

An Objective for Hierarchical Clustering in Euclidean Space and Its Connection to Bisecting K-means*

Yuyan Wang,¹ Benjamin Moseley¹

¹Tepper School of Business, Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, Pennsylvania 15213
{yuyanw, moseleyb}@andrew.cmu.edu

Abstract

This paper explores hierarchical clustering in the case where pairs of points have dissimilarity scores (e.g. distances) as a part of the input. The recently introduced objective for points with dissimilarity scores results in *every tree* being a $\frac{1}{2}$ approximation if the distances form a metric. This shows the objective does not make a significant distinction between a good and poor hierarchical clustering in metric spaces.

Motivated by this, the paper develops a new global objective for hierarchical clustering in Euclidean space. The objective captures the criterion that has motivated the use of divisive clustering algorithms: that when a split happens, points in the same cluster should be more similar than points in different clusters. Moreover, this objective gives reasonable results on ground-truth inputs for hierarchical clustering.

The paper builds a theoretical connection between this objective and the bisecting k -means algorithm. This paper proves that the optimal 2-means solution results in a constant approximation for the objective. This is the first paper to show the bisecting k -means algorithm optimizes a natural global objective over the entire tree.

1 Introduction

In hierarchical clustering, the input is a set of points, with a score that represents the pairwise similarity or dissimilarity of the points. The goal is to output a tree, often binary, whose leaves represent data points, and internal nodes represent clusters. Each internal node is a cluster of the leaves in the subtree rooted at it. When a node gets closer towards the leaves, the cluster it represents should become more refined, and the points in this cluster should become more similar. The nodes of the same level in this tree represent a partition of the given data set into clusters. Note that each data point (leaf) belongs to many clusters, one for each ancestor.

Figure 1 shows a sample hierarchical clustering tree for biological species. All the nodes in the tree on the right are

painted in the same colors with the clusters they represent in the picture on the left.

The mainstream algorithms used to do hierarchical clustering can be classified roughly into two categories: agglomerative and divisive. **Agglomerative algorithms** initialize every point to be in their own individual cluster. They iteratively pick the two clusters that are the most similar to each other to merge into a bigger cluster. Meanwhile, the algorithms create a parent in the hierarchical tree produced that is connected to the two nodes corresponding to the two clusters before merging. The merging terminates when only one cluster remains.

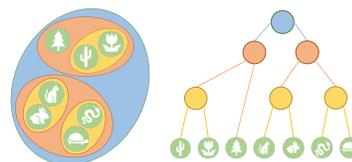


Figure 1: A hierarchical clustering tree. The green leaves are the input data points. Internal nodes represent a cluster of the leaves in their subtree.

It is necessary for an agglomerative algorithm to quantify the similarity between clusters, which can be defined in several ways. For example, average-linkage is a popular agglomerative algorithm, which measures the similarity of two clusters by calculating the average pairwise inter-cluster similarity score. Steinbach et al. (2000), Murtagh (1983), Murtagh and Contreras (2012) and Zhao, Karypis, and Fayyad (2005) discussed about common agglomerative algorithms and compared the performance of different agglomerative algorithms in a variety of backgrounds. Ackerman and Ben-David (2016) identified properties of trees produced by linkage-based agglomerative algorithms.

Divisive algorithms initialize the whole point set as one single cluster, and create a root node corresponding to this cluster in the hierarchical tree. They iteratively **split** a cluster into smaller clusters. Then the algorithms create nodes representing the separated clusters in the tree and make them the children of the parent node. During the split, we want the points in different clusters to be less similar than the points

*Y. Wang and B. Moseley were supported in part by a NSF Grants CCF-1830711, CCF-1733873, CCF-1733873 and CCF-1845146, a Google Research Award, a Bosch junior faculty chair and an Infor faculty award.
Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

in the same cluster. Again, the notion of similarity is open to different interpretations. A divisive algorithm terminates when every point is in its own individual cluster. The divisive algorithms that split a set into two subsets at each iteration are called bisecting algorithms.

Data scientists regularly use the bisecting k -mean algorithms at each level¹. This is used when the distances between the data points are used as dissimilarity scores. See Steinbach et al. (2000) and Murtagh and Contreras (2012) for more information on divisive algorithms. Ackerman and Ben-David (2016) proved that popular divisive algorithms can produce clusterings different from linkage-based agglomerative algorithms. Naturally, they may be optimizing fundamentally different criteria.

Objective Functions: There has been a recent interest in identifying a global objective for hierarchical clustering. Dasgupta (2016) developed a cost function objective for data sets with similarity scores between points. For a given data set V with n points $\{1, 2, 3, \dots, n\}$, let w_{ij} be the similarity score between points i and j . In a tree T , let $T[i \vee j]$ denote the subtree rooted at the least common ancestor of i and j , and $|\text{leaves}(T[i \vee j])|$ denote the number of leaves of $T[i \vee j]$. The cost objective function introduced in Dasgupta (2016) is defined as: $\min_T \text{cost}_T(V) = \sum_{1 \leq i < j \leq n} w_{ij} |\text{leaves}(T[i \vee j])|$.

Since every w_{ij} is multiplied with the number of leaves of the smallest tree containing both i and j , the points that are more similar (bigger w_{ij} 's) are encouraged to have $T[i \vee j]$ with fewer leaves. In other words, the objective function is encouraging points which are more similar to each other to be split at lower levels of the tree where there are fewer leaves at the least common ancestor.

