Bisect and Conquer: Hierarchical Clustering via Max-Uncut Bisection

Sara Ahmadian
Google Research, NY

Vaggos Chatziafratis
Stanford University, CA

Alessandro Epasto
Google Research, NY

Euiwoong Lee
New York University, NY

Mohammad Mahdian
Google Research, NY

Konstantin Makarychev
Northwestern University, IL

Grigory Yaroslavtsev
Indiana University, IN

Abstract

Hierarchical Clustering is an unsupervised data analysis method which has been widely used for decades. Despite its popularity, it had an underdeveloped analytical foundation and to address this, Dasgupta recently introduced an optimization viewpoint of hierarchical clustering with pairwise similarity information that spurred a line of work shedding light on old algorithms (e.g., Average-Linkage), but also designing new algorithms. Here, for the maximization dual of Dasgupta's objective (introduced by Moseley-Wang), we present polynomial-time 0.4246 approximation algorithms that use MAX-UNCUT BISECTION as a subroutine. The previous best worst-case approximation factor in polynomial time was 0.336, improving only slightly over Average-Linkage which achieves 1/3. Finally, we complement our positive results by providing APX-hardness (even for 0-1 similarities), under the SMALL SET EXPANSION hypothesis.

1 Introduction

Hierarchical Clustering (HC) is a popular unsupervised learning method which produces a recursive decomposition of a dataset into clusters of increasingly finer granularity. The output of HC is a hierarchical representation of the dataset in the form of a tree (a.k.a. dendrogram) whose leaves correspond to the data points. The internal nodes of the tree correspond to clusters organized in a hierarchical fashion.

Since HC captures cluster structure at all levels of granularity simultaneously, it offers several advantages over a basic flat clustering (a partition of the data into a fixed number of clusters) and it has been used for several decades (see e.g., Ward's method [WJ63]). It is one of the most popular methods across a wide range of fields e.g., in phylogenetics [SS62, JS68], where many of the so-called *linkage-based* algorithms (like Average/Single/Complete-Linkage) originated, in gene expression data analysis [ESBB98], the analysis of social networks [LRU14, MMO08], bioinformatics [DBE+15], image and text classification [SKK+00], and even in financial markets [TLM10]. See classic texts [JMF99, MRS08, Jai10] for a standard introduction to HC and linkage-based methods. Due to the importance of HC, many variations (including linkage-based methods) are also currently implemented in standard scientific computing packages and in large-scale systems [BBD+17].

Despite the plethora of applications, until recently there wasn't a concrete objective associated with HC methods (except for the Single-Linkage clustering which enjoys a simple combinatorial structure due to the connection to minimum spanning trees [GR69]). This is in stark contrast with flat clustering methods where k-means, k-median, k-center constitute some standard objectives. Recent breakthrough work [Das16] by Dasgupta addressed this gap by introducing the following objective function:

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. Copyright 2020 by the author(s).

Definition 1.1 (HC Objective [Das16])**.** *Given a similarity matrix with entries $w_{ij} \geq 0$ corresponding to similarities between data points i and j, let the hierarchical clustering objective for a tree $\mathcal{T}$ be defined as:*

$$\mathcal{F}^{-}(\mathcal{T}) = \sum_{i<j} w_{ij} |\mathcal{T}(i,j)|,$$

where $|\mathcal{T}(i,j)|$ denotes the number of leaves under the least common ancestor of i and j.

The goal of HC under Dasgupta's objective is to minimize the function $\mathcal{F}^{-}(\mathcal{T})$ among all possible trees. Intuitively, the objective encourages solutions that do not separate similar points (those with high w_{ij}) until the lower levels of the tree; this is because $|\mathcal{T}(i,j)| = n$, if i, j are separated at the top split of the tree $\mathcal{T}$, whereas $|\mathcal{T}(i,j)| = 2$, if i, j are separated at the very last level.

Dasgupta [Das16] further showed that many desirable properties hold for this objective with respect to recovering ground truth hierarchical clusterings. This was later strengthened both theoretically and experimentally [CAKMTM19,RP16]. Although it is not hard to see that the optimum tree should be binary, it is not clear how one can optimize for it given the vast search space. Surprisingly, [Das16] established a connection with a standard graph partitioning primitive, Sparsest-Cut, which had been previously used to obtain HC in practice [MMO08]. Later work [CC17] further showed a black-box connection: an α-approximation for Sparsest-Cut or Balanced-Cut gives an $O(\alpha)$-approximation for $\mathcal{F}^{-}(\mathcal{T})$. Hence an $O(\sqrt{\log n})$-approximation can be computed in polynomial time by using the celebrated result of [ARV08]. A constant-factor hardness is also known under the Small Set Expansion hypothesis [CC17, RP16].

Building on Dasgupta's work, Moseley and Wang [MW17] gave the first approximation analysis of the Average-Linkage method. Specifically, for the complement of Dasgupta's objective (see Definition 2.1), they proved that Average-Linkage achieves a 1/3 approximation in the worst case. That factor was marginally improved to 0.3363 in [CCN19] via an ad-hoc semidefinite programming formulation of the problem. This was the state-of-the-art in terms of approximation prior to this work and was arguably complicated and impractical.

