ is a tree-valued random variable generated as follows:*

1. *Draw random index $I \in [d]$ with probability $\mathbb{P}[I = i] = \frac{l_i}{\sum_{i=1}^{d} l_i}$, where*

$$l_i = \max_{x \in X}(x)_i - \min_{x \in X}(x)_i \ ,$$

with $(x)_i$ denoting the i-th component of vector x. Hence the above probability is proportional to the side lengths of the (axis-parallel minimum) bounding box of X;

2. *Draw threshold $\theta \sim$ Uniform$[\min_{x \in X} x_I, \max_{x \in X} x_I]$;*
3. *Let $X_1 = \{x \,|\, x \in X, (x)_I \leq \theta\}$ and $X_2 = X \backslash X_1$ correspond to the left and right subtrees of the root of $T(X)$, and recurse on X_1 and X_2, until $T(X)$ is a (singleton) leaf.*

Comparing this definition with the definition of A_μ given in Section 3, we see that we need to show that sampling a hyperplane H from μ_{RCT} is equivalent to the sampling done in steps 1 and 2 above. Note that μ_{RCT} is supported on the axis-parallel hyperplanes. Given a finite set $X \subset \mathbb{R}^d$, let its bounding box be $\prod_{i=1}^{d} [(x^l)_i, (x^h)_i]$, and set $l_i = (x^h)_i - (x^l)_i$. The set of axis-parallel hyperplanes that intersect $\text{Conv}(X)$ is $\mathcal{H}'(X) = \bigcup_{i=1}^{d} \{H_{i,v} \,|\, v \in [(x^l)_i, (x^h)_i]\}$. By definition, μ_{RCT} assigns a measure of $\sum_{i=1}^{d} l_i$ to $\mathcal{H}'(X)$. Hence the $\mu_{\text{RCT},X}$-probability of selecting a hyperplane parallel to dimension j is precisely $l_j / \sum_{i=1}^{d} l_i$ as done in step 1. Furthermore, since μ_{RCT} assigns Lebesgue measure to the hyperplanes in a given axis, sampling within a dimesion j is uniform, as is done in step 2. This concludes the proof. □

We state and prove the generalization of Lemma 4.3 for all algorithms A_μ.

Lemma B.2. *Let A_μ be any algorithm in the family of hyperplane-based HC algorithms, $X \subset \mathbb{R}^d$ be any finite set of items, and denote by $T(X)$ the output of A_μ on input X. Then for any $R \subseteq X$, the restriction of $T(X)$ to subset R has the same distribution as $T(R)$.*

Notice that since by Fact 4.1 RCT is one such A_μ, this directly implies Lemma 4.3 in the main body of the paper.

Proof of Lemma B.2. Consider A_μ applied to X. Let $H_P \sim \mu_P$ be the first cut that separates R, where P is the set of points in the region cut out by hyperplanes and containing R at the time R is separated. By Lemma A.1, $\mu_P|_R = \mu_R$. Thus H_P has the same distribution of the first cut of $T(R)$. Let P^+ and P^- denote the partition of P induced by H_P, and let R^+ and R^- denote the corresponding partition of R. Continuing recursively on the pairs (P^+, R^+) and (P^-, R^-) completes the proof. □

Lemma 4.7. *Given input X and a cut H_X sampled from μ_X, the probability $p_{i,j}$ that x_i and x_j are split by H is proportional to their ℓ_1 distance $d_{i,j}$.*

Proof. Let $B(X) = \prod_{k=1}^{d} [(x^l)_k, (x^h)_k]$ be the bounding box of X. For $k \in [d]$, we let $l_k = (x^h)_k - (x^l)_k$ denote the dimensional lengths of $B(X)$. As in the proof of Fact 4.1, the measure of axis-parallel hyperplanes that intersect $B(X)$ is given by $\sum_{k=1}^{d} l_k$, the ℓ_1 diameter of $B(X)$. Similarly, the measure of hyperplanes that separate x_i and x_j is given by the ℓ_1 diameter of their bounding box. However, the latter is simply their ℓ_1 distance $d_{i,j}$. Thus the probability that x_i and x_j are separated is given by $p_{i,j} = \frac{d_{i,j}}{\sum_{k=1}^{d} l_k}$. □

Algorithm 2 Insert operation $\mathrm{Ins}_{\mu_{\text{RCT}}}$ for RCT from (Guha et al., 2016).

```
Input ▷ Tree $T(X)$ on points $X = \{x_1, x_2, \ldots, x_n\} \subseteq \mathbb{R}^d$, and new point $x_{n+1}$
Output ▷ Tree $T(X \cup \{x_{n+1}\})$ on items $X \cup \{x_{n+1}\}$

• /* If the new point is the first one added to $T$, create the root of $T$ and return */
  If $X = \emptyset$ then return Node($x_{n+1}$);

• /* Draw an axis-parallel hyperplane $H$ that intersects the bounding box of $X$ */
  Let $B(X) = \prod_{i=1}^{d} [(x^l)_i, (x^h)_i]$ be the bounding box of $X$;
  For $i \in [d]$, let $(\hat{x}^l)_i = \min\{(x_{n+1})_i, (x^l)_i\}$, $(\hat{x}^h)_i = \max\{(x_{n+1})_i, (x^h)_i\}$, and $l_i = (\hat{x}^h)_i - (\hat{x}^l)_i$;
  Choose a dimension $j \in [d]$ with probability $p_j = l_j / \sum_{i \in [d]} l_i$ and choose a random number $v$ in $[(x^l)_i, (x^h)_i)$;
  Define $H$ as the hyper-plane $\{x \in \mathbb{R}^d | x_j = v\}$ where $x_j$ denotes the $j$th coordinate of $x$;

• /* If $H$ separates $X$ and $x_{n+1}$, add $x_{n+1}$ and return */
  If $\forall i \le n \ \mathrm{sgn}((x_i)_j - v) \neq \mathrm{sgn}((x_{n+1})_j - v)$ then return $T(X) \wedge$ Node($x_{n+1}$);
  /* Else let $H'$ be the hyperplane previously generated in $T$ to split $X$ into the two
  subsets $X^+$ and $X^-$ of $X$ */
  Else $H' \leftarrow$ hyperplane $\{x \in \mathbb{R}^d : x_{j'} = v'\}$ cutting $T(X)$;
  $X^- \leftarrow \{x \in X : x_{j'} \le v'\}$;
  $X^+ \leftarrow \{x \in X : x_{j'} > v'\}$;

• /* Recurs on the subset obtained by cutting $X$ with $H'$ where all points are on the same
  side w.r.t $x_{n+1}$ */
  If $(x_{n+1})_{j'} \le v'$ then return $\mathrm{Ins}_{\mu_{\text{RCT}}}(T(X^-), x_{n+1}) \wedge T(X^+)$;
  Else return $T(X^-) \wedge \mathrm{Ins}_{\mu_{\text{RCT}}}(T(X^+), x_{n+1})$;
```

The following lemma relates probabilities of separating a pair of points to probabilities of separating a triplet of points.

Lemma B.3. *Fix a hyperplane measure μ satisfying that for all $x \in \mathbb{R}^d$, $\mu(\{H \ni x\}) = 0$. Let $t = \{x_i, x_j, x_k\}$ and let H be a hyperplane chosen from μ_t. Let E_a (resp., E_b, E_c) be the event that H separates x_i and x_j (resp. x_j and x_k, x_i and x_k). Then the probability that H separates x_i and x_j from x_k is given by*

$$p_{i,j|k} = \frac{\mathbb{P}(E_b) + \mathbb{P}(E_c) - \mathbb{P}(E_a)}{\mathbb{P}(E_a) + \mathbb{P}(E_b) + \mathbb{P}(E_c)},$$

and similarly for $p_{i,k|j}$ and $p_{j,k|i}$.

Proof. Since H will cut two sides of the triangle defined by t, the events E_a, E_b, and E_c, are not disjoint, On the other hand, the events $E_a \cap E_b$, $E_a \cap E_c$, and $E_b \cap E_c$ are disjoint[8] and collectively exhaustive. In fact, we have

$$\begin{aligned} \mathbb{P}(E_a) &= \mathbb{P}(E_a \cap E_b) + \mathbb{P}(E_a \cap E_c), \\ \mathbb{P}(E_b) &= \mathbb{P}(E_a \cap E_b) + \mathbb{P}(E_b \cap E_c), \\ \text{and} \quad \mathbb{P}(E_c) &= \mathbb{P}(E_a \cap E_c) + \mathbb{P}(E_b \cap E_c), \end{aligned}$$

which we can invert to get

$$p_{i,j|k} = \mathbb{P}(E_b \cap E_c) = \frac{\mathbb{P}(E_b) + \mathbb{P}(E_c) - \mathbb{P}(E_a)}{2}.$$

Since $\mathbb{P}(E_a) + \mathbb{P}(E_b) + \mathbb{P}(E_c) = 2(\mathbb{P}(E_a \cap E_b) + \mathbb{P}(E_a \cap E_c) + \mathbb{P}(E_b \cap E_c)) = 2$, we may rewrite the above as

$$p_{i,j|k} = \frac{\mathbb{P}(E_b) + \mathbb{P}(E_c) - \mathbb{P}(E_a)}{\mathbb{P}(E_a) + \mathbb{P}(E_b) + \mathbb{P}(E_c)}.$$

as claimed. By symmetry, we have the analogous expressions for $p_{i,k|j}$ and $p_{j,k|i}$. □

[8] We ignore the measure zero event that H contains one of the vertices

Lemma 4.8. *Fix a triplet $\{x_i, x_j, x_k\}$ of X. Then the probability, $p_{i,j|k}$, that* RCT *$T(X)$ separates x_i and x_j from x_k is given by*

$$p_{i,j|k} = \frac{d_{i,k} + d_{j,k} - d_{i,j}}{d_{i,j} + d_{i,k} + d_{j,k}} \ ,$$

and similarly for $p_{i,k|j}$ and $p_{j,k|i}$.

Proof. Let E_a be the event that the first cut of RCT that splits the triplet separates x_i from x_j (and similarly define E_b and E_c). From Lemma B.3, we have $p_{i,j|k} = (\mathbb{P}(E_b) + \mathbb{P}(E_c) - \mathbb{P}(E_a))/(\mathbb{P}(E_a) + \mathbb{P}(E_b) + \mathbb{P}(E_c))$. On the other hand, by Lemma 4.7, we have $\mathbb{P}(E_a) = \lambda d_{i,j}$, $\mathbb{P}(E_b) = \lambda d_{j,k}$, and $\mathbb{P}(E_c) = \lambda d_{i,k}$, where the $d_{i,j}$'s are ℓ_1 distances, and λ is independent of i, j, k. Substituting these expressions above, we arrive at

$$p_{i,j|k} = \frac{d_{i,k} + d_{j,k} - d_{i,j}}{d_{i,j} + d_{i,k} + d_{j,k}} \ ,$$

and similarly for $p_{i,k|j}$ and $p_{j,k|i}$. □

B.2. Proof of Theorem 4.6

A common ingredient for all four objectives is their triplet-wise decomposition, i.e., $\mathrm{Obj}(T) = \sum_{i<j<k} f_{i,j,k}(T)$, where $f(i,j,k)$ is given by the following formulae:

1. MW Revenue:

$$f_{i,j,k} = \begin{cases} w_{i,j} & \text{if } ij|k \\ w_{i,k} & \text{if } ik|j \\ w_{j,k} & \text{if } jk|i \ . \end{cases}$$

2. MW Cost:

$$f_{i,j,k} = \begin{cases} d_{i,j} & \text{if } ij|k \\ d_{i,k} & \text{if } ik|j \\ d_{j,k} & \text{if } jk|i \ . \end{cases}$$

3. Dasgupta Cost:

$$f_{i,j,k} = (2/(n-2))(w_{i,j} + w_{i,k} + w_{j,k}) + \begin{cases} w_{i,k} + w_{j,k} & \text{if } ij|k \\ w_{i,j} + w_{j,k} & \text{if } ik|j \\ w_{i,j} + w_{i,k} & \text{if } jk|i \ . \end{cases}$$

4. CKMM Revenue:

$$f_{i,j,k} = (2/(n-2))(d_{i,j} + d_{i,k} + d_{j,k}) + \begin{cases} d_{i,k} + d_{j,k} & \text{if } ij|k \\ d_{i,j} + d_{j,k} & \text{if } ik|j \\ d_{i,j} + d_{i,k} & \text{if } jk|i \ . \end{cases}$$

In the latter two cases, define the *modified triplet term*

$$g_{i,j,k} = \begin{cases} \rho_{i,k} + \rho_{j,k} & \text{if } ij|k \\ \rho_{i,j} + \rho_{j,k} & \text{if } ik|j \\ \rho_{i,j} + \rho_{i,k} & \text{if } jk|i, \end{cases}$$

with $\rho = w$ for the Dasgupta cost and $\rho = d$ for the CKMM Revenue.

Lemma B.4. *It suffices to use $g_{i,j,k}$ in the proofs of the Dasgupta Cost and CKMM Revenue guarantees.*

Proof. Note that $\sum_{i<j<k} f_{i,j,k} = \sum_{i<j<k} g_{i,j,k} + 2\sum_{i<j} w_{i,j}$ (since, e.g., the term $(2/(n-2))w_{i,j}$ appears $n-2$ times in the sum over triplets). Now suppose we have the guarantee $\sum_{i<j<k} g_{i,j,k}(T_{\text{RCT}}) \geq \alpha \sum_{i<j<k} g_{i,j,k}(T_{\text{Opt}})$, where T_{Opt} is the optimal tree under CKMM Revenue and $\alpha \leq 1$. Then

$$\begin{aligned}\sum_{i<j<k} f_{i,j,k}(T_{\text{RCT}}) &= \sum_{i<j<k} g_{i,j,k} + 2\sum_{i<j} w_{i,j}\\ &\geq \alpha\left(\sum_{i<j<k} g_{i,j,k}(T_{\text{Opt}})\right) + \alpha\left(2\sum_{i<j} w_{i,j}\right)\\ &= \alpha(\sum_{i<j<k} f_{i,j,k}(T_{\text{Opt}})).\end{aligned}$$

A similar calculation in the case of Dasgupta cost (with $\alpha \geq 1$ and the inequalities reversed) yields the claim. □

In the following, for triplet $\{x_i, x_j, x_k\}$, we will use $\text{Rev}_{i,j,k}(T)$ to denote the MW Revenue triplet term and CKMM Revenue modified triplet term for a tree T, and denote by $C_{i,j,k}(T)$ the modified Dasgupta Cost triplet term and MW Cost triplet term. When T is the optimal tree under one of these measures, we will use the notation $\text{Rev}_{i,j,k}(\text{OPT})$, resp. $C_{i,j,k}(\text{OPT})$. We also use the notation $B_{i,j,k} = (d_{i,j} + d_{i,k} + d_{j,k})/2$ which is also the ℓ_1 diameter of the bounding box of triplet $\{x_i, x_j, x_k\}$.

B.2.1. Case 1: MW Revenue + L1-similarity

In this case $w_{i,j} = D - d_{i,j},\ \forall i \neq j$, where $D = \max_{i,j} d_{i,j}$ and $d_{i,j}$ is the ℓ_1-distance. We prove the following result.

Theorem B.5. *For $X \subseteq \mathbb{R}^d$ endowed with ℓ_1 similarity and satisfying Assumption 4.4, we have $\mathbb{E}\text{Rev}_S(\text{RCT}(X)) \geq (\sqrt{3}-1)\text{Opt}_{\text{Rev}_S}$, where the expectation is over the internal randomization of* RCT.

We will need the following inequality.

Lemma B.6. *Let x, y, z be nonnegative real numbers. Then*

$$x^2 + y^2 + z^2 \geq (\sqrt{3}-1)(x+y+z)\max(x,y,z).$$

Proof. Without loss of generality, assume $z = \max(x,y,z) = 1$. We then need to minimize $f(x,y) = \frac{1+x^2+y^2}{1+x+y}$ for $0 \leq x, y \leq 1$. Letting $u = x+y$, we can rewrite f as $\frac{1+x^2+(u-x)^2}{1+u}$. For fixed u, this is minimized at $x = u/2$. Thus it suffices to minimize f when $x = y$. Letting $g(x) = f(x,x) = \frac{1+2x^2}{1+2x}$, and setting $g'(x) = 0$ shows that g is minimized at $x = \frac{\sqrt{3}-1}{2}$ where it achieves a value of $\sqrt{3}-1$. □

Proof of Theorem B.5. We first prove a triplet-wise bound. Fix a triplet $\{x_i, x_j, x_k\}$, define $B_{i,j,k} = (d_{i,j} + d_{j,k} + d_{i,k})/2$, and define $\Delta_{i,j,k} = D - B_{i,j,k}$. Then $w_{i,j} = B_{i,j,k} - d_{i,j} + \Delta_{i,j,k}$, etc., and

$$\text{Rev}_{i,j,k}(\text{OPT}) = \max(B_{i,j,k} - d_{i,j}, B_{i,j,k} - d_{i,k}, B_{i,j,k} - d_{j,k}) + \Delta_{i,j,k}.$$

To calculate $\mathbb{E}\text{Rev}_{i,j,k}(\text{RCT})$, using Lemma 4.8 we have

$$\begin{aligned}\mathbb{E}\text{Rev}_{i,j,k}(\text{RCT}) &= \sum_{\text{cyc}} p_{i,j|k} w_{i,j}\\ &= \sum_{\text{cyc}} \frac{B_{i,j,k} - d_{i,j}}{B_{i,j,k}}(B_{i,j,k} - d_{i,j} + \Delta_{i,j,k})\\ &= \frac{1}{B_{i,j,k}}\sum_{\text{cyc}}(B_{i,j,k} - d_{i,j})^2 + \Delta_{i,j,k},\end{aligned}$$

where we recall the cyclic sum notation $\sum_{\text{cyc}} f(i,j,k) = f(i,j,k) + f(j,k,i) + f(k,i,j)$.