The work of Dasgupta (2016) has initiated an exciting line of study (Roy and Pokutta 2017; Cohen-addad et al. 2019; Charikar et al. 2019; Chatziafratis, Niazadeh, and Charikar 2018; Ghoshdastidar, Perrot, and von Luxburg 2018). Cohen-addad et al. (2019) generalized the results in Dasgupta (2016) into a class of cost functions that possess properties desirable of a valid objective function. They showed that the average-linkage algorithm is a $\frac{2}{3}$ -approximation for an objective based on the Dasgupta (2016) objective.² This objective modifies the Dasgupta objective to handle dissimilarity scores. Let $d(i, j)$ be the distance between i and j . The objective is $\max_T \sum_{1 \leq i < j \leq n} d(i, j) |\text{leaves}(T[i \vee j])|$. The motivation is similar to the Dasgupta objective, except now the similarity score w_{ij} is swapped to a dissimilarity score $d(i, j)$ and the problem is changed to a maximization problem. Contemporaneously, Moseley and Wang (2017) designed a revenue objective function based on Dasgupta (2016) and showed the average-linkage algorithm is a constant approximation for the objective. Charikar et al. (2019) showed an improved analysis of average-linkage for Euclidean data. Together Cohen-addad et al. (2019), Charikar et al. (2019) and Moseley and Wang (2017) have established a relationship between a practically

popular algorithm and global objectives. This gives a foundational understanding of the average-linkage algorithm.

Euclidean Data: This paper is interested in data embedded in Euclidean space where the ℓ_2 distance between points represents their dissimilarity. There is currently one global objective that has been proposed for data with dissimilarity scores. This is the objective of Cohen-addad et al. (2019) described above, an extension of the Dasgupta objective (Dasgupta 2016). Throughout this paper, we refer to the objective in Cohen-addad et al. (2019) as CKMM objective. This paper shows in Section 7 that *every* tree is a $\frac{1}{2}$ -approximation for the CKMM objective if the data is in a metric space. Previously, it was known that all trees gave a constant approximation (Cohen-addad et al. 2019).

In a common case where data is in Euclidean space, one type of metric, the objective does not make a large differentiation between different clusterings. In practice, it is clear that some trees are more desirable than others. It is an interesting question to find an objective that makes a stronger distinction between different clusterings. This is the target question this paper addresses.

Divisive Algorithms: While great strides have been made on the foundations of hierarchical clustering, it remains an open question to explain what popular divisive algorithms optimize. In particular, the popular bisecting k -means algorithm has been proven to be at least a factor $O(\sqrt{n})$ far from optimal for the objectives given in Moseley and Wang (2017) and Dasgupta (2016). This can be viewed as these algorithms being *extremely bad* for these objectives in the worst case. This contrasts with the performance of average-linkage for known objectives. Perhaps, this highlights that bisecting k -means and other divisive algorithms optimize something fundamentally different than average-linkage and general linkage based algorithms. It remains to discover a global objective that helps characterize the optimization criteria of divisive algorithms, another target of this paper.

Results: This paper introduces a new revenue maximization objective for hierarchical clustering on a point set in Euclidean space. The objective is designed to capture the main criterion that motivates the use of divisive algorithms: when data is split at a level of the tree, the data in each sub-cluster should be closer to each other than data points in different clusters.

Each node in the tree corresponds to a *split* that generates revenue. The objective specifies that the global revenue of the tree is the summation of the revenue at each node. The split revenue captures the quality of the split.

- **Guiding Principle:** The new objective function enforces that a split is good if the *inter-cluster distances are big compared to intra-cluster distances*³, as is indicated in Figure 1. This is the main motivation behind a generic divisive algorithm. Of course, the global tree structure influences the possible revenue at an individual split.

¹The word “bisecting” refers to the case when $k = 2$.

²Throughout this paper we use $c > 1$ for approximations on minimization problems and $c < 1$ for maximization.

³Here “inter-cluster distances” refers to that between points in different clusters, while “intra-cluster distances” refers to that between points in the same cluster.

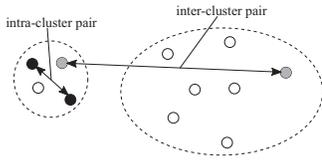


Figure 2: Intra- and inter-cluster distance of two clusters. The black pair is an example of intra-cluster pairs, and the grey pair is an example of inter-cluster pairs.

We show several interesting properties of this new objective.

- For problem instances corresponding a ground-truth as introduced in Cohen-addad et al. (2019), this objective gives desirable optimal solutions. In particular, Cohen-addad et al. (2019) introduced a large class of instances that have a natural corresponding hierarchical clustering that should be optimal. We prove that these trees are *optimal* for the new objective function we propose on such instances. We note that these instances generalize instances given in Dasgupta (2016) that were used to motivate a hierarchical clustering objective.
- The bisecting k -means algorithm is a constant approximation for the objective. This establishes that the objective is closely related to the bisecting k -means algorithm and aids in understanding the underlying structure of solutions the algorithm produces. *This is the first global objective that this algorithm is known to provably optimize.*
- The objective is trivially modular over the splits, like the objectives of Cohen-addad et al. (2019), Moseley and Wang (2017) and Dasgupta (2016).
- In the context of metric spaces, this objective has different properties compared to some proposed objectives. It is known that the Random algorithm⁴, which partitions data uniformly at random at each node, is a constant approximation for the CKMM objective with dissimilarity scores that need not form a metric. Further, it is a constant approximation for the Moseley and Wang (2017) objective with similarity scores. For these two objectives, Random is a $\frac{2}{3}$ and $\frac{1}{3}$ approximation, respectively. The Random algorithm can produce undesirable hierarchical clusterings and it is counterintuitive that it is a constant approximation for these objectives. This paper shows that Random results in an $O(\frac{1}{n^\epsilon})$ -approximation for the proposed objective for a constant $\epsilon > 0$. Therefore, the Random algorithm provably performs poorly for the new objective. This can be seen as a strength of the new objective over those proposed.

We further show the following about other objectives in metric space. These show that some other objectives do not make a large differentiation between trees in metric space, even if the trees correspond to a poor clustering. Our objective does and this can be seen as an advantage of the new objective.