Contributions: In this paper, we extend the recent line of research initiated by Dasgupta's work on objective-based hierarchical clustering for similarity data, by significantly beating the previous best-known approximation factor for the [MW17] objective and doing so with natural algorithms. Our algorithm is based on a combination of Max-Uncut Bisection and Average-Linkage and guarantees 0.4246 of the value of the optimum hierarchical clustering, whereas previous work based on semidefinite programming [CCN19] only achieved 0.3363. An advantage of our algorithm is that it uses the Max-Uncut Bisection primitive as a black-box and the approximation ratio gracefully degrades as a function of the quality of this primitive; this is in contrast with previous approaches [CCN19] which solve complicated convex relaxations tailored to the objective. Since both theoretical and practical solutions for Max-Uncut Bisection are readily available in [ABG16, SS13], this results in a family of algorithms which can be analyzed both rigorously and empirically. We also complement our algorithmic results with hardness of approximation (APX-hardness): assuming the Small Set Expansion hypothesis, we prove that even for 0-1 similarities, there exists $\varepsilon > 0$, such that it is NP-hard to approximate the [MW17] objective within a factor of $(1 - \varepsilon)$. A summary of our results compared to the previous work is given in Table 1. Here we also point out that $\frac{1}{3}$ is a simple baseline achieved by a random binary tree [MW17] and hence our work gives the first major improvement over this baseline.

	[MW17]	[CCN19]	Our paper
Approx.	1/3	0.3363	**0.4246**
Hardness	NP-hard	-	**APX-hard** under SSE

Table 1: Our results for the [MW17] objective.

Further Related Work: HC has also been studied in the "semi-supervised" or "interactive" case, where prior knowledge or expert advice is available, with or without the geometric information on the data points. Examples of these works include [EDSN11, EZK18], where they are interested in minimizing the number of (pairwise or triplet) queries necessary to determine a ground truth HC, and [VD16, CNC18] who provide techniques for incorporating triplet constraints (the analog of split/merge feedback [BB08, ABV17] or must-link/cannot-link constraints [WCR$^+$01, WC00] in standard flat clustering) to get better hierarchical trees. Furthermore, assuming the data points lie in a metric space (instead of w_{ij} being arbitrary similarities), there are previous works that measure the quality of a HC using standard flat-clustering objectives like k-means, k-median or k-center as proxies [CCFM04, Das02, Pla06, LNRW10] in order to get approximation guarantees. Finally, in high dimensions when even running simple algorithms like single-linkage becomes impractical due to the exponential dependence on the dimension [YV17], one can still efficiently achieve good approximations if the similarities are generated with the fairly common Gaussian Kernel and other smooth similarity measures, as is shown in [CCNY18].

Organization: We start by providing the necessary background in Section 2. Then, we give our 0.4246 approximation algorithm for HC based on Max-Uncut Bisection in Section 3 and our hardness of approximation result in Section 4. Our conclusion is given in Section 5.

2 Preliminaries

We begin by setting the notation used throughout the paper. The input to the HC problem is given by an (explicit or implicit) $n \times n$ matrix of pairwise similarities with non-negative entries $w_{ij} \geq 0$ corresponding to the similarity between the i-th and j-th data point (no triangle inequality is assumed for the similarities). Sometimes we also refer to the underlying weighted graph as $G(V, E, w)$.

We denote by $\mathcal{T}$ a rooted tree whose leaves correspond to the n vertices. For two leaves i, j, $\mathcal{T}(i, j)$ denotes the subtree rooted in the least common ancestor of i and j in $\mathcal{T}$ and $|\mathcal{T}(i, j)|$ the number of leaves contained in $\mathcal{T}(i, j)$.

For a set $S \subseteq V$, $w(S) := \sum_{(i<j) \in S \times S} w_{ij}$ denotes the total weight of pairwise similarities inside S. For a pair of disjoint sets $(S, T) \in V \times V$, $w(S, T) := \sum_{i \in S, j \in T} w_{ij}$ denotes the total weight of pairwise similarities between S and T.

Definition 2.1 (HC Objective [MW17])**.** *For a tree $\mathcal{T}$ consider the hierarchical clustering objective:*

$$\mathcal{F}^{+}(\mathcal{T}) = \sum_{1 \leq i < j \leq n} w_{ij}(n - |\mathcal{T}(i, j)|) \qquad (*)$$

If we denote the total weight of the graph by $W := \sum_{i<j} w_{ij}$, then note that a trivial upper bound on the value of this objective is $(n-2)W$.

Average-Linkage: One of the main algorithms used in practice for HC, it starts by merging clusters of data points that have the highest average similarity. It is known that it achieves 1/3 approximation for the Moseley-Wang objective and this is tight in the worst case[1] [CCN19]. For a formal description, please refer to Algorithm 1.

Max-Uncut Bisection: This is the complement problem to Min-Cut Bisection (which is perhaps more standard in the literature), and the goal here is to split the vertices of a weighted graph into two sets $(S, \bar{S})$, such that the weight of uncut edges $\sum_{ij \in E} w_{ij} - \sum_{i \in S, j \in \bar{S}} w_{ij}$ is maximized. It is known that one can achieve at least .8776 of the optimum value in polynomial time [ABG16, WDX15].

Algorithm 1 Average-Linkage

```
input: Similarity matrix $w \in \mathbb{R}^{n \times n}_{\geq 0}$.
Initialize clusters $\mathcal{C} \leftarrow \cup_{v \in V} \{v\}$.
while $|\mathcal{C}| \geq 2$ do
    Pick $A, B \in \mathcal{C}$ to maximize:
        $w(A, B) := \frac{1}{|A||B|} \sum_{a \in A, b \in B} w_{ab}$
    Set $\mathcal{C} \leftarrow \mathcal{C} \cup \{A \cup B\} \setminus \{A, B\}$
end while
```

Algorithm 2 HC via Max-Uncut Bisection

```
input: Similarity matrix $w \in \mathbb{R}^{n \times n}_{\geq 0}$.
Partition the underlying graph on $n$ vertices
    with edges weighted by $w$ into two parts $S$ and $\bar{S}$
    using Max-Uncut Bisection as a black box.
    This creates the top split in the hierarchical clus-
    tering tree.
Run Average-Linkage on $S$ and on $\bar{S}$ to get
    trees $\mathcal{T}_S$ and $\mathcal{T}_{\bar{S}}$.
Construct the resulting HC tree by first splitting
    into $(S, \bar{S})$, then building trees $\mathcal{T}_S$ and $\mathcal{T}_{\bar{S}}$ on the
    respective sets.
```