Let $x = B_{i,j,k} - d_{j,k}$, $y = B_{i,j,k} - d_{i,k}$, and $z = B_{i,j,k} - d_{i,j}$ and note that $B_{i,j,k} = x + y + z$ (since $\sum_{\text{cyc}} p_{i,j|k} = 1$). Then by Lemma B.6, we have

$$\begin{aligned}
\mathbb{E}\mathrm{Rev}_{i,j,k}(\textsc{rct}) &= \frac{x^2+y^2+z^2}{x+y+z} + \Delta_{i,j,k} \\
&\geq (\sqrt{3}-1)(\max(x,y,z) + \Delta_{i,j,k}) + (2-\sqrt{3})\Delta_{i,j,k} \\
&= (\sqrt{3}-1)\mathrm{Rev}_{i,j,k}(\textsc{opt}) + (2-\sqrt{3})\Delta_{i,j,k} \ .
\end{aligned}$$

Summing over triplets and using our assumption that $\binom{n}{3}^{-1}\sum_{i<j<k}\Delta_{i,j,k} \geq 0$ yields the desired bound. □

B.2.2. Case 2: Dasgupta Cost + L1-similarity

Theorem B.7. *For $X \subseteq \mathbb{R}^d$ with endowed with ℓ_1 similarity and satisfying Assumption 4.4, we have $\mathbb{E}\mathrm{Cost}_S(\textsc{rct}(X)) \leq 2\mathrm{Opt}_{\mathrm{Cost}_S}$, where the expectation is over the internal randomization of* RCT.

In this case $w_{i,j} = D - d_{i,j}$, $\forall i \neq j$, where $D = \max_{i,j} d_{i,j}$ and $d_{i,j}$ is the ℓ_1-distance. Consider the modified triplet term from Lemma B.4,

$$C_{i,j,k}(T) = \begin{cases} w_{i,k} + w_{j,k} & \text{if } ij|k \\ w_{i,j} + w_{j,k} & \text{if } ik|j \\ w_{i,j} + w_{i,k} & \text{if } jk|i \ . \end{cases}$$

We will show that under Assumption 4.4, RCT achieves a 2-approximation of Cost_S. The proof is similar to that of Theorem B.5. Fix a triplet $\{x_i, x_j, x_k\}$, let $a = d_{i,j}$, $b = d_{i,k}$, $c = d_{j,k}$, and assume without loss of generality that $c \leq b \leq a$. Then $C_{i,j,k}(\textsc{opt}) \geq 2D - (a+b) = c + 2(D - B_{i,j,k})$ (recall that $B_{i,j,k} = (a+b+c)/2$). On the other hand,

$$\begin{aligned}
C_{i,j,k}(\textsc{rct}) &= \sum_{\text{cyc}} \frac{b+c-a}{a+b+c}(2D - (b+c)) \\
&= 2(D - B_{i,j,k}) + \sum_{\text{cyc}} \frac{b+c-a}{a+b+c}(2B_{i,j,k} - (b+c)) \\
&= 2(D - B_{i,j,k}) + \sum_{\text{cyc}} \frac{a(b+c-a)}{a+b+c} \\
&= 2(D - B_{i,j,k}) + \frac{2(ab+ac+bc) - (a^2+b^2+c^2)}{a+b+c} \\
&= 2(D - B_{i,j,k}) + \frac{2c(a+b) - c^2 - (a-b)^2}{a+b+c} \\
&\leq 2(c + D - B_{i,j,k}) \\
&= 2C_{i,j,k}(\textsc{opt}) - 2(D - B_{i,j,k}) \ .
\end{aligned}$$

The result follows by summing over triplets, and using $\binom{n}{3}^{-1}\sum_{i<j<k} B_{i,j,k} \leq D$.

Remark B.8. *The example $(a,b,c) = (n,n,1)$ with n large shows that* RANDOM *can perform arbitrarily poorly for this objective. Indeed, with $D = (a+b+c)/2 = n + 1/2$, the weights are $(1/2, 1/2, n-1/2)$. $\mathrm{Cost}_S(\textsc{opt}) = 1$ while $\mathrm{Cost}_S(\textsc{random}) = \frac{2}{3}(1/2 + 1/2 + (n-1/2)) = (2n+1)/3$.*

B.2.3. Case 3: CKMM Revenue + L1-distance

Theorem B.9. *For $X \subseteq \mathbb{R}^d$ endowed with the ℓ_1 metric, we have $\mathbb{E}\mathrm{Rev}_D(\textsc{rct}(X)) \geq (2\sqrt{6}-4)\mathrm{Opt}_{\mathrm{Rev}_D}$, where the expectation is over the internal randomization of* RCT.

For this and the following dissimilarity objective, we use the distances themselves for dissimilarities, and we will not require any assumptions. Reusing notation from the previous subsection, we assume $1 = c \leq b \leq a$. Using the modified triplet term from Lemma B.4, we have $\mathrm{Rev}_{i,j,k}(\textsc{opt}) = a + b$, while

$$\begin{aligned}\mathrm{Rev}_{i,j,k}(\textsc{rct}) &= \sum_{\text{cyc}} \frac{b+c-a}{a+b+c}(b+c) \\ &= \frac{2(a^2+b^2+c^2)}{a+b+c} \\ &= \frac{2(a^2+b^2+1)}{a+b+1} \,.\end{aligned}$$

Following the approach in the proof of Lemma B.6, we set $f(a,b) = \frac{2(a^2+b^2+1)}{(a+b+1)(a+b)}$ and seek to minimize f for $a, b \geq 1$. For fixed $u = a + b$ we note that f is minimized at $a = u/2 = b$. Now setting $g(a) = f(a,a) = \frac{2a^2+1}{2a^2+a}$, we find after a routine calculation that g is minimized at $a = 1 + \sqrt{6}/2$ where it achieves a value of $2\sqrt{6} - 4 \approx 0.90$.

Remark B.10. *In this case,* RANDOM *achieves an approximation of* $2/3$*, which should be thought of as the baseline.*

B.2.4. CASE 4: MW COST + L1-DISTANCE

Theorem B.11. *For* $X \subseteq \mathbb{R}^d$ *endowed with the* ℓ_1 *metric, we have* $\mathbb{E}\mathrm{Cost}_D(\textsc{rct}(X)) \leq 2\mathrm{Opt}_{\mathrm{Cost}_D}$*, where the expectation is over the internal randomization of* RCT*.*

Following the same notation as before and assuming $c = \min(a,b,c) = 1$, in this case we have $C_{i,j,k}(\textsc{opt}) \geq c$, while $C_{i,j,k}(\textsc{rct}) = \sum_{\text{cyc}} \frac{b+c-a}{a+b+c} \cdot a$. This expression is identical to that in the calculation for the Dasgupta cost and we recover the bound $C_{i,j,k}(\textsc{rct}) \leq 2c$. Summing over the triplets yields a 2-approximation result here as well. The example $(a,b,c) = (n,n,1)$ with n large shows that RANDOM performs arbitrarily poorly for this objective as well.

B.2.5. CASES 5 AND 6: INVERSE L1 DISTANCE

We extend the approximation results for the similarity-based objectives (MW Revenue and Dasgupta Cost) to the case where weights are defined by $w_{i,j} = 1/d_{i,j}$. We reuse the notation $a = d_{i,j}$, etc. and assume $1 = c \leq b \leq a$.

Theorem B.12. *For* $X \subseteq \mathbb{R}^d$ *with* $w_{i,j} = 1/||x_i - x_j||_1$*, we have* $\mathbb{E}\mathrm{Cost}_S(\textsc{rct}(X)) \leq (3/2)\mathrm{Opt}_{\mathrm{Cost}_S}$*, where the expectation is over the internal randomization of* RCT*.*

$$\begin{aligned}C_{i,j,k}(\textsc{rct}) &= \sum_{\text{cyc}} \frac{b+c-a}{a+b+c}\left(\frac{1}{b}+\frac{1}{c}\right) \\ &= \frac{6}{a+b+c} \\ &\leq \frac{3}{2}\left(\frac{1}{a}+\frac{1}{b}\right) \\ &\leq \frac{3}{2}C_{i,j,k}(\textsc{opt}) \,.\end{aligned}$$

Theorem B.13. *For* $X \subseteq \mathbb{R}^d$ *with* $w_{i,j} = 1/||x_i - x_j||_1$*, we have* $\mathbb{E}\mathrm{Rev}_S(\textsc{rct}(X)) \geq (4\sqrt{6} - 9)\mathrm{Opt}_{\mathrm{Rev}_S}$*, where the expectation is over the internal randomization of* RCT*.*

We have $\mathrm{Rev}_{i,j,k}(\textsc{opt}) \leq 1/c = 1$ and $\mathrm{Rev}_{i,j,k}(\textsc{rct}) = \sum_{\text{cyc}} \frac{b+c-a}{a+b+c} \cdot \frac{1}{a}$. A more involved computation, but similar to the approach of Lemma B.6 shows that $\mathrm{Rev}_{i,j,k}(\textsc{rct})$ is minimized when $a = b$. It is then straightforward to show that this occurs at $a = 1 + \sqrt{6}/2$ where $\mathrm{Rev}_{i,j,k}(\textsc{rct})$ achieves a value of $4\sqrt{6} - 9 \approx 0.80$.

B.2.6. CASE 7: MW REVENUE + ABSOLUTE EXPONENTIAL KERNEL

We extend the MW Revenue approximation result to the case when the weights are given by an absolute exponential kernel, namely, $w_{i,j} = e^{-d_{i,j}/\lambda}$, with length scale $\lambda > 0$. In order to obtain better than random results, we assume that there exists a δ such that $w_{i,j} \geq \delta$ for all pairs i and j. We note that the same assumption is made in (Charikar et al., 2019b).

Theorem B.14. *For $X \subseteq \mathbb{R}^d$ with $w_{i,j} = e^{-d_{i,j}/\lambda}$ and $w_{i,j} \geq \delta$ for all pairs i and j, we have $\mathbb{E}\mathrm{Rev}_S(\textsc{rct}(X)) \geq f(\delta)\mathrm{Opt}_{\mathrm{Rev}_S}$, where the expectation is over the internal randomization of* RCT *and $f(\delta)$ is the green curve labeled "Abs. Exp. Kernel" in Figure 2.*

Fixing δ, we let $d^* = \lambda \ln(1/\delta)$ be the distance that achieves $w_{i,j} = \delta$. Using the notation $c \leq b \leq a$ for distances, we have $\mathrm{Rev}_{i,j,k}(\textsc{opt}) \leq e^{-c/\lambda}$ while

$$\mathrm{Rev}_{i,j,k}(\textsc{rct}) = ((a+b-c)e^{-c/\lambda} + (a+c-b)e^{-b/\lambda} + (b+c-a)e^{-a/\lambda})/(a+b+c)\ .$$

We wish to minimize the ratio $f(a,b,c) := \mathrm{Rev}_{i,j,k}(\textsc{rct})/\mathrm{Rev}_{i,j,k}(\textsc{opt})$ over $0 \leq c \leq b \leq a \leq b+c$. We show analytically that the minimum is achieved for $a = b = d^*$. The final minimization over c, however, is not analytically tractable and is performed numerically.

Fixing both c and $u = a+b$, we minimize

$$g(a; u, c) = (u - 2a + c)e^{(c-a)/\lambda} + (2a - u + c)e^{(c-u+a)/\lambda} + (u - c)\ .$$

Since $g(a) = g(u-a)$, $\frac{d}{da}g(a) = 0$ for $a = u/2$. We have

$$\begin{aligned}\frac{d}{da}g(a) &= ((u-2a+c)(-1/\lambda) - 2)e^{(c-a)/\lambda} + ((2a-u+c)(1/\lambda) + 2)e^{(c-u+a)/\lambda}\\ &= \left(\frac{2a-u}{\lambda}\right)(e^{(c-u+a)/\lambda} + e^{(c-a)/\lambda}) + \left(2 + \frac{c}{\lambda}\right)(e^{(c-u+a)/\lambda} - e^{(c-a)/\lambda})\ .\end{aligned}$$

Since the last expression is positive for $a > u/2$ and negative for $a < u/2$, $a = u/2$ is the global minimum of $g(a)$. Thus $f(a,b,c)$ is minimized for $a = b$. Incorporating the observation that jointly scaling up the variables decreases the ratio gives us the claim that it is minimized at $a = b = d^*$.

We now turn to the task of minimizing over c. For fixed δ, the minimum is independent over λ as can be seen from the scaling $a' = a/\lambda$, etc. We thus set $\lambda = 1$ and minimize $(2ce^{c-a} + (2a-c))/(c+2a)$, where $a = \ln(1/\delta)$ and $c \leq a$. The results are shown in Figure 2 as a function of δ, along with the comparison with the approximation ratio of $(1+2\delta)/3$ achieved by RANDOM. Note that RCT significantly improves upon RANDOM for δ bounded away from 0 and 1.

B.2.7. CASE 8: MW REVENUE + GAUSSIAN KERNEL

We extend the MW Revenue approximation result to the case when the weights are given by a Gaussian kernel, namely $w_{i,j} = e^{-d_{i,j}^2/2\sigma^2}$ for length scale $\sigma > 0$. As in the previous result, we assume that $w_{i,j} \geq \delta$ for all pairs i and j, following (Charikar et al., 2019a).

Theorem B.15. *For $X \subseteq \mathbb{R}^d$ with $w_{i,j} = e^{-d_{i,j}^2/2\sigma^2}$ and $w_{i,j} \geq \delta$ for all pairs i and j, we have $\mathbb{E}\mathrm{Rev}_S(\textsc{rct}(X)) \geq f(\delta)\mathrm{Opt}_{\mathrm{Rev}_S}$, where the expectation is over the internal randomization of* RCT *and $f(\delta)$ is the yellow curve labeled "Gaussian Kernel" in Figure 2.*

We seek to minimize the approximation ratio given by the function

$$f(a,b,c) = ((b+c-a)e^{(c^2-a^2)/2\sigma^2} + (a+c-b)e^{(c^2-b^2)/2\sigma^2} + (a+b-c))/(a+b+c)$$

over $0 \leq c \leq b \leq a \leq b+c$. Letting d^* be such that $e^{-d^{*2}/2\sigma^2} = \delta$, we first show that f is minimized when $a = b = d^*$. As before, we note that scaling up the variables only decreases f. Fixing c and $u = a+b$, we minimize

$$g(a; u, c) = (u - 2a + c)e^{(c^2-a^2)/2\sigma^2} + (2a - u + c)e^{(c^2-(u-a)^2)/2\sigma^2},$$

where we have omitted the last term and denominator of f which are constant. Since $g(a) = g(u-a)$, $\frac{d}{da}g(a) = 0$ for $a = u/2$. For $a > u/2$, we have

$$\begin{aligned}\frac{d}{da}g(a) &= ((u-2a+c)(-a/\sigma^2) - 2)e^{(c^2-a^2)/2\sigma^2} + ((2a-u+c)(u-a)/\sigma^2 + 2)e^{(c^2-(u-a)^2)/2\sigma^2}\\ &> e^{(c^2-a^2)/2\sigma^2}(1/\sigma^2)(c(u-2a) + (u-2a)(-a) + (2a-u)(u-a))\\ &= e^{(c^2-a^2)/2\sigma^2}(1/\sigma^2)(2a-u)(u-c) > 0.\end{aligned}$$

For fixed δ, the minimum is independent over σ as can be seen from the scaling $a' = a/\sigma$, etc. We thus set $\sigma = 1$ and minimize $(2ce^{(c^2-a^2)/2} + (2a-c))/(c+2a)$, where $a = \sqrt{2\ln(1/\delta)}$. The results are shown in Figure 2, along with the comparison with the approximation ratio of $(1+2\delta)/3$ achieved by RANDOM. Note again that RCT significantly improves upon RANDOM for δ bounded away from 0 and 1.

B.2.8. GENERALIZED THEOREM 4.6

We generalize Theorem 4.6 for a natural class of hyperplane measures μ that we define below and that includes μ_{RCT} and μ_{URRH}. We first introduce some notation. For $x, y \in \mathbb{R}^d$, we say that a hyperplane $H = \{z \in \mathbb{R}^d \mid z \cdot u = b\}$ *separates* x and y if $\operatorname{sgn}(x \cdot u - b) \neq \operatorname{sgn}(y \cdot u - b)$. We also write $\mathcal{H}_{xy} = \{H \in \mathcal{H} \mid H \text{ separates } x \text{ and } y\}$ for the set of hyperplanes separating x and y.