- As mentioned, we show that every tree is a $\frac{1}{2}$ -approximation for CKMM objective when points have dissimilarity scores that form a metric.

⁴See Section 6 for a formal description of the algorithm.

- We show that every tree is a 2-approximation for the Dasgupta objective (Dasgupta 2016) for similarity scores that satisfy the triangle inequality. We include this result to provide insight into this objective. However we note that this is less surprising than the similar result on the CKMM objective since some natural similarity score instances do not satisfy the triangle inequality.

We investigate empirically the performance of three popular algorithms used in practice and Random algorithm for the new objective. As is suggested by theory, the proposed objective moderately favors bisecting k -means over two agglomerative algorithms, while magnifying the gap between the performance of Random and the other three algorithms.

Other Related Work: Other work centers around when bisecting algorithms work well. The work of Dasgupta and Long; Plaxton (2005; 2006) show the remarkable result that hierarchical trees exists such that each level of the tree optimizes the corresponding k -clustering objective. These algorithms are complex and are mostly of theoretical interest. Balcan, Blum, and Vempala (2008) showed that partitioned clusterings can be uncovered from using hierarchical clustering methods under stability conditions. The work of Awasthi, Blum, and Sheffet; Balcan, Blum, and Gupta; Carlsson and Mémoli (2012; 2013; 2010) and pointers therein study stability conditions of clustering.

The work of Charikar and Chatziafratis (2017) and Roy and Pokutta (2017) were the first to give improved bounds on the objectives of Dasgupta. Currently, the best known approximations for both the objective of Dasgupta (2016) and CKMM (Cohen-addad et al. 2019) were shown in Charikar, Chatziafratis, and Niazadeh (2019). This work gave a $\frac{1}{3} + \epsilon$ and $\frac{2}{3} + \delta$ approximations for some small constants ϵ and δ , respectfully. Further, they shown that average-linkage is no better than a $\frac{1}{3}$ and $\frac{2}{3}$ approximation for the objectives respectively. Thus, new algorithms were required to improve the approximation ratio. If there data is in Euclidean space, then Charikar et al. (2019) gave improve approximation ratios.

2 Preliminaries

In this section, we give a formal mathematical definition for the hierarchical clustering problem. Then the objective function is given.

Problem Input : In the hierarchical clustering problem, the input is a set V of data points. There is a distance between each pair of points i and j denoting their dissimilarity. In this paper, the data points are assumed to be located in Euclidean spaces, one particular type of metric space. For each pair of points (i, j) , the ℓ_2 distance, denoted as $d(i, j)$, is used as their dissimilarity score. The ℓ_2 distance is known to satisfy the following properties:

- (1) Convexity. The distances satisfy Jensen's inequality: for any points i, j, k , $d(\lambda \cdot i + (1 - \lambda) \cdot j, k) \leq \lambda d(i, k) + (1 - \lambda)d(j, k)$, where $\lambda \in [0, 1]$.
- (2) Triangle inequality. For any points i, j, k , $d(i, k) \leq d(i, j) + d(j, k)$.

k -means Objective : The definition of the k -means objective is the following. Given a point set S , a k -means clus-

tering partitions S into k sets S_1, S_2, \dots, S_k . The k -means objective calculates the summation over the squared norm of the distance between a point to the **centroid** of the set it belongs to: $\sum_{j=1}^k \sum_{u \in S_j} d^2(u, \rho(S_j))$. Here $\rho(S_j)$ denotes the centroid of S_j . In Euclidean space, $\rho(S_j)$ satisfies:

$$\rho(S_j) = \frac{\sum_{u \in S_j} u}{|S_j|} = \arg \min_p \sum_{u \in S_j} d^2(u, p).$$

Let $\Delta_k(S)$ denote the optimal k -means objective function value for the point set S , where k is the number of clusters. We will be particularly interested in $\Delta_2(S)$, the 2-means objective.

Fix a hierarchical clustering tree T on a set V . Consider a node of the tree and let $S \subseteq V$ be the subset of input data that is input to the current split. These will eventually be the leaves of the subtree induced by this node. We use $S \rightarrow (S_1, S_2)$ to denote a split in the tree where a set S is separated into two non-empty subsets, S_1 and S_2 . These sets correspond to the input of the two child nodes. We let $S \rightarrow (S_1, S_2) \in T$ denote that this split exists in T .

Any split $S \rightarrow (S_1, S_2)$ where S_1 and S_2 are a partition of S is a valid 2-means solution for the point set S . Since $\Delta_2(S)$ denotes the optimal objective function value, $\Delta_2(S) \leq \Delta_1(S_1) + \Delta_1(S_2)$ by definition of the 2-means objective. In particular, if $S \rightarrow (S_1, S_2)$ is the optimal 2-means solution, we have $\Delta_2(S) = \Delta_1(S_1) + \Delta_1(S_2)$.

3 Hierarchical-Revenue : Comparing Inter vs. Intra Cluster Distance

This section defines the new objective function. We call the problem of optimizing this objective the **Hierarchical-Revenue problem**.

Defining the Revenue for a Pair: Consider a node in a hierarchical clustering whose input is S and this set is split into S_1 and S_2 . A good tree ensures that the pairs of points in $i, i' \in S_1$ (resp. S_2) are more similar than pairs $i \in S_1$ and $j \in S_2$ (i.e. $d(i, j) \geq d(i, i')$). This ensures the points corresponding to the cluster at a node in the tree become more similar at lower levels of the tree. In the following, we say i and j are **split** the first time they no longer belong to the same cluster.

Every pair i and j will be eventually split in the tree and a hierarchical clustering objective should ensure they are split at the appropriate place in the tree. Further, an objective should optimize over all pairs uniformly to determine the splits.