3 Our 0.4246 approximation for HC

Our algorithm, based on Max-Uncut Bisection and Average-Linkage, is simple to state and is given in Algorithm 2. It starts by finding an approximate solution to Max-Uncut Bisection, followed by Average-Linkage agglomerative hierarchical clustering in each of the two pieces[2]. Our main result is that the better between Algorithm 1 and Algorithm 2 will produce a binary tree which achieves 0.4246 of the optimum:

Theorem 3.1. *Given an instance of hierarchical clustering, either Algorithm 1 or Algorithm 2 will*

[1] As is common in worst-case analysis of algorithms, the 1/3 tightness for Average-Linkage is based on a rather brittle pathological counterexample, even though its performance is much better in practice (see for example [CCNY18]).

[2] With no change in approximation, one can also run Max-Uncut Bisection recursively, however running Average-Linkage is typically substantially more efficient.

output a tree achieving $\frac{4\rho}{3(2\rho+1)} - o(1) \geq 0.4246$ *(for* $\rho = 0.8776$*) of the optimum according to the objective (**), if a* ρ*-approximation for the* Max-Uncut Bisection *problem is used as a black-box.*

Remark: The current best approximation factor achievable for Max-Uncut Bisection in polynomial time is $\rho = 0.8776$. This makes our analysis almost tight, since one can't get better than 0.444 even by using an exact Max-Uncut Bisection algorithm (with $\rho = 1$).

3.1 Overview of the proof

Before delving into the technical details of the main proof, we present our high-level strategy through a series of 4 main steps:

Step 1: Consider a binary[3] tree $\mathcal{T}^*$ corresponding to the optimal solution for the hierarchical clustering problem and let $\texttt{OPT} = \mathcal{F}^+(\mathcal{T}^*)$ be the value of the objective function for this tree. Note that there exists a subtree $\widehat{\mathcal{T}^*}$ in this tree which contains more than $n/2$ leaves while its two children contain at most $n/2$ leaves each (see Figure 1). Given this decomposition of the optimum tree into three size restricted sets A, B, C, we provide an upper bound for $\texttt{OPT}$ as a function of the weight inside and across these sets (see Proposition 3.2). We then need to do a case analysis based on whether the weight across or inside these sets is larger.

Step 2: In the former case, things are easy as one can show that $\texttt{OPT}$ is *small* and that the contribution from Average-Linkage alone yields a $\frac{4}{9}$-approximation. This is carried out in Proposition 3.4 based on the Fact 3.3.

Step 3: In the latter case, we show that there exists a split of the graph into two exactly equal pieces, so that the weight of the uncut edges is relatively *large*. This is crucial in the analysis as having a good solution to the Max-Uncut Bisection directly translates into a high value returned by the ρ-approximate black box algorithm (see Lemma 3.5, Proposition 3.6 and Proposition 3.7).

Step 4: Finally, from the previous step we know that the returned value of the black box is large, hence taking into account the form of the HC objective, we can derive a lower bound for the value our Algorithm 2. The proof of the main theorem is then completed by Proposition 3.8 and Lemma 3.9.

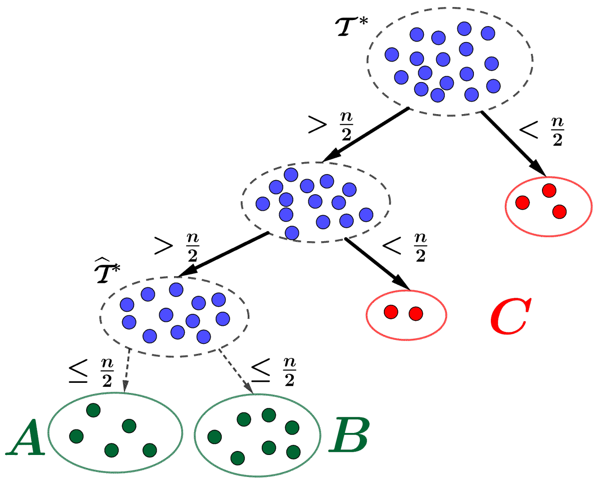

Figure 1: Splitting $\mathcal{T}^*$ to size restricted sets A, B, C.

3.2 Proof of Theorem 3.1

For ease of presentation, we assume n is even (to avoid the floor/ceiling notation) and we omit the $o(1)$ terms.

Proposition 3.2. *Let A be the set of leaves in the left subtree of $\widehat{\mathcal{T}^*}$, let B the set of leaves in the right subtree of $\widehat{\mathcal{T}^*}$ and $C = V \setminus (A \cup B)$ be the set of leaves outside of $\widehat{\mathcal{T}^*}$. Then*[4]*:*

$$\texttt{OPT} \leq (w(A) + w(B) + w(C)) \cdot (n-2) +$$

$$+ (w(A,B) + w(B,C) + w(A,C)) \cdot |C|$$

Proof. For an edge (i, j) whose endpoints lie in the same cluster (i.e., A, B or C), its contribution to the objective is at most $w_{ij}(n-2)$, using the trivial upper bound of $(n-2)$ for the non-leaves term of the objective. Consider any pair of leaves $(i, j) \in A \times B$ in $\mathcal{T}^*$. The least common ancestor for this pair is the root of $\widehat{\mathcal{T}^*}$ and hence the contribution of this pair to the objective is equal to $w_{ij}(n - |\widehat{\mathcal{T}^*}|) = w_{ij}|C|$. Similarly, for any pair of leaves $(i, j) \in A \times C$ (or in $B \times C$), their least common ancestor is a predecessor of the root of $\widehat{\mathcal{T}^*}$ and hence the contribution of this pair to the objective is at most $w_{ij}(n - |\widehat{\mathcal{T}^*}|) = w_{ij}|C|$. The desired bound now follows by summing up all the contributions of all distinct pairs of leaves. □

From now on, let $\alpha := w(A) + w(B) + w(C)$ and let $\beta := w(A,B) + w(B,C) + w(A,C)$ denote the total weights of similarities inside the three sets and

[3] W.l.o.g. the optimal tree can be made binary.