Definition B.16. *We say a hyperplane meaure μ on the manifold $\mathcal{H}$ of hyperplanes in $\mathbb{R}^d$ is* admissible *if it satisfies the following:*

(i) Non-negativity: For all (measurable) $\mathcal{H}' \subset \mathcal{H}$, $\mu(\mathcal{H}') \geq 0$

(ii) Finite on compact sets: For all finite sets X, $\mu(\mathcal{H}_X) < \infty$, where $\mathcal{H}_X$ is the set of hyperplanes intersecting $\operatorname{Conv}(X)$.

(iii) Supported: $\mu(\mathcal{H}_{xy}) > 0$ for all $x, y \in \mathbb{R}^d$, $x \neq y$.

(iv) Non-atomic: For all $x \in \mathbb{R}^d$, $\mu(\{H \in \mathcal{H} \mid x \in H\}) = 0$.

(i) and (ii) are repeated from Section 3. (iii) is needed below to establish the correspondence between such measures and distance metrics. (iv) allows us to disregard edge cases of hyperplanes passing through a specific point as measure 0 events. It is easy to see that both μ_{RCT} and μ_{URRH} as defined in Facts 4.1 and 5.1 respectively are admissible hyperplane measures.

Proposition B.17. *Let μ be an admissible hyperplane measure. Define the function $d_\mu : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ by $d_\mu(x, y) = \mu(\mathcal{H}_{xy})$. Then d_μ is a distance metric.*

Proof. Since $\mathcal{H}_{xx} = \emptyset$, $d_\mu(x, x) = 0$. By property (iii) above, $d_\mu(x, y) > 0$ for $x \neq y$. By symmetry of the separation relation, we have $\mathcal{H}_{xy} = \mathcal{H}_{yx}$ so $d_\mu(x, y) = d_\mu(y, x)$. To prove the triangle inequality, we claim that for any three points x, y, and z, $\mathcal{H}_{xz} \subseteq \mathcal{H}_{xy} \cup \mathcal{H}_{yz}$. Indeed, pick $H \in \mathcal{H}_{xz}$ so that H separates x and z. Suppose $H \notin \mathcal{H}_{xy}$, i.e., x and y are on the same side of H. Then it follows that H separates y and z, so $H \in \mathcal{H}_{yz}$. We then have

$$\begin{aligned} d_\mu(x, z) &= \mu(\mathcal{H}_{xz}) \\ &\leq \mu(\mathcal{H}_{xy} \cup \mathcal{H}_{yz}) \\ &\leq \mu(\mathcal{H}_{xy}) + \mu(\mathcal{H}_{yz}) \\ &= d_\mu(x, y) + d_\mu(y, z) \ . \end{aligned}$$

This concludes the proof. □

We have thus established a correspondence between admissible hyperplane measures μ and their associated distance metrics d_μ. In this language, Lemma 4.7 can be restated as $d_{\mu_{\text{RCT}}}(x, y) = ||x - y||_1$. As we will see in Lemma C.1 of the next section, $d_{\mu_{\text{URRH}}}(x, y) = \lambda ||x - y||_2$, with λ an absolute constant depending only on the dimension. We generalize Lemma 4.8 below.

Lemma B.18. *Let μ be an admissible hyperplane measure with associated distance metric $d = d_\mu$. Fix a triplet $\{x_i, x_j, x_k\}$ of input X. Then the probability $p_{i,j|k}$ that A_μ on input X separates x_i and x_j from x_k is given by*

$$p_{i,j|k} = \frac{d_{i,k} + d_{j,k} - d_{i,j}}{d_{i,j} + d_{i,k} + d_{j,k}} \ ,$$

and similarly for $p_{i,k|j}$ and $p_{j,k|i}$.

Proof. By Lemma B.2, we may assume $X = \{x_i, x_j, x_k\}$. Let E_a (resp., E_b, E_c) be the event that H separates x_i and x_j (resp. x_j and x_k, x_i and x_k). From Lemma B.3, we have $p_{i,j|k} = (\mathbb{P}(E_b) + \mathbb{P}(E_c) - \mathbb{P}(E_a))/(\mathbb{P}(E_a) + \mathbb{P}(E_b) + \mathbb{P}(E_c))$. On the other hand, by definition of $d = d_\mu$ we have $\mathbb{P}(E_a) = \lambda d_{i,j}$, $\mathbb{P}(E_b) = \lambda d_{j,k}$, and $\mathbb{P}(E_c) = \lambda d_{i,k}$ (where $\lambda = 1/\mu(\mathcal{H}_X)$ and $\mathcal{H}_X$ is the set of hyperplanes intersecting $\mathrm{Conv}(X)$). Substituting these expressions above, we arrive at

$$p_{i,j|k} = \frac{d_{i,k} + d_{j,k} - d_{i,j}}{d_{i,j} + d_{i,k} + d_{j,k}} \ ,$$

and similarly for $p_{i,k|j}$ and $p_{j,k|i}$. □

We generalize Assumption 4.4 below.

Assumption B.19. *For an arbitrary distance metric d, we say that a set X d-*satisfies *Assumption 4.4 if* $\binom{n}{3}^{-1} \sum_{i<j<k}(d_{i,j} + d_{i,k} + d_{j,k})/2 \le \max_{i,j} d_{i,j}$.

We are finally in a position to generalize Theorem 4.6.

Theorem B.20. *Let μ be an admissible hyperplane measure with associated distance metric d_μ. Then A_μ and Ins_μ satisfy the same approximation guarantees as* RCT *given in Theorem 4.6, with d_μ-based metrics instead of ℓ_1-based metrics, and where the data d_μ-satisfies Assumption 4.4 for the first two cases.*

Proof. We note that all of the computations for Theorem 4.6 with RCT have rested only on the functional form of the triangle cutting probabilities $p_{i,j|k} = \frac{d_{i,k}+d_{j,k}-d_{i,j}}{d_{i,j}+d_{i,k}+d_{j,k}}$, etc., with the $d_{i,j}$'s being the ℓ_1 distances $||x_i - x_j||_1$. Now, in the case of A_μ, Lemma B.18 gives us the same formula with d_μ replacing the ℓ_1 distance. Since we also generalize Assumption 4.4 in the first two cases, we obtain that A_μ achieves the same results of Theorem 4.6 for the case of d_μ-based metrics. □

B.3. Proofs of Theorems 4.10 and 4.11

The main tool we use is the following ℓ_1 embeddability result for ℓ_2 metrics (Ailon & Chazelle, 2009).

Theorem B.21. *Let $X \subset (\mathbb{R}^d, \ell_2)$ be a finite Euclidean metric on n points and fix distortion parameter $\epsilon > 0$ and failure probability $\delta > 0$. Pick $k = O(\log(n/\delta)/\epsilon^2)$ random vectors $r_1, \ldots, r_k$ on the unit sphere $\mathcal{S}^{d-1}$ and set $\phi(v) = (\langle v, r_1\rangle, \ldots, \langle v, r_k\rangle)$. Then with probability at least $1-\delta$, for all pairs of points $u, v \in X$,*

$$(1-\epsilon) \le \frac{||\phi(u) - \phi(v)||_1}{\alpha||u-v||_2} \le (1+\epsilon),$$

where α is a constant. The result also holds when the components of the r_i's are iid standard normals.

To translate the ℓ_1 bounds for the ℓ_2 case, we need the following perturbation inequality.

Lemma B.22. *Fix $\epsilon \in (0,1)$. Let (a,b,c) and (a',b',c') be non-negative triplets of real numbers satisfying the triangle inequality and such that $(1-\epsilon)a \le a' \le (1+\epsilon)a$, and similarly for (b,b') and (c,c'). Let $p = \frac{a+b-c}{a+b+c}$ and $p' = \frac{a'+b'-c'}{a'+b'+c'}$. Recall that these are the probabilities from Lemma 4.8 in the appendix. Then*

$$p(1-\epsilon)^2 - \epsilon \le p' \le p(1+\epsilon)^2 + \epsilon\ .$$

Proof. We prove the lower bound on p'. The proof of the upper bound is analogous.

$$\begin{aligned} p' &\ge \frac{a(1-\epsilon) + b(1-\epsilon) - c(1+\epsilon)}{(a+b+c)(1+\epsilon)} \\ &= \frac{(a+b-c)(1-\epsilon) - 2c\epsilon}{(a+b+c)(1+\epsilon)} \\ &\ge \frac{(a+b-c)(1-\epsilon)}{(a+b+c)(1+\epsilon)} - \epsilon \\ &\ge p(1-\epsilon)^2 - \epsilon, \end{aligned}$$

where we have used the triangle inequality to upper bound $2c/(a+b+c)$ by 1. □

Theorem 4.10. *Fix $\epsilon, \delta > 0$. Consider a revenue case from Table 2 (cases 1, 3, 5, 7, 8) but with the corresponding ℓ_2 metric. Let α be the corresponding approximation guarantee of* RCT *under ℓ_1 metric. Then there exists an absolute constant c such that with probability $1 - \delta$,* PRCT *with $k = c\log(n/\delta)/\epsilon^2$ achieves an expected approximation of $\alpha - \epsilon$ in the ℓ_2 metric.*

Proof. We first consider the MW Revenue objective. Theorem B.21 implies that, with probability at least $1 - \delta$, for any pair of points x_i, x_j, the following inequality holds:

$$(1-\epsilon)d(x_i, x_j) \leq d'(x_i, x_j) \leq (1+\epsilon)d(x_i, x_j)$$

where $d(\cdot)$ is the ℓ_1-distance in the original space $\mathbb{R}^d$, and $d'(\cdot)$ is the ℓ_2-distance in the projected space $\mathbb{R}^k$. Using the notion from Lemma B.22, we have

$$\begin{aligned}\mathrm{Rev}_{i,j,k}(\textsc{prct}) &= \sum_{\text{cyc}} p'_{i,j} w_{i,j} \\ &\geq \sum_{\text{cyc}} p_{i,j|k}(1-\epsilon)^2 w_{i,j} - \epsilon \sum_{\text{cyc}} w_{i,j} \\ &\geq \alpha(1-\epsilon)^2 \mathrm{Rev}_{i,j,k}(\textsc{opt}) - 3\epsilon \mathrm{Rev}_{i,j,k}(\textsc{opt}) \\ &\geq (\alpha - 5\epsilon)\mathrm{Rev}_{i,j,k}(\textsc{opt}),\end{aligned}$$

where have used the lower bound of Lemma B.22 in the second line, and the bound $\sum_{\text{cyc}} w_{i,j} \leq 3\mathrm{Rev}_{i,j,k}(\textsc{opt})$ in the third line. Replacing ϵ with $\epsilon/5$ completes the proof. The CKMM revenue case follows from the analogous bound $\sum_{\text{cyc}}(d_{i,j} + d_{i,k}) \leq 3\max_{\text{cyc}}(d_{i,j} + d_{i,k})$. □

Theorem 4.11. *Fix $\epsilon, \delta > 0$. Consider a cost case from Table 2 (cases 2, 4, 6) but with the corresponding ℓ_2 metric. Let α be the corresponding approximation guarantee of* RCT *under ℓ_1 metric and assume that the weights lie in the range $[\gamma, 1]$ for arbitrarily small but positive γ. Then there exists an absolute constant c such that with probability $1 - \delta$,* PRCT *with $k = c\log(n/\delta)/\epsilon^2$ achieves an expected approximation of $\alpha + \epsilon$ in the ℓ_2 metric.*

Proof. We prove this in the case of the MW Cost; the case of Dasgupta Cost is similar. Recalling that we assume that the weights are in $[\gamma, 1]$, in the MW Cost case, we have the bound

$$\sum_{i<j<k}(d_{i,j} + d_{i,k} + d_{j,k}) \leq A \sum_{i<j<k} \min(d_{i,j}, d_{i,k}, d_{j,k}) \,,$$

where $A = 3/\gamma$ suffices. The proof is identical to that of Theorem 4.10, with the above inequality controlling the additive perturbation introduced by Lemma B.22. In the Case of the Dasgupta cost, we instead have the bound $w_{i,j} + w_{i,k} + w_{j,k} \leq A\min(w_{i,j} + w_{i,k}, w_{i,j} + w_{j,k}, w_{i,k} + w_{j,k})$. □

C. Supplementary material for Section 5

Algorithm 3 and Algorithm 4 contain the pseudocode of the URRH algorithm. Algorithm 5 contains the pseudocode of the associated insertion procedure.

Notice that the sampling method used in Algorithm 4 is equivalent to selecting $H_{p,b}(\mathcal{S}(C))$ *without* rejection as follows. Once p is selected from the unit $(d-1)$-sphere with probability proportional to $\max_{x,x' \in C}|x \cdot p - x' \cdot p|$, the offset b is then drawn uniformly at random from the interval $[x_p^* \cdot p,\, x_p'^* \cdot p]$, where the pair of points $x_p^*, x_p'^*$ maximize $|x \cdot p - x' \cdot p|$ over all $x, x' \in C$. In fact this observation allows us to easily establish Fact 5.1.

Fact 5.1. *Fix dimension d, let $H_{u,v} = \{x \in \mathbb{R}^d \,|\, x \cdot u = v\}$, and $\mathcal{H} = \{H_{u,v} \,|\, u \in \mathcal{S}^{d-1}, v \in \mathbb{R}\}$ be the set of all hyperplanes in $\mathbb{R}^d$. Define $\mu_{\text{URRH}}(\mathcal{H}') = \int_{u \in \mathcal{S}^{d-1}} \mu_L(\{v \in \mathbb{R} \,|\, H_{u,v} \in \mathcal{H}'\} d\nu$ for $\mathcal{H}' \subset \mathcal{H}$, where μ_L is the Lebesgue measure on $\mathbb{R}$ and ν is the uniform measure on $\mathcal{S}^{d-1}$. Then $A_{\mu_{\text{URRH}}}$ (resp. $\mathrm{Ins}_{\mu_{\text{URRH}}}$) is the offline (resp. dynamic)* URRH *algorithm.*

Proof. As in the proof of Fact 4.1, it suffices to show that sampling a hyperplane H from μ_{URRH} is equivalent to the sampling done in URRH. Fix a finite set $X \subset \mathbb{R}^d$. For each $u \in \mathcal{S}^{d-1}$ define $v_u^* = \max_{x \in X} x \cdot u$ and $v_u'^* = \min_{x \in S} x \cdot u$. The set of hyperplanes that intersect X is then given by $\mathcal{H}'(X) = \bigcup_{u \in \mathbb{S}^{d-1}} \{H_{u,v} \,|\, v \in [v_u'^*, v_u^*]\}$. By definition, μ_{URRH} assigns a measure of $\int_{u \in \mathbb{S}^{d-1}} (v_u^* - v_u'^*) d\nu$ to $\mathcal{H}'(X)$. Hence the μ_{URRH_X}-probability of selecting a hyperplane with normal u is proportional to $v_u^* - v_u'^*$, which is equivalent to the direction sampled by URRH. Furthermore, since μ_{URRH} assigns Lebesgue measure to the hyperplanes with a given normal, μ_{URRH} samples the offset within $[v_u'^*, v_u^*]$ uniformly, which is equivalent to URRH's sampling as well. □

Algorithm 3 The URRH algorithm (Uniform Radial Random Hyperplane Algorithm).

```
Input ▷ Set $X = \{x_1, x_2, \ldots, x_n\} \subseteq \mathbb{R}^d$
Output ▷ HC tree $T$
Init: Create root $r$ of $T$;
/* Start the recursive divisive splitting; The procedure URRH_partition (Algorithm 4)
generates $T$ and therefore has access to its nodes */
URRH_partition($X, r$);
Return $T$ .
```

We now turn to establishing the approximation ratio guarantees for URRH as stated in Theorems 5.2 and 5.3. Key to both theorems are the ℓ_2 analogs of Lemmas 4.7 and 4.8 which we now establish for URRH.

Lemma C.1. *Let input X be contained in sphere $\mathcal{S}(X)$. Choose a cut H that intersects $\mathcal{S}(X)$ according to* URRH. *For any two points $x, y \in X$, the probability p_{xy} that H separates x and y is proportional to their ℓ_2 distance $||x - y||_2$ (and is independent of their location and orientation). In addition, in the language of Section B.2.8, we have $d_{\mu_{\text{URRH}}}(x, y) = \lambda||x - y||_2$.*

Proof. Let $\mathcal{S}(X)$ have radius r. The measure that μ_{URRH} assigns to hyperplanes intersecting $\mathcal{S}(X)$ is given by $2r \int_{u \in \mathcal{S}^{d-1}} d\nu$, where $d\nu$ is the uniform measure on the unit sphere $\mathcal{S}^{d-1}$. Let $u_{xy} = \frac{y-x}{||x-y||_2}$ be the unit vector from x to y, and let e_1 denote the canonical unit vector in the first coordinate direction. The measure assigned to hyperplanes separating x and y is given by

$$\begin{aligned} d_{\mu_{\text{URRH}}}(x, y) &= \int_{u \in \mathcal{S}^{d-1}} |y \cdot u - x \cdot u| \, d\nu \\ &= \int_{u \in \mathcal{S}^{d-1}} |(y - x) \cdot u| \, d\nu \\ &= \int_{u \in \mathcal{S}^{d-1}} ||y - x||_2 |u_{xy} \cdot u| \, d\nu \\ &= ||y - x||_2 \int_{u \in \mathcal{S}^{d-1}} |e_1 \cdot u| \, d\nu, \end{aligned}$$

where the last line follows from the rotational invariance of $d\nu$ (more explicitly, we may make the substitution $u \leftarrow Ou$, where O is an orthonormal matrix that sends e_1 to u_{xy}). We thus have

$$p_{xy} = ||x - y||_2 \frac{\int_{u \in \mathcal{S}^{d-1}} |e_1 \cdot u| d\nu}{2r \int_{u \in \mathcal{S}^{d-1}} d\nu},$$

which completes the proof. □

Note in connection with Lemma C.5 that the proportionality constant in the above displayed equation is $\Theta\left(\frac{1}{r\sqrt{d}}\right)$.