Guided by these principles, we develop the objective as follows. We begin by allowing every pair i and j to generate one unit of revenue. This revenue can always be obtained for a fixed pair, but not necessarily for all pairs simultaneously. This unit of revenue is obtained when the pair is split at an appropriate position in the tree. Less revenue (or even 0) will be obtained when the pair is separated at a poor position. *This is the key to determine the quality of a split.*

Say that $S \rightarrow (S_1, S_2)$ is the split at some node in the tree and $i \in S_1$ and $j \in S_2$ are split. As discussed above, points in S_1 (respectively S_2) should be more similar to each other than i and j . To measure the similarity of i to other points

in S_1 we use $d(i, \rho(S_1))$, the distances of i to the centroid of S_1 . Similarly, we use $d(j, \rho(S_2))$ to measure the distance of j to points in S_2 . The distance of a point to the centroid of a set measures the distance to the average point in the set. Thus, we would like $d(i, j)$ to be larger than both $d(i, \rho(S_1))$ and $d(j, \rho(S_2))$ for it to make sense to split i and j . That is, i and j should become more similar to their respective sets after the split than they are to each other.

Formally, define the revenue for a pair of points as follows. Let $\delta_{S_1, S_2}(i, j) = \max\{d(i, \rho(S_1)), d(j, \rho(S_2))\}$ be the maximum distance of i and j to their respective centroids. We would like $\delta_{S_1, S_2}(i, j)$ to be smaller than $d(i, j)$ and therefore i and j generate a unit of revenue when this is the case. When $\delta_{S_1, S_2}(i, j) \leq d(i, j)$ we assume the revenue decays linearly. That is, the revenue is $\frac{d(i, j)}{\delta_{S_1, S_2}(i, j)}$.

Putting the above together, define the revenue for splitting i and j as $rev(i, j) = \min\{\frac{d(i, j)}{\delta_{S_1, S_2}(i, j)}, 1\}$ ⁵. This is the **revenue** i and j generates. Notice that a revenue of a unit can always be obtained since we can let j be the unique last point split from i . However, a good hierarchical splitting structure is needed to get good revenue for many pairs of points.

The Global Objective: The global objective is defined as follows. We note that while the revenue is summed over each split in the tree, obtaining a large amount of revenue at a split hinges on a good global tree structure.

Definition 3.1 (Hierarchical-Revenue). For a data set V and a given hierarchical clustering tree T , define the hierarchical tree revenue function as follows. Let $rev(S_1, S_2) = \sum_{i \in S_1} \sum_{j \in S_2} rev(i, j) = \sum_{i \in S_1} \sum_{j \in S_2} \min\{\frac{d(i, j)}{\delta_{S_1, S_2}(i, j)}, 1\}$ be the revenue over all pairs of points split across S_1 and S_2 . The aggregate revenue is $rev_T(V) = \sum_{\{i, j\} \subseteq V} rev(i, j) = \sum_{S \rightarrow (S_1, S_2) \in T} rev(S_1, S_2)$, and it should be maximized over all trees.

As is shown in Definition 3.1, there are two ways of computing $rev_T(V)$. One is to sum up the revenue over the pairs, while the other is to sum up the revenue over the splits. Both methods lead to the same value. The second form allows us to judge whether a split at some internal node of the tree is good or not compared to the number of pairs it separated.

4 Ground-truth Inputs

The work of Cohen-addad et al. (2019) gave a characterization of desirable hierarchical clustering objectives. The idea is to give a class of instances that naturally correspond to a specific hierarchical clustering tree. These trees should be optimal solutions for a good hierarchical clustering objective.

In particular, Cohen-addad et al. (2019) defined input instances that correspond to *ultrametrics*. Such inputs will be referred to as *ground-truth* inputs. For such an input, they define *generating trees*, which should be optimal for the hierarchical clustering objective to be valid. Intuitively, in an ultrametric either it is clear what the split should be at each

⁵We assume dividing by 0 gives revenue 1.

point in the tree or all splits are equivalent.⁶ The resulting tree is a generating tree.

We prove a generating tree is an optimal solution for our objective, if the input in Euclidean space is ground-truth.

4.1 Definition of Ground-Truth Inputs

We cite the following definitions from Cohen-addad et al. (2019).

Definition 4.1. A metric space (X, d) is an ultrametric if for every $x, y, z \in X$, $d(x, y) \leq \max\{d(x, z), d(y, z)\}$.

Intuitively, the definition of ultrametric implies that any three points u, v, w form an isosceles triangle, whose equal sides are at least as large as the other side. Cohen-addad et al. (2019) then defined an instance generated from ultrametric, which is treated as ground-truth input for hierarchical clustering.

Definition 4.2. An input instance on a set of points V with pairwise distance function d is *generated from an ultrametric* if the distances function d corresponds to a ultrametric.

Following Cohen-addad et al. (2019), we define *generating trees*, which are considered the most well-behaving hierarchical clustering trees for a ground-truth input.

Definition 4.3. If the instance V is generated by ultrametric, a binary tree T is a generating tree for G if it satisfies the following properties:

1. It has $|V|$ leaves and $|V| - 1$ internal nodes. Let L denote its leaves and each point in L corresponds to a unique point in V . Let \mathcal{N} denote its internal nodes, corresponding to clusters of the leaves of the subtree rooted at the node.
2. There exists a weight function $W : \mathcal{N} \mapsto \mathbb{R}_+$. For $N_1, N_2 \in \mathcal{N}$, if N_1 is on the path from N_2 to the root, $W(N_1) \geq W(N_2)$. For every $x, y \in V$, $d(x, y) = W(LCA_T(x, y))$, where $LCA_T(x, y)$ denotes the Least Common Ancestor of leaves corresponding to x and y in T .