[4] By definition, A, B contain at most $\frac{n}{2}$ leaves, while C contains strictly fewer than $\frac{n}{2}$ leaves.

crossing a pair of these sets respectively. Note the total weight of all similarities is $W = \alpha + \beta$.

Fact 3.3 (Average-Linkage [MW17]). *The* Average-Linkage *algorithm gives a solution whose* $\mathcal{F}^+$ *objective is at least* $\frac{1}{3}W(n-2) = \frac{1}{3}(\alpha + \beta)(n-2)$.

Proposition 3.4. *If* $\alpha \leq \beta$, *our Algorithm 2 outputs a solution of value at least* $4/9\mathit{OPT} \geq 0.44\mathit{OPT}$, *where* OPT *denotes the HC value of any optimum solution.*

Proof. Recall that by definition of C it holds that $|C| < \frac{n}{2} \implies |C| \leq \frac{n}{2} - 1 \leq \frac{n-2}{2}$. Hence by Proposition 3.2 we have $\mathtt{OPT} \leq \alpha(n-2) + |C| \cdot \beta \leq \alpha(n-2) + \frac{n-2}{2}\beta$. On the other hand, by Fact 3.3, Average-Linkage outputs a solution whose expected value is $\frac{1}{3}(\alpha+\beta)(n-2)$. We have $\frac{1}{3}(\alpha+\beta)(n-2) - \frac{4}{9}(\alpha(n-2) + \frac{n-2}{2}\beta) = \frac{1}{9}(\beta - \alpha) \geq 0$. Hence, just running Average-Linkage alone (i.e., Algorithm 1) gives a $\frac{4}{9}$-approximation in this case. □

Lemma 3.5. *Suppose* $\alpha \geq \beta$. *Then, there exists a balanced cut* (L, R) *of the nodes in* G, *such that the weight of the uncut edges is at least* $\alpha - (\alpha - \beta)\delta_{max}(c)$, *where* $c = |C|/n$ *and* $\delta_{max}(c) = \frac{c(1-2c)}{(1-3c^2)}$.

Proof. For the partition (A, B, C) we will refer to edges whose endpoints are both inside one of the three sets as *red* edges (i.e., $(i,j) \in (A \times A) \cup (B \times B) \cup (C \times C)$). We refer to the edges whose two endpoints are contained in two different sets as *blue* edges (i.e., $(i,j) \in (A \times B) \cup (A \times C) \cup (B \times C)$). Our goal here is to give a randomized partitioning scheme that produces the bisection (L, R) with high value of uncut weight lying inside L, R.

For simplicity, recall that n is even. The case of odd n is handled similarly. Denote $a = |A|/n$ and $b = |B|/n$. Let $\widetilde{a} = 1/2 - a$, $\widetilde{b} = 1/2 - b$, and $\widetilde{c} = 1/2 - c$. Note that $\widetilde{a}$, $\widetilde{b}$ are non-negative, and $\widetilde{c}$ is strictly positive due to the size restrictions. Define:

$$q_A = \frac{2\widetilde{b}\widetilde{c}}{(\widetilde{b}+\widetilde{c})^2}, \quad q_B = \frac{2\widetilde{a}\widetilde{c}}{(\widetilde{a}+\widetilde{c})^2}, \quad q_C = \frac{2\widetilde{a}\widetilde{b}}{(\widetilde{a}+\widetilde{b})^2},$$

and

$$p_A = \frac{q_B q_C}{q_A q_B + q_B q_C + q_A q_C},$$

$$p_B = \frac{q_A q_C}{q_A q_B + q_B q_C + q_A q_C},$$

$$p_C = \frac{q_A q_B}{q_A q_B + q_B q_C + q_A q_C}.$$

We also denote the following expression by δ:

$$\delta = \frac{q_A \, q_B \, q_C}{q_A q_B + q_B q_C + q_A q_C}.$$

Consider the following partitioning procedure:

- Pick one of the sets A, B, or C with probability p_A, p_B, and p_C, respectively (note that $p_A + p_B + p_C = 1$).
- If the chosen set is A, partition it into two random sets S_B and S_C of size $\widetilde{b}|A|/(\widetilde{b}+\widetilde{c})$ and $\widetilde{c}|A|/(\widetilde{b}+\widetilde{c})$ and output the cut $L = B \cup S_B$, $R = C \cup S_C$.
- Similarly, if the chosen set is B, we partition it into two random sets S_A and S_C of size $\widetilde{a}|B|/(\widetilde{a}+\widetilde{c})$ and $\widetilde{c}|B|/(\widetilde{a}+\widetilde{c})$ and output the cut $L = C \cup S_C$, $R = A \cup S_A$.
- If the chosen set is C, we partition it into two random sets S_A and S_B of size $\widetilde{a}|C|/(\widetilde{a}+\widetilde{b})$ and $\widetilde{b}|C|/(\widetilde{a}+\widetilde{b})$ and output the cut $L = A \cup S_A$, $R = B \cup S_B$.

We first observe that each of the output sets L and R has $n/2$ vertices, i.e., (L, R) is a bisection of the graph. If for instance, the algorithm picks set A at the first step, then the set L contains $|B| + \widetilde{b}|A|/(\widetilde{b}+\widetilde{c})$ vertices. We have

$$|L| = |B| + \frac{\widetilde{b}}{\widetilde{b}+\widetilde{c}}|A| = bn + \frac{1/2 - b}{1 - b - c} \cdot an =$$

$$= bn + \frac{1/2 - b}{a} \cdot an = \frac{n}{2}.$$

The set R is the complement to L, thus, it also contains $n/2$ vertices. The cases when the algorithm picks the set B or C are identical.