Given any triplet $\{x_i, x_j, x_k\}$ from the input set X, we denote by $p_{i,j|k}$ the probability that URRH separates x_i and x_j from x_k.

Lemma C.2. *Given any triplet $\{x_i, x_j, x_k\}$ from the input set X, with $a = d_{i,j}$, $b = d_{j,k}$, $c = d_{k,i}$ denoting their ℓ_2 distances, we have*

$$p_{i,j|k} = \frac{b + c - a}{a + b + c}$$

(and similarly for $p_{i,k|j}$ and $p_{j,k|i}$).

Algorithm 4 URRH_partition(C, v)

```
Input ▷ subset $C = \{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}$ of $k \geq 1$ items ;  node $v$ of $T$ .

/* If the input subset is a singleton, then create a leaf with that point and return */
if $k = 1$ then
  • Set $v$ as a leaf of $T$;
  • Associate $v$ with $x_{i_1}$;
  • Return ;

/* Keep generating new random hyperplanes until the input subset is cut */
cut ← False;
while cut = False do
  • /* Select a random direction */
    $p \leftarrow$ point selected uniformly at random from the unit $(d-1)$-sphere;
  • /* Use any chosen sphere containing the whole input */
    $\mathcal{S}(C) \leftarrow (d-1)$-sphere containing all input points in $C$;
  • $r \leftarrow$ radius of $\mathcal{S}(C)$;
  • $c \leftarrow$ center of $\mathcal{S}(C)$;
  • /* Select a random hyperplane with the chosen direction and intersecting $\mathcal{S}(C)$ */
    $b \leftarrow$ real value selected uniformly at random from interval $[c \cdot p - r,\ c \cdot p + r]$;
  • $H_{p,b}(\mathcal{S}(C)) \leftarrow \{x \in \mathbb{R}^d \ : \ x \cdot p = b\}$ ;
  • /* If hyperplane $H_{p,b}(\mathcal{S}(C))$ cuts set $C$ then recurse on the two subsets of $C$ split by
    $H_{p,b}(\mathcal{S}(C))$; otherwise reject $H_{p,b}(\mathcal{S}(C))$ */
    if $\exists j', j'' \in [k] \ : \ \mathrm{sgn}\left((x_{i_{j'}} - c) \cdot p\right) \neq \mathrm{sgn}\left((x_{i_{j''}} - c) \cdot p\right)$ then
      – /* Partition $C$ according to the cutting hyperplane $H_{p,b}(\mathcal{S}(C))$ into $C'$ and $C''$ */
        $C' \leftarrow \{x_{i_j} \in C \ : \ (x_{i_j} - c) \cdot p \geq 0\}$;      $C'' \leftarrow C \setminus C'$;
      – /* Generate the children of input node $v$ */
        Create the left child $v'$ of $v$;        Create the right child $v''$ of $v$;
      – /* Generate the children of input node $v$ and recurse */
        URRH_partition$(C', v')$;       URRH_partition$(C'', v'')$;
      – /* The input point subset is now partitioned; Return */
        cut ← True;
```

Proof. Let E_a be the event that the first cut of URRH that splits the triplet separates x_i from x_j (and similarly define E_b and E_c). From Lemma B.3, we have $p_{i,j|k} = (\mathbb{P}(E_b) + \mathbb{P}(E_c) - \mathbb{P}(E_a))/(\mathbb{P}(E_a) + \mathbb{P}(E_b) + \mathbb{P}(E_c))$. On the other hand, by Lemma C.1, we have $\mathbb{P}(E_a) = \lambda a$, $\mathbb{P}(E_b) = \lambda b$, and $\mathbb{P}(E_c) = \lambda c$. Substituting these expressions above, we conclude that

$$p_{i,j|k} = \frac{b + c - a}{a + b + c} \,,$$

and by symmetry

$$p_{i,k|j} = \frac{a + b - c}{a + b + c} \,, \qquad p_{j,k|i} = \frac{a + c - b}{a + b + c} \,,$$

as claimed.

□

Algorithm 5 Insert operation for $\mathrm{Ins}_{\mu_{\text{URRH}}}$ for URRH.

```
Input ▷ Tree $T(X)$ on points $X = \{x_1, x_2, \ldots, x_n\} \subseteq \mathbb{R}^d$, and new point $x_{n+1}$
Output ▷ Tree $T(X \cup \{x_{n+1}\})$ on points $X \cup \{x_{n+1}\}$

• /* If the new point is the first one added to $T$, create the root of $T$ and return */
  If $X = \emptyset$ then return Node$(x_{n+1})$;
• $Y \leftarrow X \cup \{x_{n+1}\}$
• /* Keep generating new random hyperplanes $H$ until $Y$ is cut */.
  $\mathcal{S}(Y) \leftarrow$ sphere enclosing $Y$.
• $r \leftarrow$ radius of $\mathcal{S}(Y)$;
• $c \leftarrow$ center of $\mathcal{S}(Y)$;
  cut $\leftarrow$ False;
  while cut $=$ False do
    – /* Select a random direction */
      $p \leftarrow$ point selected uniformly at random from the unit $(d-1)$-sphere; /* Select a random hyperplane
      with the chosen direction and intersecting $\mathcal{S}(Y)$ */
      $b \leftarrow$ real value selected uniformly at random from interval $[c \cdot p - r,\, c \cdot p + r]$;
    – $H \leftarrow \{x \in \mathbb{R}^d \,:\, x \cdot p = b\}$ ;
    – /* If hyperplane $H$ separate $Y$ then use this; otherwise retry */;
      if $\exists j', j'' \in [n+1] \,:\, \mathrm{sgn}\Big((x_{i_{j'}} - c) \cdot p\Big) \neq \mathrm{sgn}\Big((x_{i_{j''}} - c) \cdot p\Big)$ then cut $\leftarrow$ True;
• /* If $H$ separates $X$ and $x_{n+1}$, add $x_{n+1}$ and return */
  If $\forall i \leq n$ $\mathrm{sgn}(p \cdot x_i - b) \neq \mathrm{sgn}(p \cdot x_{n+1} - b)$ then return $T(X) \wedge$ Node$(x_{n+1})$;
  /* Else let $H'$ be the hyperplane previously generated in $T$ to split $X$ into the two
  subsets $X^+$ and $X^-$ of $X$ */
  Else $H' \leftarrow$ hyperplane $\{x \in \mathbb{R}^d : p' \cdot x = b', \|p'\| = 1\}$ cutting $T(X)$;
  $X^- \leftarrow \{x \in X : p' \cdot x - b' \leq 0\}$;
  $X^+ \leftarrow \{x \in X : p' \cdot x - b' > 0\}$;
• /* Recurs on the subset obtained by cutting $X$ with $H'$ where all points are on the same
  side w.r.t $x_{n+1}$ */
  If $p' \cdot x_{n+1} - b' \leq 0$ then return $\mathrm{Ins}_{\mu_{\text{RCT}}}(T(X^-), x_{n+1}) \wedge T(X^+)$;
  Else return $T(X^-) \wedge \mathrm{Ins}_{\mu_{\text{RCT}}}(T(X^+), x_{n+1})$;
```

We now focus our attention on the proofs of Theorems 5.2 and 5.3.

Proof of Theorem 5.2. Recall the definition of μ_{URRH} from Fact 5.1. We use the notation μ_L for Lebesgue measure on $\mathbb{R}$ and $H_{u,v} = \{x \in \mathbb{R}^d \,|\, x \cdot u = v\}$. It is easy to see that μ_{URRH} satisfies properties (i)-(iv) of Definition B.16 and hence is an admissible hyperplane measure. Indeed, (i) is immediate. For (ii), let X be a finite set and note that $\mu_L(\{v \in \mathbb{R} \,|\, H_{u,v} \in \mathcal{H}_X\}$ is bounded over $u \in \mathcal{S}^{d-1}$. (iii) follows from Lemma C.1, and (iv) follows from $\mu_L(\{v \in \mathbb{R} \,|\, H_{u,v} \ni x\} = 0$ for $x \in \mathbb{R}^d$ and $u \in \mathcal{S}^{d-1}$. By Lemma C.1, $d_{\mu_{\text{URRH}}}(x, y) = \lambda ||x - y||_2$. Finally, given the scale invariance of the approximation guarantees with respect to the distance function, the proof of Theorem 5.2 follows from Theorem B.20. □

We then turn to the proof of Theorem 5.3. Without loss of generality, throughout the analysis we normalize distances so as to get $D = 1$. This way our weights are now simply

$$w_{i,j} := 1 - d_{i,j} \ ,$$

being $d_{i,j}$ the normalized Euclidean distance between x_i and x_j. Clearly, this normalization does not affect the algorithm's approximation ratio.

Given any triplet $\{x_i, x_j, x_k\}$ of points in the input set X and any randomized algorithm ALG, we now define the expected revenue of $\{x_i, x_j, x_k\}$ by

$$\mathrm{Rev}_{i,j,k}(\textsc{alg}) := p_{i,j|k}(1-d_{i,j}) + p_{j,k|i}(1-d_{j,k}) + p_{k,i|j}(1-d_{k,i}) \,.$$

We also define a lower bound on ALG's expected approximation ratio as

$$Apx_{i,j,k}(\textsc{alg}) := \frac{Rev_{i,j,k}(\textsc{alg})}{\max\{1-d_{i,j}, 1-d_{j,k}, 1-d_{k,i}\}} = \frac{Rev_{i,j,k}(\textsc{alg})}{1-\min\{d_{i,j}, d_{j,k}, d_{k,i}\}} \,.$$

Also, let $p_{i,j,k}$ be the average length of the side of the triangle having x_i, x_j and x_k as its vertices, i.e., $p_{i,j,k} = \frac{1}{3}(d_{i,j} + d_{j,k} + d_{k,i})$. For simplicity, when the triplet $\{x_i, x_j, x_k\}$ is clear from the surrounding context, or when a statement involving $p_{i,j,k}$ holds for any triplet, we shall abbreviate $p_{i,j,k}$ by p.

Finally, we define the piecewise function $\phi(p) : [0,1] \to [0,1]$ as follows (see Figure 3):

$$\phi(p) := [\![p=0]\!] + [\![0 < p < \tfrac{8}{9}]\!]\,\frac{2p + 2\sqrt{1-p} - 2}{p} + [\![\tfrac{8}{9} \le p < 1]\!]\,\frac{4-3p}{3p} + [\![p=1]\!] \,,$$

where $[\![\cdot]\!]$ denotes the indicator function of the predicate at argument.

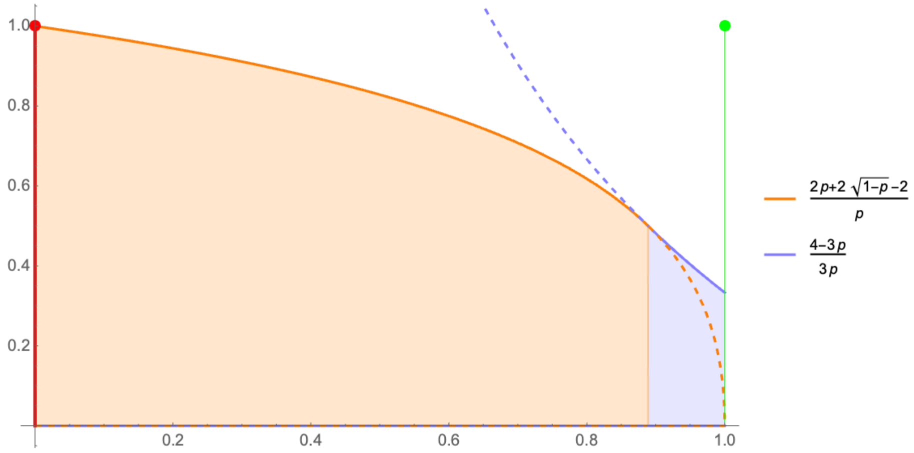

Figure 3: Plot of ϕ as a function of $p \in [0,1]$. Orange and blue solid lines, together with the red and green dots, represent the function plot for $p \in [0,1]$. Dashed lines plot the parts of function $\frac{2p+2\sqrt{1-p}-2}{p}$ for $p \ge \frac{8}{9}$, and function $\frac{4-3p}{3p}$ for $p < \frac{8}{9}$, i.e., outside the intervals they are associated with in the definition of $\phi(p)$.

We have the following lemma.

Lemma C.3. *Given any triplet $\{x_i, x_j, x_k\}$ in the input set X, we have $Apx_{i,j,k}(\textsc{urrh}) \ge \phi(p_{i,j,k})$.*

Proof. Let t be the triangle having x_i, x_j and x_k as vertices, and denote by a, b and c the lengths of its sides, with $a \le b \le c$. We first analyze the special cases $p_{i,j,k} = 0$ and $p_{i,j,k} = 1$, for which we must have $a = b = c = 0$ and $a = b = c = 1$, respectively. These two cases are easily handled by observing that, more generally, when t is an equilateral triangle ($a = b = c$), we have $Rev_{i,j,k}(\textsc{urrh}) = 1 - \min(d_{i,j}, d_{j,k}, d_{k,i})$, so that $Apx_{i,j,k}(\textsc{urrh}) = 1$.

In the case when $p_{i,j,k} \neq 1$, by the definition of $Apx_{i,j,k}$, we can combine Lemma C.2 with $\min\{d_{i,j}, d_{j,k}, d_{k,i}\} = \min\{a, b, c\} = a$ to obtain

$$\begin{aligned} Apx_{i,j,k}(\text{URRH}) &= (1-a)^{-1} \cdot \left(\frac{b+c-a}{a+b+c} \cdot (1-a) + \frac{a+c-b}{a+b+c} \cdot (1-b) + \frac{a+b-c}{a+b+c} \cdot (1-c) \right) \\ &= \frac{b+c-a}{a+b+c} + \frac{a+c-b}{a+b+c} \cdot \left(\frac{1-b}{1-a} \right) + \frac{a+b-c}{a+b+c} \cdot \left(\frac{1-c}{1-a} \right) . \end{aligned} \tag{2}$$

Our goal is now to minimize the above expression under the constraint that $p = \frac{1}{3}(a+b+c)$, that is, we would like to compute

$$f(p) := \min_{\substack{0 \leq a \leq b \leq c \leq 1: \\ 0 \neq a+b+c = 3p \neq 3}} Apx_{i,j,k}(\text{URRH}) . \tag{3}$$

First of all, notice that for any triplet with $0 < a \leq b \leq c < 1$, if we replace b and c by their arithmetic average $\frac{b+c}{2}$, the value of $Apx_{i,j,k}(\text{URRH})$ decreases whenever $b \neq c$, while p does not change. Let t' be another triangle having $x_{i'}, x_{j'}$ and $x_{k'}$ as vertices, and denote by a', b' and c' its three side lengths, with $a' \leq b' \leq c'$. Setting $a' = a$, and $b' = c' = \frac{b+c}{2}$, we get $p_{i,j,k} = p_{i',j',k'}$. On the other hand, in this case we also have

$$\begin{aligned} & Apx_{i',j',k'}(\text{URRH}) - Apx_{i,j,k}(\text{URRH}) \\ &\quad = \frac{\frac{b+c}{2} + \frac{b+c}{2} - a}{a + \frac{b+c}{2} + \frac{b+c}{2}} + \frac{a + \frac{b+c}{2} - \frac{b+c}{2}}{a + \frac{b+c}{2} + \frac{b+c}{2}} \cdot \frac{1 - \frac{b+c}{2}}{1-a} + \frac{a + \frac{b+c}{2} - \frac{b+c}{2}}{a + \frac{b+c}{2} + \frac{b+c}{2}} \cdot \frac{1 - \frac{b+c}{2}}{1-a} \\ &\quad\quad - \left(\frac{b+c-a}{a+b+c} + \frac{a+c-b}{a+b+c} \cdot \left(\frac{1-b}{1-a} \right) + \frac{a+b-c}{a+b+c} \cdot \left(\frac{1-c}{1-a} \right) \right) \\ &\quad = -\frac{(1-b)(c-b+a)}{(1-a)(c+b+a)} - \frac{(1-c)(-c+b+a)}{(1-a)(c+b+a)} + \frac{2\left(\frac{-c-b}{2}+1\right)\left(\frac{c+b}{2} + \frac{-c-b}{2} + a\right)}{(1-a)(c+b+a)} \\ &\quad = \frac{(c-b)^2}{(a-1)(a+b+c)} , \end{aligned}$$

which is always negative because we assumed $a < 1$ and $a+b+c > 0$, except for the case $b = c$ in which case it is equal to 0.