Cohen-addad et al. (2019) proposed that for a ground-truth input graph as defined in Definition 4.2, if there exists any corresponding generating tree T , it is considered one of the best solutions among all the solutions, and thus should be one of the optimal solutions for the hierarchical clustering objective function used.

We give some intuition for why a generating tree is considered the best tree on such inputs. The value $W(LCA_T(x, y))$ can be interpreted as the distances of edges cut in split at the LCA of x and y . All points separated in a split have equal pairwise distance, the maximum pairwise distances in the current point set. Naturally, the higher up this LCA is, the larger the distance should be. For each ground-truth input graph, there is always a generating tree T , which separates the farthest pairwise points in every split.

⁶If there is a natural split then the points can be divided into two groups A and B such that inter-group distances are larger than intra-group distances. If all splits are equivalent then pairwise the points are all the same distance.

4.2 Optimality of Generating Trees

Now we prove that given an input that is generated from an ultrametric, every generating tree is optimal for Hierarchical-Revenue function introduced in this paper. In particular, every pair of points will get full revenue.

Lemma 4.1. A binary tree T , with $|V|$ leaves corresponding to the points in V and $|V| - 1$ internal nodes, is a generating tree for an instance V generated from a ultrametric if and only if it satisfies the following property:

- For every split $A \cup B \rightarrow (A, B)$ from top to bottom, $\forall i \in A, j \in B, d(i, j) = \max_{x \in A, y \in B} d(x, y)$.

Every ground-truth input has at least one generating tree, as stated in the following theorem.

Theorem 4.2. For every instance generated from some ultrametric, there is always a generating tree T as defined in Definition 4.3.

Using Lemma 4.1, the optimality of T is proved by arguing every split gives a revenue of 1 for every pair of points it separates.

Theorem 4.3. A generating tree T for an instance generated V from an ultrametric is optimal for the Hierarchical-Revenue objective.

5 Bisecting k -means Approximates the Revenue Objective

This section shows that the bisecting k -means algorithm is a constant approximation for the proposed objective. This establishes a foundational connection between a natural objective function and the bisecting k -means algorithm. This is the first analysis showing that bisecting k -means optimizes a global objective function. This helps explain the structure of the solutions produced by the algorithm.

The goal of this section is to show the following theorem.

Theorem 5.1. Fix any input set V and let T be the tree created by the bisecting k -means algorithm. The tree T is a constant approximation for the Hierarchical-Revenue objective.

The analysis is based on analyzing each split performed by bisecting k -means individually. The following lemma shows that if every split in a hierarchical clustering tree is good for the objective function proposed, then the whole tree is also good. By “good” we mean that the split gains a revenue which is at least some constant factor times the number of pairs separated. This lemma follows immediately by definition of the objective.

Lemma 5.2. A hierarchical clustering tree T is a γ -approximation for the Hierarchical-Revenue problem if it satisfies the following condition: $\forall S \rightarrow (S_1, S_2) \in T$, $rev(S_1, S_2) \geq \gamma|S_1||S_2|$ holds for some constant $\gamma > 0$.

The above lemma allows us to focus on a single iteration of the bisecting k -means algorithm. Suppose at some iteration, a cluster $A \cup B$ is split into A and B . We give the following definition of a *high-revenue point*. A point u in A is a high-revenue point if for most of the points in B , it gains acceptable amount of revenue.

Definition 5.1. Given a split $A \cup B$ and a point $u \in A$, the **high-revenue set** for u for set B is: $HR_B(u) = \{v \in B : rev(u, v) \geq \frac{1}{10}\}$. Define the **low-revenue set** for $u \in A$ to be defined as: $LR_B(u) = \{v \in B : rev(u, v) < \frac{1}{10}\} = B \setminus HR_B(u)$.

Definition 5.2. Given a split $A \cup B$, a point $u \in A$ is a **high-revenue point** if $|HR_B(u)| \geq \frac{1}{2}|B|$. Otherwise, it is called a low-revenue point.

With the definition of high-revenue points in place, the next lemma claims that given split $A \cup B \rightarrow (A, B)$ created by the optimal 2-means algorithm, if $|A| \geq |B|$, at least half of A are high-revenue points. This is the main technical lemma. This combined with Lemma 5.2 implies Theorem 5.1.

Lemma 5.3. *Let A and B be the optimal 2-means solution for the point set $A \cup B$. Without loss of generality, suppose $|A| \geq |B|$. Then, at least $\frac{1}{7}|A|$ points in A are high-revenue. This gives a lower bound of at least $\frac{1}{35}|A||B|$ revenue in total for splitting A and B .*

The rest of the section is devoted to proving Lemma 5.3 by contradiction, with proofs partially omitted due to space limits. For the rest of the section fix a set $A \cup B$ and let the partition A, B correspond to the optimal solution to the 2-means problem on $A \cup B$. For sake of contradiction suppose more than $\frac{3}{7}|A|$ points in A are low-revenue points. We will show that such a split $A \cup B \rightarrow (A, B)$ cannot be optimal for the 2-means objective. Indeed, we will show that another split has a smaller 2-means objective value, proving the lemma.

Say we have $i \in A$ and $j \in B$, such that $rev(i, j) < \frac{1}{10}$. Let H be the hyperplane such that $H = \{y : d(y, \rho(A)) = d(y, \rho(B))\}$. Then, H separates the Euclidean space into two half-spaces: $H^+ = \{y : d(y, \rho(A)) \geq d(y, \rho(B))\}$ and $H^- = \{y : d(y, \rho(A)) \leq d(y, \rho(B))\}$. By the assumption that the split $A \cup B \rightarrow (A, B)$ is the optimal 2-means solution, we have $A \subseteq H^+$, and $B \subseteq H^-$. Next we show the following structural lemma. This lemma says that if $rev(i, j)$ is small then $d(i, \rho(A))$ and $d(j, \rho(B))$ are within a constant factor of each other, which is close to 1. Geometrically, this implies that both i and j are located close to the hyperplane H . See Figure 5 for an illustration. The following lemma's proof is in the appendix.