We now compute the expected weight of red edges in the bisection (L, R).

Proposition 3.6. *The expected weight of uncut red edges is* $(1 - \delta)\alpha$.

Proof. Again, assume that the algorithm picks the set A at the first step. Then, the sets B and C are contained in the sets L and R, respectively. Consequently, no edges in B and C are in the cut between L and R. Every edge in A is cut with probability $2\widetilde{b}\widetilde{c}/(\widetilde{b}+\widetilde{c})^2$. Thus, the weight of red edges in the cut between L and R (denoted as $E^{red}(L,R)$) given that the algorithm picks set A equals:

$$\mathbb{E}\left[|E^{red}(L,R)| \mid \text{ algorithm picks } A \text{ at first step}\right] =$$

$$= \frac{2\widetilde{b}\widetilde{c}}{(\widetilde{b}+\widetilde{c})^2} w(A) = q_A w(A).$$

Similarly, if the algorithm picks the set B or C, the expected sizes of the cuts equal $q_B w(B)$ and $q_C w(C)$, respectively. Hence, the expected weight of the red edges between L and R (when we do not condition on the first step of the algorithm) equals

$$\mathbb{E}\left[|E^{red}(L,R)|\right] = p_A q_A w(A) + p_B q_B w(B) + p_C\, q_C w(C)$$

Observe that

$$p_A q_A = p_B q_B = p_C q_C = \frac{q_A\, q_B\, q_C}{q_A q_B + q_B q_C + q_A q_C} = \delta.$$

Then, the expected weight of red edges between L and R equals:

$$\mathbb{E}\left[|E^{red}(L,R)|\right] = \delta(w(A) + w(B) + w(C)) = \delta\alpha.$$

Here, we used that $w(A) + w(B) + w(C) = \alpha$ and we conclude that the expected weight of uncut red edges equals $(1-\delta)\alpha$. □

We now lower bound the weight of uncut blue edges.

Proposition 3.7. *The expected weight of uncut blue edges is at least $\delta\beta$.*

Proof. We separately consider edges between sets A and B, B and C, A and C. Consider an edge $(u,v) \in A \times B$. This edge is not in the cut (L,R) if both endpoints u and v belong to L or both endpoints belong to R. The former event – $\{u,v \in L\}$ – occurs if the set B is chosen in the first step of the algorithm and the set S_A contains vertex v; the latter event – $\{u,v \in R\}$ – occurs if A is chosen in the first step of the algorithm and the set S_B contains vertex u. The probability of the union of these events is[5]

$$\Pr[(u,v) \notin (L,R)] = p_B \cdot \frac{\widetilde{a}}{\widetilde{a}+\widetilde{c}} + p_A \cdot \frac{\widetilde{b}}{\widetilde{b}+\widetilde{c}} =$$

$$= p_B q_B \cdot \frac{(\widetilde{a}+\widetilde{c})}{2\widetilde{c}} + p_A q_A \cdot \frac{(\widetilde{b}+\widetilde{c})}{2\widetilde{c}}.$$

Since $p_A q_A = p_B q_B = \delta$, we have

$$\Pr[(u,v) \notin (L,R)] = \delta \cdot \frac{\widetilde{a}+\widetilde{b}+2\widetilde{c}}{2\widetilde{c}} = \delta \cdot \frac{1/2+\widetilde{c}}{2\widetilde{c}} \geq \delta.$$

The last inequality holds because $(1/2+\widetilde{c})/\widetilde{c} \geq 2$ for all $\widetilde{c} \in (0,1/2]$. The same bound holds for edges between sets B and C and sets A and C. Therefore, the expected weight of uncut blue edges is at least $\delta\beta$. □

By the above two propositions (Proposition 3.6 and Proposition 3.7) the expected total weight of uncut edges is at least:

$$\mathbb{E}\left[w(L) + w(R)\right] \geq (1-\delta)\alpha + \delta\beta = \alpha - (\alpha-\beta)\delta.$$

Note that we are in the case with $\alpha - \beta \geq 0$. Thus to establish a lower bound on the expectation, we need to show an upper bound on δ. Write

$$\delta = \frac{q_A\, q_B\, q_C}{q_A q_B + q_B q_C + q_A q_C} = \frac{1}{\frac{1}{q_A} + \frac{1}{q_B} + \frac{1}{q_C}}.$$

After plugging in the values of q_A, q_B, and q_C, we obtain the following expression for δ.

$$\delta = \frac{1}{\frac{(\widetilde{b}+\widetilde{c})^2}{2\widetilde{b}\widetilde{c}} + \frac{(\widetilde{a}+\widetilde{c})^2}{2\widetilde{a}\widetilde{c}} + \frac{(\widetilde{a}+\widetilde{b})^2}{2\widetilde{a}\widetilde{b}}} =$$

$$= \frac{1}{3 + \frac{1}{2}\left(\frac{\widetilde{b}+\widetilde{c}}{\widetilde{a}} + \frac{\widetilde{a}+\widetilde{c}}{\widetilde{b}} + \frac{\widetilde{a}+\widetilde{b}}{\widetilde{c}}\right)}$$

Observe that $a+b+c = 1$ and $\widetilde{a}+\widetilde{b}+\widetilde{c} = 1/2$. Thus, $\widetilde{b}+\widetilde{c} = 1/2-\widetilde{a}$, $\widetilde{a}+\widetilde{c} = 1/2-\widetilde{b}$, and $\widetilde{a}+\widetilde{b} = 1/2-\widetilde{c}$. Hence,

$$\delta = \frac{1}{3 + \frac{1}{2}\left(\frac{1/2-\widetilde{a}}{\widetilde{a}} + \frac{1/2-\widetilde{b}}{\widetilde{b}} + \frac{1/2-\widetilde{c}}{\widetilde{c}}\right)} =$$

$$= \frac{1}{\frac{3}{2} + \frac{1}{4}\left(\frac{1}{\widetilde{a}} + \frac{1}{\widetilde{b}} + \frac{1}{\widetilde{c}}\right)}.$$