Hence we always have $Apx_{i',j',k'}(\text{URRH}) \leq Apx_{i,j,k}(\text{URRH})$ while $p_{i,j,k} = p_{i',j',k'}$, which implies that we can find the minimizer $f(p)$ assuming $b = c$. Replacing b by c in (2), Equation (3) becomes

$$f(p) = \min_{\substack{0 \leq a \leq c \leq 1: \\ 0 \neq a+2c = 3p \neq 3}} \frac{4ac - 2c - a^2 - a}{(a-1)(2c+a)} .$$

Replacing a by $3p - 2c$ in the above expression gives

$$f(p) = \min_{\substack{0 < p \leq c \leq 1: \\ p \neq 1}} \frac{4c^2 - 8cp + 3p^2 + p}{2cp - 3p^2 + p} . \tag{4}$$

Now we can easily minimize $f(p)$ by setting the derivative

$$\frac{\partial f}{\partial c} = \frac{2\left(p(9p-5) + c(4-12p) + 4c^2\right)}{(-3p+2c+1)^2 \, p}$$

to 0, which finally gives $c = \dfrac{2\sqrt{1-p} + 6p - 2}{4}$ for $0 < p < \frac{8}{9}$ and $c = 1$ for $\frac{8}{9} \leq p < 1$. Replacing back into (4) the two values of c we just found, we see that $f(p)$ is equal to $\frac{2p + 2\sqrt{1-p} - 2}{p}$ for $0 < p < \frac{8}{9}$, and to $\frac{4-3p}{3p}$ for $\frac{8}{9} \leq p < 1$, which gives $f(p) = \phi(p)$ for all $p \in (0, 1)$. Combining with the initial observation for the two extreme cases $p = 0$ and $p = 1$ concludes the proof. □

Lemma C.4. *Let $\mathcal{S}_d$ be the collection of all sets of $n > d+1$ points in the d-dimensional Euclidean space with unitary maximum pairwise distance. Let $C = \{x_1, \dots, x_n\} \in \mathcal{S}_d$ be any such set. Then the average square distance*

$$\binom{n}{2}^{-1} \sum_{1 \le i < j \le n} \|x_i - x_j\|_2^2$$

between pairs of points in C is upper bounded by $\left(1 + \frac{1}{n-1}\right)\left(1 - \frac{1}{d+1}\right)$.

Proof. Without loss of generality, let the center of the circumsphere $\mathcal{S}(C)$ of the input points C be the origin $\mathbf{0}$ of $\mathbb{R}^d$, and let r be the radius of $\mathcal{S}(C)$. Let $\mathrm{Conv}(C)$ be the convex hull of C. Finally, denote by $\widetilde{C} \subseteq C$ the subset of all points of C belonging to the circumsphere $\mathcal{S}(C)$, i.e., all points $x \in C$ such that $\|x\| = r$.

We must have $\mathbf{0} \in \mathrm{Conv}(\widetilde{C})$, since if $\mathbf{0} \notin \mathrm{Conv}(\widetilde{C})$, then $\mathcal{S}(C)$ would not be the circumsphere of the points of C, for there would exist a sphere $\mathcal{S}' \neq \mathcal{S}(C)$ containing all points of C and having a radius strictly smaller than r.

Applying Carathodory's theorem, which states that every $x \in \mathrm{Conv}(P)$ with $P \subset \mathbb{R}^d$ can be written as the convex combination of at most $d+1$ points of P, we know that $\mathbf{0} = \sum_{i=1}^{n'} a_i y_i$, for some $y_1, y_2, \dots, y_{n'} \in \widetilde{C}$, and $a_1, a_2, \dots, a_{n'} \ge 0$ with $\sum_{i=1}^{n'} a_i = 1$, and $n' \le d+1$.

Thus we have

$$0 = \left\| \sum_{i=1}^{n'} a_i y_i \right\|^2 = \left(\sum_{1 \le i,j \le n' : i \neq j} a_i\, a_j (y_i \cdot y_j) \right) + \left(\sum_{k=1}^{n'} a_k^2 \|y_k\|^2 \right) . \tag{5}$$

Let i^*, j^* the two indices minimizing $y_i \cdot y_j$ over all pairs $i, j \in \{1, 2, \dots, n'\}$ such that $i \neq j$. We can now write

$$\begin{aligned} \left(\sum_{1 \le i,j \le n' : i \neq j} a_i\, a_j \right) (y_{i^*} \cdot y_{j^*}) &\le \sum_{1 \le i,j \le n' : i \neq j} a_i\, a_j (y_i \cdot y_j) \\ &= -\sum_{k=1}^{n'} a_k^2 \|y_k\|^2 \qquad (6) \\ &= -r^2 \sum_{k=1}^{n'} a_k^2 , \qquad (7) \end{aligned}$$

where we plugged the result of (5) into (6).

Applying the Cauchy-Schwartz inequality we have

$$\begin{aligned} \left(\sum_{1 \le i,j \le n' : i \neq j} a_i\, a_j \right)^2 &\le \left(\sum_{1 \le i,j \le n' : i \neq j} a_i^2 \right) \left(\sum_{1 \le i,j \le n' : i \neq j} a_j^2 \right) \\ &= \left((n'-1) \sum_{i=1}^{n'} a_i^2 \right) \left((n'-1) \sum_{j=1}^{n'} a_j^2 \right) \\ &= (n'-1)^2 \left(\sum_{i=1}^{n'} a_i^2 \right)^2 , \end{aligned}$$

from which we obtain

$$\frac{\sum_{i=1}^{n'} a_i^2}{\sum_{1 \le i,j \le n' : i \neq j} a_i\, a_j} \ge \frac{1}{n'-1} . \tag{8}$$

Combining (7) with (8) gives

$$y_{i^*} \cdot y_{j^*} \le -r^2 \frac{\sum_{k=1}^{n'} a_k^2}{\sum_{1 \le i,j \le n' : i \ne j} a_i \, a_j} \le -r^2 \frac{1}{n' - 1} \,. \tag{9}$$

We now use the assumption $\max_{i,j} \|x_i - x_j\| = 1$ over all $i, j \in \{1, 2, \dots, n\}$, which implies that $\|y_k - y_\ell\| \le 1$ for all $k, \ell \in \{1, 2, \dots, n'\}$. Hence we can write

$$\begin{aligned} 1 &\ge \|y_{i^*} - y_{j^*}\|^2 \\ &= \|y_{i^*}\|^2 + \|y_{j^*}\|^2 - 2(y_{i^*} \cdot y_{j^*}) \\ &= 2r^2 - 2(y_{i^*} \cdot y_{j^*}) \\ &\ge 2r^2 + 2r^2 \frac{1}{n' - 1} \qquad (10) \\ &= r^2 \frac{2n'}{n' - 1} \,, \end{aligned}$$

where (10) follows from (9).

Using the above inequalities and recalling that $n' \le d + 1$ implies

$$r^2 \le \frac{n' - 1}{2n'} \le \frac{d}{2(d+1)} \,.$$

We can now apply the above upper bound on r^2 to bound from below the average square norm of x_i over all $i \in \{1, 2, \dots, n\}$ as follows:

$$\frac{\sum_{i=1}^{n} \|x_i\|^2}{n} \le \max_{1 \le i \le n} \|x_i\|^2 = r^2 \le \frac{d}{2(d+1)} \,. \tag{11}$$

Finally, we conclude the proof by upper bounding the average square distance between any two points x_i and x_j as follows:

$$\begin{aligned} \binom{n}{2}^{-1} \sum_{1 \le i < j \le n} \|x_i - x_j\|^2 &= \left(\frac{n}{n-1}\right) \cdot \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \|x_i - x_j\|^2}{n^2} \\ &\le \left(\frac{n}{n-1}\right) \cdot \frac{2}{n} \sum_{k=1}^{n} \|x_k\|^2 \\ &= \left(\frac{n}{n-1}\right) \cdot 2r^2 \\ &\le \left(\frac{n}{n-1}\right) \cdot \frac{d}{d+1} \qquad (12) \\ &= \left(1 + \frac{1}{n-1}\right)\left(1 - \frac{1}{d+1}\right) \,, \end{aligned}$$

where in (12) we used (11). □

We restate here for convenience Theorem 5.3 from Section 5.

Theorem C.5. *Given any input set $X = \{x_1, \dots, x_n\} \subseteq \mathbb{R}^d$, with $d > 3$, the approximation ratio $\frac{\mathbb{E}[\mathrm{Rev}_S(\textsc{urrh}(X))]}{\mathrm{Opt}_{\mathrm{Rev}_S}}$ is lower bounded by where $g(d,n)$ is a function of d and n such that $g(d,n) > 0$ for all $n > \frac{605}{116} d \approx 5.22d$. In particular, if $n \ge \left(9 + \frac{38}{d - 3.98}\right) d$ and $d > 3$, we have*

$$\mathbb{E}[\mathrm{Rev}_S(\textsc{urrh}(X))] \ge \left(\frac{1}{3} + \frac{1}{31 d^3}\right) \mathrm{Opt}_{\mathrm{Rev}_S} \,.$$

In the above, the expectation is over the internal randomization of URRH.

Proof. Let $\mathcal{T} = \{\{x_i, x_j, x_k\} : x_i, x_j, x_k \in X,\ i < j < k\}$ be the set of triplets on X. For any given triplet $t = \{x_i, x_j, x_k\} \in \mathcal{T}$, let $p(t)$ be equal to $p_{i,j,k} := \frac{1}{3}(\|x_i - x_j\|_2 + \|x_j - x_k\|_2 + \|x_k - x_i\|_2)$, which denotes the average side length of the triangle having x_i, x_j and x_k as its vertices. Finally, let $L(X) = \sum_{t \in \mathcal{T}} p(t)$.

First of all, observe that averaging $p(t)$ over all $t \in \mathcal{T}$ yields the average distance between any two items in X. This is because

$$\begin{aligned}
\binom{n}{3}^{-1} \sum_{t \in \mathcal{T}} p(t) &= \frac{6}{n(n-1)(n-2)} \sum_{1 \le i < j < k \le n} \tfrac{1}{3}(d_{i,j} + d_{j,k} + d_{k,i}) \qquad (13)\\
&= \frac{2}{n(n-1)(n-2)} \sum_{1 \le i < j \le n} \left(d_{i,j} + \sum_{1 \le k \le n:\, k \ne i,j} (d_{j,k} + d_{k,i}) \right)\\
&= \frac{2}{n(n-1)(n-2)} \sum_{1 \le i < j \le n} ((n-2) d_{i,j})\\
&= \binom{n}{2}^{-1} \sum_{1 \le i < j \le n} d_{i,j} \ .
\end{aligned}$$

Let $\mu(d,n) := \sqrt{\left(1 + \frac{1}{n-1}\right)\left(1 - \frac{1}{d+1}\right)}$. Applying now the Cauchy-Schwartz inequality to the bound of Lemma C.4, we obtain $L(X) \le \mu(d,n)\binom{n}{3}$ for all $d > 3$. Let $p^* \in (\mu(d,n), 1)$ a threshold value (that we will choose later to optimize the URRH approximation factor) used to split $\mathcal{T}$ into two sets: The bad triplets $B_{p^*} = \{t \in \mathcal{T} : p(t) > p^*\}$ and the good triplets $G_{p^*} = \{t \in \mathcal{T} : p(t) \le p^*\}$. Define the following quantities:

- $UB = \sum_{t \in B_{p^*}} \mathbb{E}[\mathrm{Rev}(\textsc{urrh}(t))]$,
- $UG = \sum_{t \in G_{p^*}} \mathbb{E}[\mathrm{Rev}(\textsc{urrh}(t))]$,
- $OB = \sum_{t \in B_{p^*}} \mathrm{Rev}(\textsc{opt}(t))$,
- $OG = \sum_{t \in G_{p^*}} \mathrm{Rev}(\textsc{opt}(t))$,

where we denote by $\mathrm{Rev}(\textsc{alg}(t))$ the local revenue of any algorithm ALG on the given triplet $t \in \mathcal{T}$.

Our goal is to minimize $\alpha = \frac{UB+UG}{OB+OG}$. We will use the following bounds:

(i) $UB \ge \frac{1}{3} OB$,

(ii) $UG \ge \frac{4-3p^*}{3p^*} OG$,

(iii) $OB \le \left(\frac{\mu(n,d)}{(p^*)}\right)^2 \binom{n}{3}$,

(iv) $OG \ge (1-p^*)\left(1 - \left(\frac{\mu(n,d)}{(p^*)}\right)^2\right)\binom{n}{3}$.

In the above, (i) can be derived by combining Lemma C.3 with the fact that, as immediately seen from the definition of function $\phi(\cdot)$, we have $\phi(p^*) \ge \frac{1}{3}$ for all $p^* \in [0,1]$.

(ii) follows from combining Lemma C.3 with the definition of the range of values for p^*, which is $(\mu(d,n), 1)$, the inequality $\mu(d,n) > \frac{8}{9}$ holding for all $d > 3$, and the definition of function $\phi(\cdot)$.

(iii) and (iv) follow from combining Lemma C.4, with the second moment Markov's inequality applied to $L(X)$. This is coupled for (iv) with the observation that for any triplet t, $\mathrm{Rev}(\textsc{opt}(t)) \in [1 - p(t), 1]$. This is because, for any t, the length of any side of a triangle having as vertices the points of t cannot be larger than $p(t)$.

From (i) and (ii), we have

$$\alpha \geq \frac{\frac{1}{3}OB + \frac{4-3p^*}{3p^*}OG}{OB+OG} \ .$$

For fixed OB, this is increasing in OG and hence it is minimized for the smallest possible OG. Combining (iii) with (iv), we obtain $OG \geq (1-p^*)\left(\left(\frac{(p^*)}{\mu(d,n)}\right)^2 - 1\right) OB$. Plugging this into the above expression we get

$$\alpha \geq f(d,p^*) := \frac{\frac{1}{3} + \frac{4-3p^*}{3p^*}(1-p^*)\left(\left(\frac{(p^*)}{\mu(d,n)}\right)^2 - 1\right)}{1 + (1-p^*)\left(\left(\frac{(p^*)}{\mu(d,n)}\right)^2 - 1\right)} \ .$$

In the above expression, we now set $p^* = 1 - \frac{1}{3d}$, which is in the interval $(\mu(d,n), 1)$ for all $d > 3$. To conclude the proof it is sufficient to show that $h(d) := f\left(d, 1 - \frac{1}{3d}\right) - \frac{1}{3} > 0$ for all *real*[9] values $d \geq 4$ and all $n \geq \frac{605}{116}d$. Thus, we want to evaluate for $d \geq 4$ the sign of

$$\begin{aligned}
g(d,n) &= \frac{27nd^5 - 9\,(n+1)\,d^4 + 3\,(n-4)\,d^3 - 2\,(n-1)\,d^2 - 4\,(n-1)\,d + n - 1}{(1-3d)^2\left(9nd^3 + 3\,(n-1)\,d^2 + 2\,(n-1)\,d - n + 1\right)} - \frac{1}{3} \\
&= \frac{4\left(-9d^3 + 3\,(n-1)\,d^2 - 5\,(n-1)\,d + n - 1\right)}{3\,(1-3d)^2\left(9nd^3 + 3\,(n-1)\,d^2 + 2\,(n-1)\,d - n + 1\right)} \\
&= \frac{n\left(12d^2 - 20d + 4\right) - 36d^3 - 12d^2 + 20d - 4}{n\left(243d^5 - 81d^4 + 27d^3 - 54d^2 + 24d - 3\right) - 81d^4 + 54d^2 - 24d + 3} \ .
\end{aligned}$$

Setting the numerator and the denominator of the above expression greater than 0 we respectively obtain

$$n > \frac{36d^3 + 12d^2 - 20d + 4}{12d^2 - 20d + 4} = \frac{(d+1)(3d-1)^2}{3d^2 - 5d + 1} = \frac{9d^3 + 3d^2 - 5d + 1}{3d^2 - 5d + 1} \ ,$$

and

$$n > \frac{81d^4 - 54d^2 + 24d - 3}{243d^5 - 81d^4 + 27d^3 - 54d^2 + 24d - 3} = \frac{3d^2 + 2d - 1}{9d^3 + 3d^2 + 2d - 1} \ .$$

Since for all $d \geq 4$ we have

$$\frac{9d^3 + 3d^2 - 5d + 1}{3d^2 - 5d + 1} > \frac{3d^2 + 2d - 1}{9d^3 + 3d^2 + 2d - 1} \ ,$$

we can conclude that it is sufficient to have

$$n > \frac{9d^3 + 3d^2 - 5d + 1}{3d^2 - 5d + 1} \ .$$

Finally, for all $d \geq 4$, we have $\frac{605}{116}d \geq \frac{9d^3+3d^2-5d+1}{3d^2-5d+1}$. This is because the first derivative of $\frac{1}{d}\,\frac{9d^3+3d^2-5d+1}{3d^2-5d+1}$ is $-\frac{54d^4-48d^3+31d^2-10d+1}{d^2(3d^2-5d+1)^2}$, which is negative for all $d > 3$, This in turn implies that the maximum in the region $d \in [4, \infty)$ can be simply obtained by plugging $d = 4$ into $\frac{1}{d}\,\frac{9d^3+3d^2-5d+1}{3d^2-5d+1}$, which gives $\frac{605}{116}$.