Lemma 5.4. *Consider any $i \in A$ and $j \in B$. If $rev(i, j) < \frac{1}{10}$, we have $\frac{9}{10}d(i, \rho(A)) < d(j, \rho(B)) < \frac{10}{9}d(i, \rho(A))$, and $\frac{9}{10}d(j, \rho(B)) < d(i, \rho(A)) < \frac{10}{9}d(j, \rho(B))$, and $d(i, j) < \frac{1}{9} \min\{d(i, \rho(A)), d(j, \rho(B))\}$.*

Let S be the subset of low-revenue points in A . By assumption, $|S| > \frac{3}{7}|A|$. The next lemma establishes that any two points in S are very close to each other as compared to their distance to the centroid $\rho(A)$. The following lemma's proof is in the appendix.

Lemma 5.5. *Let S be the low-revenue points in A . For any two points $u, v \in S$, $d(u, v) \leq \frac{2}{9} \max\{d(u, \rho(A)), d(v, \rho(A))\}$.*

Let x be the point in S such that $x \in \arg \max_{u \in S} d(u, \rho(A))$, the farthest points from $\rho(A)$ in S . Notice that, $d(x, \rho(A)) > 0$, otherwise it implies S is overlapping with $\rho(A)$, for any $u \in S$ and $v \in LR_B(u)$,

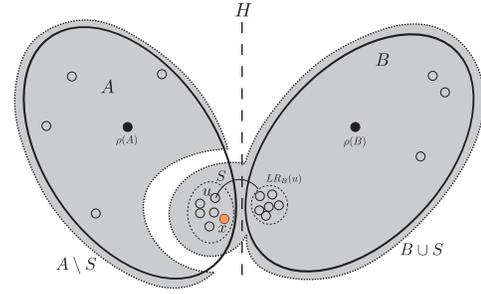


Figure 3: Proof by constructing a better 2-means solution. The bold dashed line in the middle is the hyperplane H . The two bold ellipses are clusters A and B , respectively. The dashed ellipse in A is the set S , and the dashed ellipse in B is the low-revenue set $LR_B(u)$ for point $u \in A$ in B . S and $LR_B(u)$ are both close to the separating hyperplane H . A new partition $A \cup B \rightarrow (A \setminus S, B \cup S)$ is constructed, represented by the two grey areas.

by Lemma 5.4, we have $d(v, \rho(B)) = 0$, but this implies $rev(u, v) = 1$. Therefore, $d(x, \rho(A)) > 0$.

Lemma 5.5 implies that $\forall u \in S, d(u, x) \leq \frac{2}{9}d(x, \rho(A))$. This result tells us the set S is contained in a ball centered at x , with radius $\frac{2}{9}d(x, \rho(A))$. So we can bound the distance between centroid of S , $\rho(S)$ and $\rho(A)$ using convexity of the ℓ_2 norm. The following lemma's proof is omitted due to space.

Lemma 5.6. *Let S be the low-revenue points in A and $x \in \arg \max_{u \in S} d(u, \rho(A))$. It is the case that $d(\rho(S), \rho(A)) \geq \frac{7}{9}d(x, \rho(A))$.*

This is proved by combining Lemma 5.5 with the convexity of ℓ_2 norm. Notice that $\rho(A)$ is a convex combination of all points in A , Jensen's inequality gives us the conclusion. Since we proved $\rho(S)$ is far from $\rho(A)$. Next, we upper-bound $d(\rho(S), \rho(B))$. Recall that points in the set S are far away from $\rho(A)$, but close to the hyperplane $H = \{y : d(y, \rho(A)) = d(y, \rho(B))\}$. The following lemma's proof is in the appendix.

Lemma 5.7. *Let S be the low-revenue points in A . For any $u \in S$, $d(u, \rho(B)) \leq \frac{11}{9}d(x, \rho(A))$.*

Therefore, we can upper bound $d(\rho(S), \rho(B))$: $d(\rho(S), \rho(B)) \leq \frac{\sum_{u \in S} d(u, \rho(B))}{|S|} \leq \frac{11}{9}d(x, \rho(A))$. The first inequality follows by definition of a centroid. The second from Lemma 5.7. This, combined with $d(\rho(S), \rho(A)) \geq \frac{7}{9}d(x, \rho(A))$ from Lemma 5.6, gives us the following: $\frac{d^2(\rho(S), \rho(A))}{d^2(\rho(S), \rho(B))} \geq (\frac{7}{9})^2 / (\frac{11}{9})^2 = \frac{49}{121}$. Recall that $\Delta_k(U)$ denotes the optimal k -means value for a set U . Let S_1 and S_2 be two sets. We quote the following lemma from Ostrovsky et al. (2012).

Lemma 5.8 ((Ostrovsky et al. 2012)). *For any two sets of points S_1 and S_2 it is the case that $\Delta_1(S_1 \cup S_2) = \Delta_1(S_1) + \Delta_1(S_2) + \frac{|S_1||S_2|}{|S_1|+|S_2|}d^2(\rho(S_1), \rho(S_2))$.*

With this in place Lemma 5.3 can be shown. In general, we show this by take the set S away from A and assign it

into cluster B instead, and prove that this is a better 2-means solution than the previous one. Due to space, the proof of Lemma 5.3 is omitted.

6 Randomly Partitioning Poorly Approximates the Revenue Objective

Consider the following algorithm which can create undesirable trees. The *Random* algorithm splits a set S into (S_1, S_2) by flipping an independent, fair coin for each point in S . If the coin comes up heads then the point gets added to S_1 , and otherwise gets added to S_2 . The algorithm is intuitively undesirable because it does not take the structure of the input into the construction of the solution. Further, the solutions produced do not give much insight into the data.