Note that since the function $t \mapsto 1/t$ is convex for $t > 0$, we have

$$\frac{1}{2}\left(\frac{1}{\widetilde{a}} + \frac{1}{\widetilde{b}}\right) \geq \frac{2}{\widetilde{a}+\widetilde{b}} = \frac{2}{1/2-\widetilde{c}} = \frac{2}{c}$$

Therefore,

$$\delta \leq \frac{1}{\frac{3}{2} + \frac{1}{c} + \frac{1}{4\widetilde{c}}} = \frac{c(1-2c)}{1-3c^2}.$$

We conclude that the expected weight of uncut edges is at least $\alpha - (\alpha-\beta)\delta_{max}(c)$, where $\delta_{max}(c) = c(1-2c)/(1-3c^2)$. □

Proposition 3.8. *Let $\rho = 0.8776$ be the approximation factor of the* Max-Uncut Bisection *algorithm [ABG16, WDX15]. Then if $\beta \leq \alpha$, our Algorithm 2 outputs a solution of value at least $\frac{2\rho}{3}(n-1)\left((1-\delta_{max}(c))\alpha + \delta_{max}(c)\beta\right)$.*

[5] Note that $\widetilde{c} = 0$ cannot happen by definition of the sets A, B, C.

Proof. Let (L, R) be the bisection produced by the ρ-approximate Max-Uncut Bisection algorithm. This partition satisfies:

$$w(L) + w(R) \geq \rho \texttt{OPT}_{\text{Max-Uncut Bisection}}$$

Our Algorithm 2 produces a tree where at the top level the left subtree is L, the right subtree is R and both of these subtrees are then generated by Average-Linkage. Hence each edge $(i, j) \in L \times L$ (and similarly for edges in $R \times R$) contributes:

$$w_{ij}\left(\tfrac{n}{2} + \tfrac{1}{3}(\tfrac{n}{2} - 2)\right) = \tfrac{2}{3} w_{ij}(n-1)$$

to the objective. Thus the overall value of our solution is at least:

$$\tfrac{2}{3}(n-1)(w(L) + w(R)) \geq$$

$$\geq \frac{2\rho}{3}(n-1) \cdot \texttt{OPT}_{\text{Max-Uncut Bisection}}$$

If $\beta \leq \alpha$ then by Lemma 3.5 we have that $\texttt{OPT}_{\text{Max-Uncut Bisection}} \geq \alpha - (\alpha - \beta)\delta_{max}(c)$ and the proof follows by rearranging the terms. □

Lemma 3.9. *The approximation factor ξ of our Algorithm 2 is at least $\frac{4\rho}{3(2\rho+1)} \geq 0.42469$.*

Proof. First, note that if $\beta \geq \alpha$ then by Proposition 3.4 the approximation is at least 0.44. Hence it suffices to only consider the case when $\beta \leq \alpha$. Recall that by Fact 3.3, Average-Linkage outputs a solution of value $\frac{1}{3}(\alpha + \beta)(n-2)$ and by Proposition 3.2, we have $\texttt{OPT} \leq \alpha(n-2) + |C|\beta$. Hence if $\frac{1}{3}(\alpha + \beta)(n-2) \geq \xi\left(\alpha(n-2) + |C|\beta\right)$ then the desired approximation holds.

Thus we only need to consider the case when $\frac{1}{3}(\alpha + \beta)(n-2) \leq \xi\left(\alpha(n-2) + |C|\beta\right)$ or equivalently:

$$\frac{1}{3}(\alpha + \beta) \leq \xi\left(\alpha + \tfrac{|C|}{n-2}\beta\right) \iff$$

$$\iff \beta \leq \frac{3\xi - 1}{1 - 3\frac{|C|}{n-2}\xi}\alpha.$$

Let $c_1 = \frac{2\rho}{3}(1 - \delta_{max}(c))$ and $c_2 = \frac{2\rho}{3}\delta_{max}(c)$. In this case by Proposition 3.8, our Algorithm 2 gives value at least $c_1(n-1)\alpha + c_2(n-1)\beta$. Hence it suffices to show that $c_1(n-2)\alpha + c_2(n-2)\beta \geq \xi(\alpha(n-2) + |C|\beta)$. Or equivalently that:

$$\beta(\tfrac{|C|}{n-2}\xi - c_2) \leq \alpha(c_1 - \xi)$$

Using the bound on β above it suffices to show that:

$$\frac{3\xi - 1}{1 - 3\frac{|C|}{n-2}\xi}(\tfrac{|C|}{n-2}\xi - c_2) \leq c_1 - \xi.$$

After simplifying this expression the above bound holds for:

$$\xi \leq \frac{c_1 - c_2}{1 + 3\frac{cn}{n-2}c_1 - \frac{cn}{n-2} - 3c_2}.$$

Hence it suffices to find the minimum of the RHS over $c \in [0, \frac{1}{2} - \frac{1}{n}]$. Plugging in the expressions for c_1 and c_2 after simplification the RHS is equal to:

$$\frac{2}{3}\frac{\rho(1-c)}{2c^2\rho + (1 - 3c^2)}$$

Differentiating over c one can show that the minimum of this expression is attained for $c = \frac{1}{2} - \frac{1}{n}$. Indeed, the numerator of the derivative is a quadratic function with negative leading coefficient whose roots are $1 \pm \frac{\sqrt{t(t+1)}}{t}$ for $t = 2\rho - 3$. The left root is approximately 0.545 and hence the derivative is negative on $[0, \frac{1}{2} - \frac{1}{n}]$. The value at the minimum $c = \frac{1}{2} - \frac{1}{n}$ is thus equal[6] to ($\rho = 0.8776$):

$$\frac{4\rho}{3(2\rho + 1)} \geq 0.42469$$

□

4 Hardness of Approximation

In this section, we prove that maximizing the Moseley-Wang HC objective (*) is APX-hard:

Theorem 4.1. *Under the* Small Set Expansion *(*SSE*) hypothesis, there exists $\varepsilon > 0$, such that it is NP-hard to approximate the Moseley-Wang HC objective function (*) within a factor $(1 - \varepsilon)$.*

Initially introduced by Raghavendra and Steurer [RS10], SSE has been used to prove improved hardness results for optimization problems including Balanced Separator and Minimum Linear Arrangement [RST12].

Given a d-regular, unweighted graph $G = (V, E)$ and $S \subseteq V$, let $\mu(S) := |S|/|V|$ and $\Phi(S) := |E(S, V \backslash S)|/d|S|$. Raghavendra et al. [RST12] prove the following strong hardness result. (While it is not explicitly stated that the result holds for regular graphs, it can be checked that their reduction produces a regular graph [Tul19].)

Theorem 4.2 (Theorem 3.6 of [RST12])**.** *Assuming the* SSE*, for any $q \in \mathbb{N}$ and $\varepsilon, \gamma > 0$, given a regular graph $G = (V, E)$, it is NP-hard to distinguish the following two cases.*

[6] Observe that our analysis based on Max-Uncut Bisection is almost tight since even if we were given exact access to the optimum (i.e., $\rho = 1$), the approximation ratio for HC would only slightly increase to $\frac{4}{9} = 0.444$.

- *YES: There exist q disjoint sets $S_1, \ldots, S_q \subseteq V$ such that for all $\ell \in [q]$,*

  $$\mu(S_\ell) = 1/q \qquad \textit{and} \qquad \Phi(S_\ell) \leq \varepsilon + o(\varepsilon).$$

- *NO: For all sets $S \subseteq V$,*

  $$\Phi(S) \geq \phi_{1-\varepsilon/2}(\mu(S)) - \gamma/\mu(S)$$

  where $\phi_{1-\varepsilon/2}(\mu(S))$ is the expansion of the sets of volume $\mu(S)$ in the infinite Gaussian graph with correlation $1 - \varepsilon/2$.

Proof of Theorem 4.1. Let us consider the instance of Hierarchical Clustering defined by the same graph where each pair has weight 1 if there is an edge, and 0 otherwise. Then $W = |E|$ is the total weight.

- YES: The fraction of edges crossing between different S_i's is at most $\varepsilon + o(\varepsilon)$, and all edges inside some S_i are multiplied by at least $n(1 - 1/q)$ in the objective function. So the objective function for Hierarchical Clustering is at least

  $$(1 - \varepsilon - o(\varepsilon))W \cdot (1 - \tfrac{1}{q})n \geq nW(1 - \tfrac{1}{q} - \varepsilon - o(\varepsilon)).$$

- NO: Consider an arbitrary binary tree $\mathcal{T}$ that maximizes the Moseley-Wang objective function (*). For a tree node $a \in \mathcal{T}$, let $\mathcal{T}_a$ be the subtree of $\mathcal{T}$ rooted at a, and $V_a \subseteq V$ be the set of graph vertices corresponding the leaves of T_a.

  Let $b \in \mathcal{T}$ be a highest node such that $n/3 \leq |V_b| \leq 2n/3$ (such a node always exists in a binary tree). By Theorem 4.2, we have

  $$\Phi(V_b) \geq \phi_{1-\varepsilon/2}(\mu(V_b)) - \gamma/\mu(V_b) \geq C\sqrt{\varepsilon}$$

  for some absolute constant C. Here we use the fact that

  $$\phi_{1-\varepsilon/2}(\mu(V_b)) \geq \Omega(\sqrt{\varepsilon}) \text{ for } \mu(V_b) \in [1/3, 2/3]$$

  and take γ small enough depending on ε.

  So the total fraction of edges in $E(V_b, V \setminus V_b)$ is at least

  $$\mu(V_b) \cdot \Phi(V_b) \geq \tfrac{C\sqrt{\varepsilon}}{3}.$$

  Note that edges in $E(T_b, V \setminus T_b)$ will be multiplied by at most $n/3$ in the objective function. (Let a be the parent of b. Then $|V_a| > 2n/3$ by the choice of b and for any edge crossing V_b, then the least common ancestors of the two endpoints will be a or one of its ancestors.) Therefore, the objective function is at most

  $$nW - \tfrac{C\sqrt{\varepsilon}}{3} W \cdot \tfrac{2n}{3} = nW(1 - \tfrac{2C\sqrt{\varepsilon}}{9}).$$

Therefore, the value is at least $nW(1 - 1/q - \varepsilon - o(\varepsilon))$ in the YES case and $nW(1 - (2C\sqrt{\varepsilon}/9))$ in the NO case. By taking $\varepsilon > 0$ sufficiently small and q arbitrarily large, there is a constant gap between the YES case value and the NO case value. □

5 Conclusion

In this paper, we presented a 0.4246 approximation algorithm for the hierarchical clustering problem with pairwise similarities under the Moseley-Wang objective (*), which is the complement to Dasgupta's objective. Our algorithm uses Max-Uncut Bisection as a black box and improves upon previous state-of-the-art approximation algorithms that were more complicated and only guaranteed 0.3363 of the optimum value. In terms of hardness of approximation, under the Small Set Expansion hypothesis, we prove that even for unweighted graphs, there exists $\varepsilon > 0$, such that it is NP-hard to approximate the objective function (*) within a factor of $(1 - \varepsilon)$.