We now continue with the second part of the proof.

[9] Although we are interested in the case $d \in \mathbb{N}$, in this part of the analysis we treat function h by extending its domain to real numbers.

Let now $\psi(d,n) := f\left(d, 1-\frac{1}{3d}\right) - \frac{1}{3} - \frac{1}{31d^3} = g(d,n) - \frac{1}{31d^3}$. To conclude the proof, it is sufficient to show that $\psi(d,n) \geq 0$ for all $d \geq 4$ and $n \geq \left(9 + \frac{38}{d-3.98}\right) d$. We have

$$\psi(d,n) = \frac{\left(129d^5 - 539d^4 + 97d^3 + 54d^2 - 24d + 3\right) n - 1116d^6 - 372d^5 + 701d^4 - 124d^3 - 54d^2 + 24d - 3}{\left(7533d^8 - 2511d^7 + 837d^6 - 1674d^5 + 744d^4 - 93d^3\right) n - 2511d^7 + 1674d^5 - 744d^4 + 93d^3} \,.$$

We will focus on the case in which both the numerator $\psi_N(d,n)$ and the denominator $\psi_D(d,n)$ of the above expression are positive. We can easily show that both the terms multiplying n in $\psi_N(d,n)$ and $\psi_D(d,n)$ are positive for all $d \geq 4$. Indeed, for $\psi_N(d,n)$ we have $129d^5 - 539d^4 + 97d^3 + 54d^2 - 24d + 3$, which is larger than $(129d^4 - 539d^3 + 97d^2 + 54d - 24)d$, which in turn is positive for all $d \geq 4$. For $\psi_D(d,n)$, the expression multiplying n can be split into two terms: **(i)** $7533d^8 - 2511d^7 + 837d^6 - 1674d^5$ which is equal to d^5 times $7533d^3 - 2511d^2 + 837d - 1674$, which in turn is positive for all $d \geq 1$, and **(ii)** $744d^4 - 93d^3 = (744d - 93)d^3$, which is positive for all $d \geq 1$. Thus, when $d \geq 4$, both $\psi_N(d,n)$ and $\psi_D(d,n)$ increase as n grows. Hence, it is sufficient to evaluate the sign of both the numerator and the denominator of

$$\psi\left(d, \left(9 + \frac{38}{d-3.98}\right) d\right) =$$

$$= \frac{\left(129d^5 - 539d^4 + 97d^3 + 54d^2 - 24d + 3\right)\left(9 + \frac{38}{d - \frac{199}{50}}\right) d - 1116d^6 - 372d^5 + 701d^4 - 124d^3 - 54d^2 + 24d - 3}{\left(7533d^8 - 2511d^7 + 837d^6 - 1674d^5 + 744d^4 - 93d^3\right)\left(9 + \frac{38}{d - \frac{199}{50}}\right) d - 2511d^7 + 1674d^5 - 744d^4 + 93d^3}$$

$$= \frac{2250d^7 - 25005d^6 + 93977d^5 - 110826d^4 + 17062d^3 + 10680d^2 - 4599d + 597}{93d^3\left(3d-1\right)^2\left(4050d^5 + 2331d^4 + 1077d^3 + 265d^2 + 339d - 199\right)} \,.$$

In the last expression, the first part of numerator $(2250d^4 - 25005d^3 + 93977d^2 - 110826d + 17062)d^3$ is positive for all $d \geq 2$, and its second part $10680d^2 - 4599d + 597$ is always positive for all d. The first part of the denominator $93d^3\left(3d-1\right)^2$ is positive for all integer values of d. The last term of the denominator $4050d^5 + 2331d^4 + 1077d^3 + 265d^2 + 339d - 199$ can be split into its first part $4050d^5 + 2331d^4 + 1077d^3 = (4050d^2 + 2331d + 1077)d^3$ which is always positive for all $d \geq 1$, and the last part $265d^2 + 339d - 199$, which is positive for all $d \geq 1$. Hence, we can finally conclude that $\psi(d,n) \geq 0$ for all $d \geq 4$ and $n \geq \left(9 + \frac{38}{d-3.98}\right) d$, as claimed. □

Lemma C.5. *At any recursive step of* URRH*, the probability of accepting a randomly selected hyperplane is lower bounded by*

$$\max\left\{\frac{D_C}{r}\sqrt{\frac{2}{\pi}}, \frac{2r_C^*}{r}\sqrt{\frac{2}{3\pi}}\right\}\frac{1}{\sqrt{d}} \,,$$

for any $d \in \mathbb{N}$ and any subset of points C, where D_C is the maximum pairwise distance in C, and r_C^* is the radius of the circumsphere of the convex hull of C.

Proof. We denote by P_a the probability that, at any recursive step, URRH accepts a randomly selected hyperplane. Since P_a is clearly invariant to distance scaling, w.l.o.g. we assume that the maximum distance D_C between any two points of C is equal to 2, and we denote by $x, x' \in C$ two points such that $\|x - x'\| = 2$, with middle point $m := \frac{x+x'}{2}$. We then denote by $\bar{r}$ the resulting *scaled* radius of the sphere $\mathcal{S}(C)$ used at any given recursive step. We recall that the center of $\mathcal{S}(C)$ is denoted by c and that the hyperplane $H_{p.b}(\mathcal{S}(C)) = \{x \in \mathbb{R}^d \,:\, x \cdot p = b\}$ is chosen by first generating p uniformly at random from the unit $(d-1)$-sphere, and thereafter b uniformly at random from the interval $[c \cdot p - \bar{r}, c \cdot p + \bar{r}]$.

Let $\mathcal{S}^*(C)$ be the circumsphere of C, and c^* be the center of $\mathcal{S}^*(C)$. Consider for the moment the special case where $\mathcal{S}(C) = \mathcal{S}^*(C)$ and $\bar{r} = 1$. In this case, we then have $\bar{r} = r_C^*$ and $m = c = c^*$.

If the $\mathrm{Conv}(C)$ contains some point that does not belong to the segment connecting x to x', then P_a can only increase. Hence, in order to find a lower bound on P_a when $\mathcal{S}(C) = \mathcal{S}^*(C)$ and $\bar{r} = 1$, we can simply focus on the input case $C = \{x, x'\}$. In this case, P_a is equal to the probability that $H_{p,b}(\mathcal{S}(C))$ separates x from x'.

We recall that $|b|$ is the distance between $H_{p,b}(\mathcal{S}(C))$ and $c = c^*$. For any given b, we denote by $\mathcal{C}(|b|)$ the hyperspherical cap obtained by cutting $\mathcal{S}(C) = \mathcal{S}^*(C)$ when $H_{p,b}(\mathcal{S}(C))$ is *fixed beforehand*. It is now crucially important to note that, for each given b, the probability P_a corresponds to the probability that (see Figure 4, top, for reference):

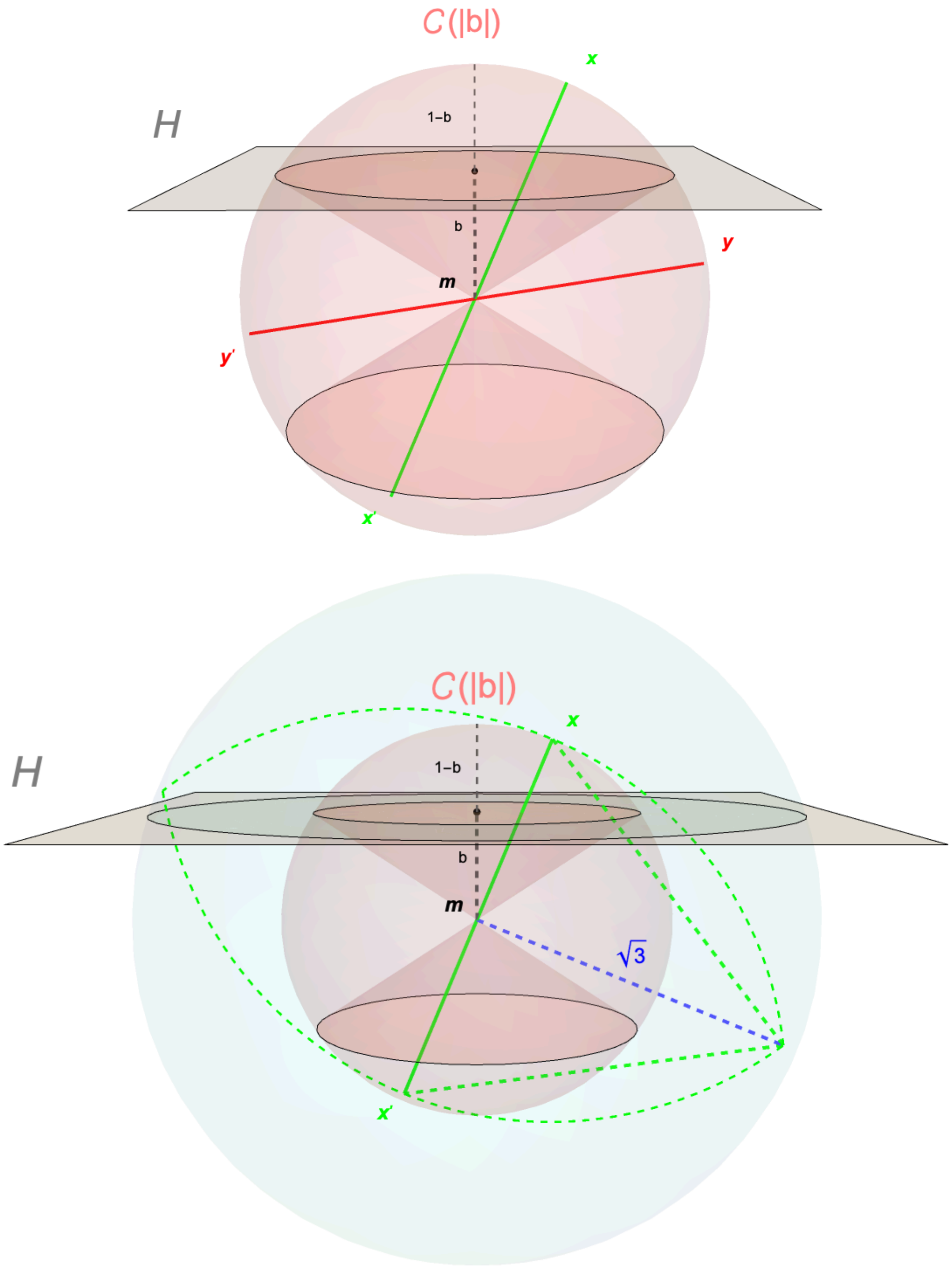

Figure 4: Lower bounding the probability that URRH accepts hyperplane $H_{p,b}(\mathcal{S}(C))$. **Top:** Unit Sphere $\mathcal{S}(C) = \mathcal{S}^*(C)$ for $d = 3$ and set of points $C = \{x, x'\}$ with *scaled* maximum distance in C equal to $D_C := \|x - x'\| = 2$ and $\bar{r} = r^*_C = 1$. The middle point m of the segment connecting x to x' is therefore $\frac{1}{2}(x + x')$. Our goal is to calculate the probability that $H_{p,b}(\mathcal{S}(C))$ separates x from x'. To this effect, we view instead plane $H_{p,b}(\mathcal{S}(C))$ as *fixed beforehand* and calculate the probability that x or x' belong to spherical cap $\mathcal{C}(|b|)$, once x is selected uniformly at random on the sphere, and then integrating the result over b. Note that the position of x uniquely determines the position of x'. In this picture, unlike x and x', the two points y and y' are not cut by $H_{p,b}(\mathcal{S}(C))$ because none of them belongs to $\mathcal{C}(|b|)$. **Bottom:** If the set of point $C \supsetneq \{\boldsymbol{x}, \boldsymbol{x}'\}$ contains more than two points, any point of C cannot be far from $m = \frac{1}{2}(\boldsymbol{x} + \boldsymbol{x}')$ more than $\sqrt{3}\|m - x\| = \sqrt{3}\|m - x'\|$. This implies that we can scale the lower bound of the probability that $H_{p,b}(\mathcal{S}(C))$ is accepted by dividing it by $\sqrt{3}$.

(i) x is selected uniformly at random from $\mathcal{S}(C)$, and

(ii) x or x' belong to $\mathcal{C}(|b|)$.

For any given hyperplane H, this probability is equal to the sum of twice the area of $\mathcal{C}(|b|)$, divided by the area of $\mathcal{S}^*(C)$. Hence, because b is chosen at uniformly at random from $[c \cdot p - \bar{r}, c \cdot p + \bar{r}]$, in this case

$$P_a \geq \int_{-R_C^*}^{R_C^*} \frac{2\text{Area}(\mathcal{C}(|b|))}{\text{Area}(\mathcal{S}^*)} db = 4 \int_0^{R_C^*} \frac{\text{Area}(\mathcal{C}(|b|))}{\text{Area}(\mathcal{S}^*)} db \, .$$

It is well known (e.g., (Li, 2011)) that, the area of a hyperspherical cap $\mathcal{C}$ with height h of a unit $(d-1)$-sphere, can be expressed as

$$\text{Area}(\mathcal{C}) = \frac{1}{2}\text{Area}(\mathcal{S}) I_{2h-h^2}\left(\frac{d-1}{2}, \frac{1}{2}\right) \, ,$$

where $I_{2h-h^2}\left(\frac{d-1}{2}, \frac{1}{2}\right)$ is the regularized incomplete Beta function.

Observing that in this case for any given $b \geq 0$ we have $h = 1 - b$, we can write

$$\begin{aligned}
P_a &\geq 4 \int_0^R \frac{\text{Area}(\mathcal{C}(|1-h|))}{\text{Area}(\mathcal{S}^*)} dh \\
&= 2 \int_0^1 I_{2h-h^2}\left(\frac{d-1}{2}, \frac{1}{2}\right) dh \\
&= 2 \int_0^1 I_{2b-b^2}\left(\frac{d-1}{2}, \frac{1}{2}\right) db \\
&= \frac{2}{B\left(\frac{d-1}{2}, \frac{1}{2}\right)} \int_0^1 \int_0^{2t-t^2} s^{\frac{d-3}{2}} (1-s)^{-\frac{1}{2}} ds dt \qquad (14) \\
&= \frac{2}{B\left(\frac{d-1}{2}, \frac{1}{2}\right)} \int_0^1 \int_{1-\sqrt{1-s}}^1 s^{\frac{d-3}{2}} (1-s)^{-\frac{1}{2}} dt ds \\
&= \frac{2}{B\left(\frac{d-1}{2}, \frac{1}{2}\right)} \int_0^1 s^{\frac{d-3}{2}} ds \\
&= \frac{4}{(d-1)B\left(\frac{d-1}{2}, \frac{1}{2}\right)} \\
&\geq 2\sqrt{\frac{2}{\pi d}} \, ,
\end{aligned}$$

where in (14) we used the Beta function integral form.

In order to generalize this argument and remove the initial assumptions $\mathcal{S}(C) = \mathcal{S}^*(C)$ and $\bar{r} = \frac{D_C}{2} = 1$, it is sufficient to observe that for any $\mathcal{S}(C)$ such that $\bar{r} > \frac{D_C}{2}$, we need to scale the final bound by dividing it by $\bar{r}$, i.e., we have

$$P_a \geq \frac{2}{\bar{r}}\sqrt{\frac{2}{\pi d}} \, .$$

In fact, for any given direction vector p selected by URRH and two $(d-1)$-sphere $\mathcal{S}'(C)$ and $\mathcal{S}''(C)$ both containing the pair of points $\{x, x'\}$ and having radius $\bar{r}'$ and $\bar{r}''$, respectively, with $\bar{r}'' \geq \bar{r}'$, we have

$$\frac{|x \cdot p - x' \cdot p|/\bar{r}''}{|x \cdot p - x' \cdot p|/\bar{r}'} = \frac{\bar{r}'}{\bar{r}''} \, .$$

Note that this argument is independent of the relative position of x and x' inside the spheres. In particular, it holds even when m does not coincide with the center of $\mathcal{S}'(C)$ or that of $\mathcal{S}''(C)$.

The above bound for P_a can be expressed in terms of D_C by replacing $\bar{r}$ with $\frac{2r}{D_C}$:

$$P_a \geq \frac{D_C}{r}\sqrt{\frac{2}{\pi d}}\,.$$

Finally, in order to express this lower bound in terms of the radius of the circumsphere of $\mathrm{Conv}(C)$, we observe that given any set of points C, we have $r^*_C \leq \sqrt{3}\frac{\|x-x'\|}{2} = \sqrt{3}\frac{D_C}{2}$. This is due to the fact that the distance between any point of C and m cannot exceed $\sqrt{3}\frac{D_C}{2}$ (see Figure 4, bottom, for reference). Hence, we also have

$$P_a \geq \frac{2r^*_C}{r}\sqrt{\frac{2}{3\pi d}}\,.$$

Putting together concludes the proof. □

C.1. On the computational complexity of URRH

We now briefly and informally discuss the most significant aspects of the computational complexity of URRH. The time complexity bottleneck lies in testing whether a random hyperplane is accepted, which yields a dominant expected time of $\mathcal{O}(t\,h\,d^{3/2})$ for inserting the $(t+1)$-th point, where h is the expected height of T. The worst-case space complexity is instead $\mathcal{O}(t\,d)$. However, it is worth to point out that Lemma C.5 gives a worst-case result which applies only to pathological cases. In practice, for real-world input datasets, whenever $n \gg d$ the probability of accepting a randomly drawn hyperplane is not as low as evinced by Lemma C.5. This suggests a number of ways for modifying URRH to design heuristics approximating its behavior, and enjoying a remarkably better expected time complexity.