While intuitively bad, this algorithm is known to be a $\frac{1}{3}$ -approximation for the objective of Moseley and Wang (2017) with similarity scores and it is a $\frac{2}{3}$ -approximation for CKMM objective for dissimilarity scores. These results hold for any set of similarity or dissimilarity scores, regardless of if they form a metric.

We show that the our objective does not have this shortcoming. The approximation ratio of the Random algorithm is at most $O(\frac{1}{n^\epsilon})$ for a constant $\epsilon > 0$, indicating that it performs very poorly, as is stated by Theorem 6.1. Proof is omitted.

Theorem 6.1. *Let $OPT(V)$ be the optimal solution for V . Let the expected revenue be $\mathbb{E}_T[rev_T(V)]$ for set V . Then, there exists a construction of V , such that for a constant $\epsilon \in (0, 1)$, $\mathbb{E}_T[rev_T(V)] = O(\frac{1}{n^\epsilon}) \cdot OPT(V)$.*

7 Objectives for Data in Metric Space

This section studies data with similarity/dissimilarity scores in a metric space. First we investigate the CKMM objective for hierarchical clustering on point sets using dissimilarity scores. Recall that this objective is the same as the Dasgupta (Dasgupta 2016) objective except the minimization is swapped for a maximization and the similarity scores are swapped for dissimilarity scores. We also study the Dasgupta objective (Dasgupta 2016) for similarity scores. We show for each case that if the pairwise similarity/dissimilarity scores form *any metric*, then *every tree* is a at most a factor 2 from optimal.

For a tree T let $T[i \vee j]$ denote the subtree rooted at the least common ancestor of i and j , and $|\text{leaves}(T[i \vee j])|$ denote the number of leaves of $T[i \vee j]$. Recall that the CKMM objective is the following: $\max_T \sum_{i,j \in V} d(i,j) |\text{leaves} T[i \vee j]|$.

In Cohen-addad et al. (2019), it has been proved that any solution is a constant approximation of the optimal solution for CKMM objective, given that the distance is a metric. Here we prove a stronger conclusion:

Theorem 7.1. *Any solution is a $\frac{1}{2}$ -approximation for CKMM objective if the distance $d(i, j)$ is a metric, i.e., it satisfies triangle inequality.*

Next consider the objective in Dasgupta (Dasgupta 2016). Here each pair of points i and j have a similarity score w_{ij} where higher weights mean points are more similar. Recall from the introduction that Dasgupta’s objective is $\min_T \sum_{i,j \in V} w_{ij} |\text{leaves} T[i \vee j]|$.

We show the following corollary that follows from the proof of the prior theorem.

Corollary 7.1.1. *If the similarity score in the setting of Dasgupta (2016) is a metric, any hierarchical clustering tree is a 2-approximation for the objective in Dasgupta (2016): $\min_T cost_T(V) = \sum_{1 \leq i < j \leq n} w_{ij} |\text{leaves}(T[i \vee j])|$.*

We note that for similarity scores, it is not a standard assumption that data lies in a metric space. Thus, this corollary is perhaps interesting to understand the structure of the objective. However, it does not suggest that any tree will be 2-approximate for most data sets with similarity scores.

8 Empirical Results

The goal of this section is to study the performance of different algorithms for the new objective empirically. The experimental results support the following claims:

- Algorithms that are popular in practice give high revenue for the new objective, with bisecting k-means performing the best. This demonstrates that the objective value is correlated with algorithms that perform well and highly connected to the bisecting k-means algorithm, as the theory suggests.
- Random algorithm, as mentioned in previous section, performs poorly for the new objective.

Data sets: We use two data sets from the UCI data repository: *Census*⁷ and *Bank*⁸. Only the numerical features are used.

Algorithms studied: We study four algorithms⁹: bisecting k-means, average-linkage, single-linkage, and Random. In each experiment, we subsample 2000 data points from the given data set and run the algorithms with subsampled data. We conduct five experiments with each data set and report the mean and variance. Since optimal 2-means solution is intractible, in practice we import the k-means implementation from package *Scikit-learn*¹⁰, which uses Lloyd’s algorithm seeded with k-means++ for each split.

Results: Table 1 shows the comparison between performance for our objective and the CKMM objective. For each algorithm, the columns $(\hat{\mu}_1, \hat{\sigma}_1)$ and $(\hat{\mu}_2, \hat{\sigma}_2)$ denote the mean and standard deviation for our objective and CKMM objective respectively, calculated over results of the five experiments.

Regarding the new objective, the results show bisecting k-means performs the best of the four algorithms for it. Further, bisecting k-means is within 1% of the upper bound on the optimal solution for the objective, which is the total number of pairs of data points. This suggests that the objective is closely related to bisecting k-means, as the theory suggests. It also shows that experimentally bisecting k-means performs much better than the approximation ratio established.

⁷<https://archive.ics.uci.edu/ml/datasets/census+income>

⁸<https://archive.ics.uci.edu/ml/datasets/Bank+Marketing>

⁹https://github.com/wangyuyan2333/hier_clustering_split_rev_obj_test.git

¹⁰<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

Algorithm	$(\hat{\mu}_1, \hat{\sigma}_1)$ -Census	$(\hat{\mu}_2, \hat{\sigma}_2)$ -Census	$(\hat{\mu}_1, \hat{\sigma}_1)$ -Bank	$(\hat{\mu}_2, \hat{\sigma}_2)$ -Bank
bisecting k-means	(4.931e5, 304.980)	(1.094e12, 1.714e11)	(4.912e5, 474.451)	(1.049e12, 1.158e11)
average-linkage	(4.900e5, 1.151e3)	(1.093e12, 1.710e11)	(4.907e5, 802.665)	(1.052e12, 1.163e11)
single-linkage	(4.869e5, 1.392e3)	(1.094e12, 1.712e11)	(4.818e5, 1.365e3)	(1.035e12, 1.168e11)
Random	(1.311e5, 1.072e4)	(7.463e11, 1.152e11)	(3.339e5, 8.825e3)	(7.789e11, 7.993e10)
upper bound	(499500, 0)	(1.119e12, 1.725e11)	(499500, 0)	(1.167e12, 1.199e11)

Table 1: Summary of stats for all algorithms, on Census and Bank

All the three algorithms which are popular in practice perform well for our objective, with bisecting k-means performing marginally better than average-linkage and single-linkage on average. Moreover, bisecting k-means also has the smallest standard deviation across different subsamples. Random is significantly worse, with potentially over 30 times more loss compared to optimal than the other algorithms. This perhaps suggests that trees created by good algorithms perform well for the objective and poorly constructed trees do not perform well.

Compared with the CKMM objective from prior work, the results further show that average-linkage performs slightly better than bisecting k-means for the CKMM objective. This result matches the theory, which suggests this objective is closer to average-linkage than bisecting k-means. Again, all three algorithms used in practice perform well for CKMM. However, Random also gives about 2/3 of the upper-bound, as the theoretical bound suggests. This perhaps shows the CKMM objective gives similar judgements on algorithm performance with our objective, the latter showing a more significant gap between Random and the other three algorithms.

9 Conclusion

This paper gives a new objective function for hierarchical clustering designed to mathematically capture the principle used to motivate most divisive algorithms. That is, comparing inter vs. intra cluster distances at splits in the tree.

The paper proved a close relationship between the objective and the bisecting k -means algorithm. This was done by showing the bisecting k -means provably optimizes the objective. This helps to understand the structures of trees produced using bisecting k -means.

The results in this paper leave directions for future work. How tight can the approximation ratio be for the k -means algorithm? How do other hierarchical clustering algorithms perform for this objective? Can we improve on the bisecting k -means algorithm to better optimize the objective?

References

Ackerman, M., and Ben-David, S. 2016. A characterization of linkage-based hierarchical clustering. *Journal of Machine Learning Research* 17:232:1–232:17.

Awasthi, P.; Blum, A.; and Sheffet, O. 2012. Center-based clustering under perturbation stability. *Inf. Process. Lett.* 112(1-2):49–54.

Balcan, M.; Blum, A.; and Gupta, A. 2013. Clustering under approximation stability. *J. ACM* 60(2):8:1–8:34.

Balcan, M.; Blum, A.; and Vempala, S. 2008. A discriminative framework for clustering via similarity functions. In *Proceedings of STOC*, 671–680.

Carlsson, G. E., and Mémoli, F. 2010. Characterization, stability and convergence of hierarchical clustering methods. *Journal of Machine Learning Research* 11:1425–1470.

Charikar, M., and Chatziafratis, V. 2017. Approximate hierarchical clustering via sparsest cut and spreading metrics. In *Proceedings of SODA*, 841–854.

Charikar, M.; Chatziafratis, V.; Niazadeh, R.; and Yaroslavtsev, G. 2019. Hierarchical clustering for euclidean data. In Chaudhuri, K., and Sugiyama, M., eds., *PMLR*, volume 89, 2721–2730. PMLR.

Charikar, M.; Chatziafratis, V.; and Niazadeh, R. 2019. Hierarchical clustering better than average-linkage. In *Proceedings of SODA*, 2291–2304.

Chatziafratis, V.; Niazadeh, R.; and Charikar, M. 2018. Hierarchical clustering with structural constraints. In *Proceedings of the ICML*, 773–782.

Cohen-addad, V.; Kanade, V.; Mallmann-trenn, F.; and Mathieu, C. 2019. Hierarchical clustering: Objective functions and algorithms. *J. ACM* 66(4):26:1–26:42.

Dasgupta, S., and Long, P. M. 2005. Performance guarantees for hierarchical clustering. *J. Comput. Syst. Sci.* 70(4):555–569.

Dasgupta, S. 2016. A cost function for similarity-based hierarchical clustering. In *STOC*, 118–127.

Ghoshdastidar, D.; Perrot, M.; and von Luxburg, U. 2018. Foundations of comparison-based hierarchical clustering. *CoRR* abs/1811.00928.

Moseley, B., and Wang, J. 2017. Approximation bounds for hierarchical clustering: Average linkage, bisecting k-means, and local search. In *Advances in Neural Information Processing Systems, 2017.*, 3097–3106.

Murtagh, F., and Contreras, P. 2012. Algorithms for hierarchical clustering: an overview. *Wiley Interdisc. Rev.: Data Mining and Knowledge Discovery* 2(1):86–97.

Murtagh, F. 1983. A survey of recent advances in hierarchical clustering algorithms. *Comput. J.* 26(4):354–359.

Ostrovsky, R.; Rabani, Y.; Schulman, L. J.; and Swamy, C. 2012. The effectiveness of lloyd-type methods for the k-means problem. *Journal of the ACM (JACM)* 59(6):28.

Plaxton, C. G. 2006. Approximation algorithms for hierarchical location problems. *J. Comput. Syst. Sci.* 72(3):425–443.

Roy, A., and Pokutta, S. 2017. Hierarchical clustering via spreading metrics. *JMLR* 18:88:1–88:35.

Steinbach, M.; Karypis, G.; Kumar, V.; et al. 2000. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, 525–526. Boston.

Zhao, Y.; Karypis, G.; and Fayyad, U. M. 2005. Hierarchical clustering algorithms for document datasets. *Data Min. Knowl. Discov.* 10(2):141–168.