For instance, the expected time for inserting a new point (excluding the hyperplane rejection test), can be reduced in several ways. For each subset of leaves in the subtree rooted at a given (internal) node in T, one can maintain information of a sphere whose radius is at most *twice* the one of the circumsphere through a strategy that takes $\mathcal{O}(h\,d)$ expected time per insertion. Interestingly enough, for the insert operation that splits X while descending the tree, an ad hoc data-structure exists that makes the expected time $\mathcal{O}\left(t(d+\log t)\right)$ per insertion, at the cost of increasing the space complexity by $\mathcal{O}\left(t\log t\right)$, hence requiring $\mathcal{O}\left(t(d+\log t)\right)$ overall space.

Following the above implementation, the expected time of URRH can be viewed as the sum of two terms: **(i)** $\mathcal{O}(h\,d + t(d+\log t)) = \mathcal{O}(t(d+\log t))$ and **(ii)** the total time for testing if a hyperplane is rejected, which still remains $\mathcal{O}(t\,h\,d^{3/2})$. Yet, it is worth stressing that on inputs where T happens to be balanced, the worst-case expected time is dramatically reduced even taking into account the bottleneck of the hyperplane rejection test we mentioned above. For example, if T is a complete binary tree, the expected time complexity per insertion becomes $\mathcal{O}\left(t(d^{3/2}+\log t)\right)$.

D. One-dimensional streaming data

In this section, we present a series of results that apply to data on the real line. The first result directly applies to RCT, for which we have $\mathbb{E}[\mathrm{Rev}_S(\mathrm{RCT}(X))] \geq 0.83028\,\mathrm{Opt}_{\mathrm{Rev}_S}$ (Theorem D.1). The second result concerns an algorithm conceived for the batch setting, that we name I-BISEC, which has a $\frac{3}{4}$ approximation ratio for $n \to \infty$. Although it is smaller than the approximation ratio of RCT for $d = 1$, this result applies to an algorithm, I-BISEC, which has the advantage to be deterministic (Theorem D.2). The third result is the $\frac{1}{2}$ approximation ratio of a very simple algorithm that we name 1D–BESTCATERPILLAR, which chooses the caterpillar having the highest approximation factor between the two that can be built following the order of one-dimensional point coordinates, in the two opposite directions (from left to right or from right to left, representing the points on a horizontal straight line). This is interesting especially from a theoretical viewpoint because of the simplicity of the hierarchical tree structure of the caterpillar tree, which indeed can viewed as one of the basic ways to build a hierarchical tree. It is also a deterministic algorithm (Corollary D.3.1). We also show that there is a randomized version achieving an $\frac{1}{2}$ approximation ratio in expectation, which is worth to mention because of its extreme simplicity (Theorem D.3): this algorithm just sorts the points according to their coordinates and chooses with probability $\frac{1}{2}$ one of the two possoble directions (from left to right or from right to left). Finally, we prove that 1D–BESTCATERPILLAR has also a $\frac{3}{4}$ approximation ratio for the CKMM Revenue (Theorem D.4). Although we do not discuss in detail the dynamic implementation of 1D–BESTCATERPILLAR, and we provide the result in a batch setting, this algorithm can naturally operate in a dynamic setting with no special modifications, with the above approximation ratio that holds *at any round*, i.e., after having received *any* number of points in input in sequential mode.

Theorem D.1. *Given any input set $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}$, we have, for $n \to \infty$,*

$$\mathbb{E}[\mathrm{Rev}_S(\textsc{rct}(X))] \geq 0.83028\, \mathrm{Opt}_{\mathrm{Rev}_S}\ ,$$

where the expectation is over the internal randomization of RCT.

Proof. The proof of this theorem follows the same line as the one of Theorem 5.3. The main idea is partitioning the set of triplets $\{x_i, x_j, x_k\}$ of points in X into *good* and *bad* triplets based on their *length*, i.e., the distance between the leftmost and rightmost point of each triplet. Note that the length of a triplet can be viewed as half the *perimeter* of the (degenerate) triangle having as vertices its three points, while in Theorem 5.3 we used the perimeter of the triangle divided by 3. Note also that when $d = 1$, URRH operating on (degenerate) circumspheres and RCT collapse to the same algorithm. Indeed, the probability of accepting any (degenerate) hyperplane drawn by URRH is 1, and all threshold points are accepted as for RCT. Hence, the approximation ratio of this theorem holds for URRH too.

To apply the same strategy as in Theorem 5.3, the main ingredients are:

- An upper bound on the average pairwise distance over the points in X (or the sum of all pairwise distances).
- A lower bound on the expected revenue for any single triplet of points in X.
- A lower bound on the expected revenue of each good triplet, expressed as a function of their length for both RCT and OPT.

We start by upper bounding the sum of all pairwise distances in X for asymptotically for $n \to \infty$.

Without loss of generality, assume that the smallest interval containing all points of X is $[0, 1]$ and let $\mathcal{T} = \{\{x_i, x_j, x_k\} : x_i, x_j, x_k \in X,\ i < j < k\}$ be the set of triplets of X. For each triplet $t \in \mathcal{T}$, let $l(t) = \max(t) - \min(t)$ denote its length and let $L(X) = \sum_{t \in T} l(t)$. We claim that $L(X) \leq \frac{3}{4}\binom{n}{3}(1 + o(1))$. This bound is tight as witnessed by the example where $\lfloor \frac{n}{2} \rfloor$ points are at 0 and $\lceil \frac{n}{2} \rceil$ points are at 1. To prove the bound, order the points of X by $x_1 \leq \ldots \leq x_n$. For any configuration of x_i's, note that $L(X)$ is increased by making $x_1 = 0$. This is because x_1 belongs to more triplets where it is the left end-point than where it is the right end-point. Similarly, we increase $L(X)$ by making $x_n = 1$. Next, setting $x_2 = 0$ and $x_{n-1} = 1$ increases $L(X)$ for the same reason. Continuing in this way, we find that $L(X)$ is maximized as in the tight example described above (for an odd number of points, $L(X)$ is independent of the position of the middle point $x_{(n+1)/2}$).

We now calculate a lower bound of the expected revenue for any single triplet of points in X as a function of its length.

Fix a triplet $t = \{x_i, x_j, x_k\}$ with $x_i \leq x_j \leq x_k$ so that $l = l(t) = x_k - x_i$. We show that its expected revenue under RCT , $\mathbb{E}[\mathrm{Rev}(\textsc{rct}(t))]$ satisfies $\mathbb{E}[\mathrm{Rev}(\textsc{rct}(t))] \geq (1 - (3 - 2\sqrt{2})l)\mathrm{Rev}(\textsc{opt}(t))$. In particular, setting $l = 1$, we have $\mathbb{E}[\mathrm{Rev}(\textsc{rct}(t))] \geq (2\sqrt{2} - 2)\mathrm{Rev}(\textsc{opt}(t))$. Let $a = x_j - x_i$ and $b = x_k - x_j$ and assume without loss of generality that $a \leq b$. The revenue for this triplet by RCT is given by $\frac{a}{a+b}(1 - b) + \frac{b}{a+b}(1 - a) = \frac{a+b-2ab}{a+b}$, while it is at most $1 - a$ for OPT. Letting $f(a, b) = \frac{a+b-2ab}{(a+b)(1-a)}$, we are led to the following optimization problem:

$$\begin{aligned} \underset{a,b}{\text{minimize}} \quad & f(a, b) \\ \text{subject to} \quad & a + b = l \text{ and } 0 \leq a \leq \min(b, 1 - b). \end{aligned}$$

We compute

$$\begin{aligned} \frac{\partial f}{\partial b} &= \frac{(a + b)(1 - 2a) - (a + b - 2ab)}{((a + b)(1 - a))^2} \\ &= \frac{-2a^2}{((a + b)(1 - a))^2} \leq 0. \end{aligned}$$

Hence for a given a, f is minimized by setting b as large as possible, i.e., $b = l - a$. Now let $g(a) = f(a, l-a) = \frac{l-2a(l-a)}{l(1-a)}$. Setting the derivative of g to 0 gives

$$\begin{aligned}(1-a^*)(4a^*-2l) &= (l-2a^*l+2a^{*2})(-1)\\ 2a^{*2}-4a^*+l &= 0\\ a^* &= 1-\sqrt{1-l/2}.\end{aligned}$$

Letting $h(l) = g(1-\sqrt{1-l/2})$, we thus have $\mathbb{E}[\mathrm{Rev}(\textsc{rct}(t))] \geq h(l)\mathrm{Rev}(\textsc{opt}(t))$. We now show that $h(l)$ is convex in l. This will give $h(l) \geq (1-l)h(0) + lh(1)$. Since $h(0) = 1$, and $h(1) = 2\sqrt{2}-2$, we will have $h(l) \geq 1-(3-2\sqrt{2})l$. Showing $h(l)$ is convex is tedious but straightforward. We compute the second derivative $h''(l)$ and show that $h''(l) \leq 0$. Letting $s = s(l) = \sqrt{1-l/2}$, we simplify $h(l)$:

$$\begin{aligned}h(l) &= g(1-s)\\ &= \frac{l-2(1-s)(l-1+s)}{ls}\\ &= \frac{l-2(l-1+s-(l-1)s-1+l/2)}{ls} \quad \text{(using } s^2 = 1-l/2\text{)}\\ &= \frac{2(2-l)(1-s)}{ls}\\ &= \frac{4s^2(1-s)}{(2-2s^2)s} \quad \text{(using } l = 2-2s^2\text{)}\\ &= 2(1-1/(1+s)).\end{aligned}$$

Using $\frac{ds}{dl} = -\frac{1}{4s}$, we have $h'(l) = \frac{2}{(1+s)^2}\left(-\frac{1}{4s}\right) = -\frac{1}{2s(1+s)^2}$. Finally, $h''(l) = -\frac{1}{2}\left(-\frac{2}{s(1+s)^3} - \frac{1}{s^2(1+s)^2}\right)\left(-\frac{1}{4s}\right)$ which is clearly non-positive.

We now fix $p \in (\frac{3}{4}, 1)$ (we will choose p later to optimize the approximation ratio) and split T into two sets: The bad triplets $B_p = \{t \in T : l(t) > p\}$ and good triplets $G_p = \{t \in T : l(t) \leq p\}$. Define the following quantities:

(i) $RB = \sum_{t \in B_p} \mathbb{E}[\mathrm{Rev}(\textsc{rct}(t))]$,

(ii) $RG = \sum_{t \in G_p} \mathbb{E}[\mathrm{Rev}(\textsc{rct}(t))]$,

(iii) $OB = \sum_{t \in B_p} \mathrm{Rev}(\textsc{opt}(t))$, and

(vi) $OG = \sum_{t \in G_p} \mathrm{Rev}(\textsc{opt}(t))$.

We wish to minimize $\alpha = \frac{RB+RG}{OB+OG}$. We will use the following bounds:

(i) $RB \geq (2\sqrt{2}-2)OB$,

(ii) $RG \geq \left(1-(3-2\sqrt{2})p\right)OG$,

(iii) $OB \leq \frac{3/4}{p}\binom{n}{3}$, and

(vi) $OG \geq \left(1-\frac{p}{2}\right)\left(1-\frac{3/4}{p}\right)\binom{n}{3}$.

(i) and (ii) follow from the lower bound on $\mathbb{E}[\mathrm{Rev}(\textsc{rct}(t))]$ established above. (iii) and (iv) follow from Markov's inequality applied to the sum of lengths upper-bound coupled with the observations that for a triplet t, $\mathrm{Rev}(\textsc{opt}(t)) \in [1-l(t)/2, 1]$.

From (i) and (ii), we have $\alpha \geq \frac{(2\sqrt{2}-2)OB+\left(1-(3-2\sqrt{2})p\right)OG}{OB+OG}$. For fixed OB, this is increasing in OG and hence is minimized for the smallest possible OG. Combining (iii) and (iv), we obtain $OG \geq \left(1-\frac{p}{2}\right)\left(\frac{4}{3}p-1\right)OB$. Plugging this into the above expression we get

$$\alpha \geq \frac{2\sqrt{2}-2+\left(1-(3-2\sqrt{2})p\right)\left(1-\frac{p}{2}\right)\left(\frac{4}{3}p-1\right)}{1+\left(1-\frac{p}{2}\right)\left(\frac{4}{3}p-1\right)}\ .$$

Maximizing this over $p \in (3/4, 1)$ yields a maximum of $\alpha \geq 0.83028$, which can be obtained by setting $p = 0.8633$. □

The I-BISEC algorithm

We now describe a very simple yet efficient divisive algorithm for one-dimensional input data, that we call I-BISEC (Interval-Bisection), having an MW revenue lower bounded by $\frac{3}{4}\mathrm{Opt}_{\mathrm{Rev}_S}$ for $n \to \infty$. At the first round this algorithm selects a threshold τ equal to the middle point between the leftmost and the rightmost input points (representing horizontally the one-dimensional straight line containing the whole input X). In the subsequent rounds, I-BISEC proceeds recursively with the two subsets of points respectively lying on the two semi-intervals separated by τ, until the number of points processed in a single recursion step is equal to 1. We highlight that the choice of threshold point τ, at each round, only depends on the leftmost and the rightmost input points of each subset of points considered. Finally, in the special case in which n' points are coincident with the selected threshold τ in a recursive step of the algorithm, $\left\lfloor \frac{n'}{2} \right\rfloor$ are placed in the left partition and $\left\lceil \frac{n'}{2} \right\rceil$ in the right partition.

Besides being easy to implement, this algorithm is also computationally very efficient. It is not difficult to verify that using a balanced binary tree where each leaf contains one point of X, we can easily find the points closest to the middle point threshold of each sub-interval considered. Hence, the worst-case time complexity is $\mathcal{O}(n \log n)$, while the space required is always equal to $\mathcal{O}(n)$.

The following theorem quantifies the approximation ratio of I-BISEC.

Theorem D.2. *Given any input set $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}$ with $n \geq 3$, we have*

$$\mathbb{E}[\mathrm{Rev}_S(\text{I-BISEC}(X))] \geq \frac{3n-8}{4n-8}\,\mathrm{Opt}_{\mathrm{Rev}_S}\ .$$

Proof. Without loss of generality, assume that the smallest interval containing all points of X is $[0, 1]$ and that the x_i's are in non-decreasing order. Let n' and n'' be the number of points respectively in $\left[0, \frac{1}{2}\right]$ and $\left(\frac{1}{2}, 1\right]$.

Thus we have

$$0 = x_1 \leq x_2 \leq \ldots \leq x_{n'} \leq \tfrac{1}{2} \leq x_{n'+1} \leq x_{n'+2} \cdots \leq x_{n'+n''} \equiv x_n = 1\ .$$

Without loss of generality, assume now that $n' \geq n''$, which implies $n' \geq 2$. The sum of all pairwise distances between any two points in $[0, \frac{1}{2}]$ is

$$\begin{aligned}
\sum_{1\leq i<j\leq n'} (x_j - x_i) &= (0-(n'-1))x_1 + (1-(n'-2))x_2 - \ldots + ((n'-1)-0)x_{n'} \\
&= -(n'-1)x_1 - (n'-3)x_2 - \ldots - (1-n')x_{n'} \\
&= -\sum_{k=1}^{n'} (n'-(2k-1))x_k \\
&= -\sum_{k=1}^{\lceil (n'-1)/2 \rceil} (n'-(2k-1))x_k \ - \sum_{k'=\lceil (n'+1)/2 \rceil}^{n'} (n'-(2k'-1))x_{k'} \\
&= -\sum_{k=1}^{\lceil (n'-1)/2 \rceil} (n'-(2k-1))x_k \ + \sum_{\ell=1}^{\lfloor (n'+1)/2 \rfloor} (2\ell-1-(n' \bmod 2))x_{\ell+\lceil (n'-1)/2 \rceil} \\
&\leq -\sum_{k=1}^{\lceil (n'-1)/2 \rceil} (n'-(2k-1))\cdot 0 \ + \sum_{\ell=1}^{\lfloor (n'+1)/2 \rfloor} (2\ell-1-(n' \bmod 2))\cdot 1/2\ .
\end{aligned}$$

The above inequality shows that maximum of this sum is attained when the leftmost half of points in the interval $\left[0, \frac{1}{2}\right]$ is placed at 0 and the rightmost half is placed at[10] $\frac{1}{2}$. This immediately implies that the average pairwise distance of the points in $[0, \frac{1}{2}]$ is upper bounded by $\frac{\lceil n'/2 \rceil \lfloor n'/2 \rfloor}{\binom{n'}{2}} \frac{1}{2} \le \frac{1}{4}\left(1 + \frac{1}{n'-1}\right)$. Analogously, the average pairwise distance of the points in $(\frac{1}{2}, 1]$ is upper bounded by $\frac{\lceil n''/2 \rceil \lfloor n''/2 \rfloor}{\binom{n''}{2}} \frac{1}{2} \le \frac{1}{4}\left(1 + \frac{1}{n''-1}\right)$ if[11] $n'' \ge 2$. Let now r' and r'' be respectively $1 - \frac{1}{4}\left(1 + \frac{1}{n'-1}\right)$ and $1 - \frac{1}{4}\left(1 + \frac{1}{n''-1}\right)$. We can finally state that the approximation factor of I-BISEC corresponding to the triplets broken *by the first cut solely*, is lower bounded by

$$\frac{\left(\frac{n'(n'-1)}{2} n'' r' + \frac{n''(n''-1)}{2} n' r''\right)}{\left(\frac{n'(n'-1)}{2} n'' + \frac{n''(n''-1)}{2} n'\right)} .$$

It is easy to verify that, when $n' + n'' = n$, the above expression is exactly equal to $\frac{3n-8}{4n-8}$. To conclude the proof, we observe that for all the subsets of triplets broken by subsequent cuts, the corresponding approximation factor cannot be smaller than the one obtained by analyzing the triplets broken by the first cut. Indeed, the sub-intervals split by the other cuts are narrower than $[0, 1]$. Hence, the approximation factor of I-BISEC is lower bounded by $\frac{3n-8}{4n-8}$, as claimed.

□

The 1D–BESTCATERPILLAR algorithm

When $d = 1$, a baseline method consists in selecting the caterpillar tree that is **(i)** compatible with the Euclidean metric of the input points and **(ii)** has the maximum MW revenue. This selection is performed considering *only the two* caterpillar trees embedded in a plane P where the points lies on a straight line L contained in P. More precisely, their vertices are contained in P such that the edges, represented by line segments on P, do not cross. The points are embedded on L preserving all distances $d_{i,j}$ between any two points x_i and x_j. As anticipated above, there are only two such caterpillar trees: representing L horizontally, one of these two caterpillars trees clusters the points from left to right, that we call 1D–LEFTCATERPILLAR. The other caterpillar tree clusters the points from right to left, that we call 1D–RIGHTCATERPILLAR. This algorithm, that we call 1D–BESTCATERPILLAR, outputs the caterpillar tree having the maximum MW revenue.

Note that we can use Theorem D.3 to state that, once we build the two caterpillars (which can be done in a very easy and fast way, i.e., with a worst-case time complexity equal to $\mathcal{O}(n \log n)$), if we select 1D–LEFTCATERPILLAR and 1D–RIGHTCATERPILLAR both with probability $\frac{1}{2}$, then we obtain an *expected* approximation ratio equal to $\frac{1}{2}$.

Moreover, even the deterministic algorithm 1D–BESTCATERPILLAR can be easily implemented with a *total* worst-case time complexity equal to $\mathcal{O}(n \log n)$. We now briefly sketch the implementation for computing the revenue of 1D–LEFTCATERPILLAR. The strategy to calculate the revenue of 1D–RIGHTCATERPILLAR is analogous because of the symmetry between the two caterpillars. We first generate an array $A[]$ where the i-th record stores the i-th point coordinate in an non-decreasing order. This operation requires $\mathcal{O}(n \log n)$ time because we need to sort thee n coordinates. Let d_i be the distance between the first (leftmost) element and the i-th element. We now scan $A[]$ from the first to the last element, to associate with the i-th element both the distance d_i and the sum $\sum_{j<i} d_j$. This operation requires $\mathcal{O}(n)$ time. The key point in this implementation is the following. To compute the revenue of all triplets having x_k as middle point, we simply need to sum $D - d_{j,k} = D - (d_k - d_j)$ over all $j < k$, and multiply the result by $n - k$. To see why, observe that using 1D–LEFTCATERPILLAR, only the distances between x_k and the elements on its left will contribute to the total revenue, and we have $n - k$ elements on the right of k. Hence, the total revenue contribution of all triplets having x_k as middle point is equal to

$$(n-k)\sum_{j<k}(D - d_{j,k}) = (n-k)\sum_{j<k}(D - (d_k - d_j)) = (n-k)\left((k-1)(D - d_k) + \sum_{j<k} d_j\right) ,$$

[10] If n' is odd, then the middle point $x_{(n'+1)/2}$ can be anywhere in $[0, \frac{1}{2}]$. For the sake of simplicity, we set it to be equal to $\frac{1}{2}$ in the above inequality.

[11] If $n'' = 1$ there are no pairs of points in $\left(\frac{1}{2}, 1\right]$.

which can now be easily calculated for *all* $k \in [n]$ by scanning $A[]$ from left to right because we previously stored d_k and $\sum_{j<k} d_j$ for all $k \in [n]$. This operation requires therefore a total time equal to $\mathcal{O}(n)$. Hence, we can compute the revenue of 1D–LeftCaterpillar, and, by symmetry, of 1D–RightCaterpillar, in time $\mathcal{O}(n \log n) + \mathcal{O}(n) = \mathcal{O}(n \log n)$. Finally, to determine the caterpillar 1D–BestCaterpillar, we only need to find the maximum value by comparing the revenues of 1D–LeftCaterpillar and 1D–RightCaterpillar, which can be done in constant time once we have the two revenue values.

Theorem D.3. *Given any input set $X = \{x_1, \dots, x_n\} \subseteq \mathbb{R}$, we have*

$$\mathrm{Rev}_S(\text{1D–}\textsc{LeftCaterpillar}(X)) + \mathrm{Rev}_S(\text{1D–}\textsc{RightCaterpillar}(X)) \geq \mathrm{Opt}_{\mathrm{Rev}_S} \,.$$

Proof. For each $i \in [n]$, we denote by d_i the Euclidean distance between the first and the i-th input point x_i on L, from left to right where L is viewed as a horizontal line. For the sake of simplicity, hereinafter we indicate point x_ℓ for any $\ell \in [n]$ by simply writing its index ℓ.

Note that for any tree T, for any triplet $\{i, j, k\}$, where $i, j, k \in [n]$ such that $i < j < k$, there are only two possible cases:

1. If i and j are clustered before k, then we have an MW gain equal to $D - (d_j - d_i)$.
2. If j and k are clustered before i, then we have an MW gain equal to $D - (d_k - d_j)$.

Let now $\mathcal{T}$ be the set of all triplets $\{i, j, k\}$, where $i, j, k \in [n]$ such that $i < j < k$. Let $\mathcal{T}_\ell \subseteq \mathcal{T}$ and $\mathcal{T}_r \subseteq \mathcal{T}$ be respectively defined as the following set of triplets[12]

$$\begin{aligned}
\mathcal{T}_\ell &= \{\{i, j, k\} : i, j, k \in [n], i < j < k, \;\; D - (d_j - d_i) \geq D - (d_k - d_j)\} \,, \\
\mathcal{T}_r &= \{\{i, j, k\} : i, j, k \in [n], i < j < k, \;\; D - (d_k - d_j) > D - (d_i - d_j)\} \,.
\end{aligned}$$

We have therefore $\mathcal{T} = \mathcal{T}_\ell \cup \mathcal{T}_r$ and $\mathcal{T}_\ell \cap \mathcal{T}_r = \emptyset$.

Thus, $\mathrm{Opt}_{\mathrm{Rev}_S}$ can be bounded as follows:

$$\begin{aligned}
\mathrm{Opt}_{\mathrm{Rev}_S} &\leq \sum_{\substack{\{i,j,k\} \in \mathcal{T}: \\ i<j<k}} \max\{D - (d_j - d_i), D - (d_k - d_j)\} \\
&= \left(\sum_{\substack{\{i,j,k\} \in \mathcal{T}_\ell: \\ i<j<k}} D - (d_j - d_i) \right) + \left(\sum_{\substack{\{i,j,k\} \in \mathcal{T}_r: \\ i<j<k}} D - (d_k - d_j) \right) \,.
\end{aligned}$$

Let now 1D–LeftCaterpillar and 1D–RightCaterpillar be the caterpillar trees following the order of the points on L respectively from left to right and from right to left. Note that, for all triplets $\{i, j, k\}$, where $i, j, k \in [n]$ such that $i < j < k$, we have that i and j are clustered before k using 1D–LeftCaterpillar, while j and k are clustered before i using 1D–RightCaterpillar. Then, we can write the MW revenue of each caterpillar tree as the sum of two terms as we previously did for opt:

$$\mathrm{Rev}_S(\text{1D–}\textsc{LeftCaterpillar}(X)) = \left(\sum_{\substack{(i,j,k) \in \mathcal{T}_\ell: \\ i<j<k}} D - (d_j - d_i) \right) + \left(\sum_{\substack{(i,j,k) \in \mathcal{T}_r: \\ i<j<k}} D - (d_j - d_i) \right) \,,$$

[12] The triplets for which $d_j - d_i = d_k - d_j$ can be included arbitrarily either into $\mathcal{T}_\ell$ (as we did above) or $\mathcal{T}_r$, without affecting the rest of the proof.

and

$$\mathrm{Rev}_S(\text{1D--}\textsc{RightCaterpillar}(X)) = \left(\sum_{\substack{(i,j,k)\in\mathcal{T}_\ell: \\ i<j<k}} D - (d_k - d_j) \right) + \left(\sum_{\substack{(i,j,k)\in\mathcal{T}_r: \\ i<j<k}} D - (d_k - d_j) \right) .$$

It is now sufficient to inspect the two terms above of each MW revenue, i.e., the one of 1D–LeftCaterpillar, 1D–RightCaterpillar and OPT, to finally state

$$\mathrm{Rev}_S(\text{1D--}\textsc{LeftCaterpillar}(X)) + \mathrm{Rev}_S(\text{1D--}\textsc{RightCaterpillar}(X)) \geq \mathrm{Opt}_{\mathrm{Rev}_S} ,$$

as claimed. □

An immediate consequence of Theorem D.3 is the following corollary on the approximation ratio of $\mathrm{Rev}_S(\text{1D--}\textsc{BestCaterpillar})$.

Corollary D.3.1. *Given any input set* $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}$*, we have*

$$\mathrm{Rev}_S(\text{1D--}\textsc{BestCaterpillar}(X)) \geq \frac{1}{2}\,\mathrm{Opt}_{\mathrm{Rev}_S} .$$

Finally we have the following analogous result for the CKMM Revenue.

Theorem D.4. *Given any input set* $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}$*, we have*

$$\mathrm{Rev}_D(\text{1D--}\textsc{BestCaterpillar}(X)) \geq \frac{3}{4}\,\mathrm{Opt}_{\mathrm{Rev}_D} .$$

Proof. We begin by renaming our data points such that they correspond with the 1-d metric. I.e., such that $i < j$ if and only if $d_i < d_j$. By considering the triplet-wise formulation given in Section B.2 it is clear that $\mathrm{Opt}_{\mathrm{Rev}_D} \leq \sum_{i<j<k} \max\{d_{i,j} + d_{i,k}, d_{i,j} + d_{j,k}, d_{i,k} + d_{j,k}\}$. Since $i < j < k$ we are guaranteed that $d_{i,j} + d_{j,k} = d_{i,k}$ and thus $\max\{d_{i,j} + d_{i,k}, d_{i,j} + d_{j,k}, d_{i,k} + d_{j,k}\} = \max\{d_{i,j} + d_{i,k}, d_{i,k} + d_{j,k}\}$. Therefore, $\mathrm{Opt}_{\mathrm{Rev}_D} \leq \sum \max\{d_{i,j} + d_{i,k}, d_{i,k} + d_{j,k}\} \leq \sum 2d_{i,k}$.

On the other hand, for every triplet $i < j < k$, 1D–LeftCaterpillar clusters x_i, x_j before clustering x_k and 1D–RightCaterpillar clusters x_j, x_k before x_i. Therefore, $\mathrm{Rev}_D(\text{1D--}\textsc{BestCaterpillar}(X)) = \max\{\sum_{i<j<k}(d_{i,j} + d_{i,k}), \sum_{i<j<k}(d_{j,k} + d_{i,k})\} \geq \sum \frac{1}{2}(d_{i,j} + d_{i,k}) + \frac{1}{2}(d_{j,k} + d_{i,k}) = \sum \frac{3}{2} d_{i,k}$. Thus overall, $\mathrm{Rev}_D(\text{1D--}\textsc{BestCaterpillar}(X)) \geq \frac{3}{4}\mathrm{Opt}_{Rev_D}$. □

E. Supplementary material for Section 6

In this section we expand on the experimental results from Section 6. Table 4 shows several characteristics of the datasets we used in our experiments: the number of points n, the dimensionality d, the number of ground-truth clusters, and a description of the input.

Dataset	n	d	#classes	Description
MNIST	$6.0 \cdot 10^4$	784	10	Image pixels
ILSVRC12	$1.3 \cdot 10^6$	512	1000	ResNet34 embeddings
ALOI	$1.1 \cdot 10^5$	128	1000	Color Histograms
OneG	$1.0 \cdot 10^4$	2	1	Standard Gaussian
TwoG	$1.0 \cdot 10^4$	2	2	Two Standard Gaussians, 4σ separation in one dimension

Table 4: Summary of datasets used in the experiments in Section 6.

	MNIST	ILSVRC12	ALOI	OneG	TwoG
RCT	0.93±0.01	0.94±0.0	0.91±0.01	0.90±0.01	0.90±0.06
URRH	0.93±0.0	0.94±0.0	0.90±0.01	0.90±0.01	0.90±0.03
BIRCH	0.93	0.94	0.91	0.87	0.98
PERCH	0.92	0.94	0.91	0.87	0.90
GRINCH	0.93	0.93	0.89	0.88	0.97
PROJECTED RANDOM CUT	0.92±0.0	0.94±0.0	0.88±0.01	0.87±0.0	0.86±0.07
RANDOM	0.92	0.93	0.85	0.74	0.71
UPPER BOUND	1.0	1.0	1.0	1.0	1.0

Table 5: MW Revenue approximation factors using RBF kernel similarity; ↑ is better. Each revenue is shown as a percentage of the corresponding upper bound for that dataset.

The other four tables in this section show the values of the four objective measures we consider in this paper, as a percentage of the upper bound (for Revenue measures), resp. lower bound (for Cost measures) for the specific dataset. The MW Revenue table is also shown in the main body of the paper, but we include it here, as well, for ease of comparison with the other results.

As discussed in the body, the trends observed for the MW Revenue objective are also observed in the CKMM Revenue, Dasgupta Cost, and MW Cost objectives. Namely, RCT and URRH excel on OneG where the data are noisy, but are outperformed by other algorithms on TwoG where ground truth cluster separation is apparent. In addition, RCT and URRH perform competitively on all real-world datasets.

	MNIST	ILSVRC12	ALOI	OneG	TwoG
RCT	0.96±0.0	0.97±0.0	0.95±0.0	0.94±0.0	0.93±0.04
URRH	0.96±0.0	0.97±0.0	0.94±0.0	0.94±0.0	0.93±0.03
BIRCH	0.96	0.97	0.95	0.91	0.98
PERCH	0.96	0.97	0.94	0.91	0.93
GRINCH	0.96	0.97	0.94	0.92	0.98
PROJECTED RANDOM CUT	0.96±0.0	0.97±0.0	0.93±0.01	0.92±0.0	0.91±0.04
RANDOM	0.95	0.97	0.91	0.83	0.80
UPPER BOUND	1.0	1.0	1.0	1.0	1.0

Table 6: CKMM Revenue approximation factors using ℓ_2-distance; ↑ is better. Each revenue is shown as a percentage of the corresponding upper bound for that dataset.

	MNIST	ILSVRC12	ALOI	OneG	TwoG
RCT	1.04±0.0	1.03±0.0	1.06±0.01	1.08±0.01	1.10±0.05
URRH	1.04±0.0	1.03±0.0	1.07±0.01	1.08±0.0	1.09±0.04
BIRCH	1.04	1.03	1.06	1.11	1.02
PERCH	1.04	1.03	1.06	1.11	1.09
GRINCH	1.04	1.04	1.07	1.10	1.03
PROJECTED RANDOM CUT	1.05±0.0	1.03±0.0	1.07±0.01	1.1±0.0	1.13±0.06
RANDOM	1.05	1.04	1.10	1.21	1.26
LOWER BOUND	1.0	1.0	1.0	1.0	1.0

Table 7: Dasgupta Cost approximation factors using RBF kernel similarity; ↓ is better. Each cost is computed relative to the lower bound for that dataset.

	MNIST	ILSVRC12	ALOI	OneG	TwoG
RCT	1.08±0.01	1.07±0.0	1.14±0.01	1.27±0.02	1.34±0.19
URRH	1.09±0.0	1.07±0.0	1.16±0.01	1.26±0.02	1.34±0.15
BIRCH	1.08	1.06	1.14	1.36	1.09
PERCH	1.09	1.07	1.15	1.36	1.34
GRINCH	1.09	1.08	1.17	1.32	1.10
PROJECTED RANDOM CUT	1.1±0.01	1.07±0.0	1.18±0.02	1.34±0.01	1.47±0.22
RANDOM	1.10	1.08	1.24	1.71	2.00
LOWER BOUND	1.0	1.0	1.0	1.0	1.0

Table 8: MW Cost approximation factors using ℓ_2-distance; ↓ is better. Each cost is computed relative to the lower bound for that dataset.