6 Acknowledgements

Vaggos Chatziafratis was supported by an Onassis Foundation Scholarship and part of this work was done while interning at Google Research NY. Grigory Yaroslavtsev was supported by NSF grant 1657477. We would like to thank the anonymous reviewers for their feedback.

References

[ABG16] Per Austrin, Siavosh Benabbas, and Konstantinos Georgiou. Better balance by being biased: A 0.8776-approximation for max bisection. *ACM Transactions on Algorithms (TALG)*, 13(1):2, 2016.

[ABV17] Pranjal Awasthi, Maria Florina Balcan, and Konstantin Voevodski. Local algorithms for interactive clustering. *The Journal of Machine Learning Research*, 18(1):75–109, 2017.

[ARV08] Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Geometry, flows, and graph-partitioning algorithms. *Commun. ACM*, 51(10):96–105, 2008.

[BB08] Maria-Florina Balcan and Avrim Blum. Clustering with interactive feedback. In *International Conference on Algorithmic Learning Theory*, pages 316–328. Springer, 2008.

[BBD^{+}17] Mohammadhossein Bateni, Soheil Behnezhad, Mahsa Derakhshan, MohammadTaghi Hajiaghayi, Raimondas Kiveris, Silvio Lattanzi, and Vahab Mirrokni. Affinity clustering: Hierarchical clustering at scale. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6864–6874. Curran Associates, Inc., 2017.

[CAKMTM19] Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and Claire Mathieu. Hierarchical clustering: Objective functions and algorithms. *Journal of the ACM (JACM)*, 66(4):26, 2019.

[CC17] Moses Charikar and Vaggos Chatziafratis. Approximate hierarchical clustering via sparsest cut and spreading metrics. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 841–854. Society for Industrial and Applied Mathematics, 2017.

[CCFM04] Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. Incremental clustering and dynamic information retrieval. *SIAM Journal on Computing*, 33(6):1417–1440, 2004.

[CCN19] Moses Charikar, Vaggos Chatziafratis, and Rad Niazadeh. Hierarchical clustering better than average-linkage. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2291–2304. SIAM, 2019.

[CCNY18] Moses Charikar, Vaggos Chatziafratis, Rad Niazadeh, and Grigory Yaroslavtsev. Hierarchical clustering for euclidean data. *arXiv preprint arXiv:1812.10582*, 2018.

[CNC18] Vaggos Chatziafratis, Rad Niazadeh, and Moses Charikar. Hierarchical clustering with structural constraints. In *International Conference on Machine Learning*, pages 773–782, 2018.

[Das02] Sanjoy Dasgupta. Performance guarantees for hierarchical clustering. In *International Conference on Computational Learning Theory*, pages 351–363. Springer, 2002.

[Das16] Sanjoy Dasgupta. A cost function for similarity-based hierarchical clustering. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 118–127. ACM, 2016.

[DBE^{+}15] Ibai Diez, Paolo Bonifazi, Iñaki Escudero, Beatriz Mateos, Miguel A Muñoz, Sebastiano Stramaglia, and Jesus M Cortes. A novel brain partition highlights the modular skeleton shared by structure and function. *Scientific reports*, 5:10532, 2015.

[EDSN11] Brian Eriksson, Gautam Dasarathy, Aarti Singh, and Rob Nowak. Active clustering: Robust and efficient hierarchical clustering using adaptively selected similarities. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 260–268, 2011.

[ESBB98] Michael B Eisen, Paul T Spellman, Patrick O Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, 1998.

[EZK18] Ehsan Emamjomeh-Zadeh and David Kempe. Adaptive hierarchical clustering using ordinal queries. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 415–429. Society for Industrial and Applied Mathematics, 2018.

[GR69] John C Gower and Gavin JS Ross. Minimum spanning trees and single linkage cluster analysis. *Journal of* *the Royal Statistical Society: Series C (Applied Statistics)*, 18(1):54–64, 1969.

[Jai10] Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[JMF99] Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[JS68] N Jardine and R Sibson. A model for taxonomy. *Mathematical Biosciences*, 2(3-4):465–482, 1968.

[LNRW10] Guolong Lin, Chandrashekhar Nagarajan, Rajmohan Rajaraman, and David P Williamson. A general approach for incremental approximation and hierarchical clustering. *SIAM Journal on Computing*, 39(8):3633–3669, 2010.

[LRU14] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge university press, 2014.

[MMO08] Charles F Mann, David W Matula, and Eli V Olinick. The use of sparsest cuts to reveal the hierarchical community structure of social networks. *Social Networks*, 30(3):223–234, 2008.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[MW17] Benjamin Moseley and Joshua Wang. Approximation bounds for hierarchical clustering: Average linkage, bisecting k-means, and local search. In *Advances in Neural Information Processing Systems*, pages 3097–3106, 2017.

[Pla06] C Greg Plaxton. Approximation algorithms for hierarchical location problems. *Journal of Computer and System Sciences*, 72(3):425–443, 2006.

[RP16] Aurko Roy and Sebastian Pokutta. Hierarchical clustering via spreading metrics. In *Advances in Neural Information Processing Systems*, pages 2316–2324, 2016.

[RS10] Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 755–764. ACM, 2010.

[RST12] Prasad Raghavendra, David Steurer, and Madhur Tulsiani. Reductions between expansion problems. In *2012 IEEE 27th Conference on Computational Complexity*, pages 64–73. IEEE, 2012.

[SKK$^+$00] Michael Steinbach, George Karypis, Vipin Kumar, et al. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston, 2000.

[SS62] Peter HA Sneath and Robert R Sokal. Numerical taxonomy. *Nature*, 193(4818):855–860, 1962.

[SS13] Peter Sanders and Christian Schulz. Kahip v0.53 - karlsruhe high quality partitioning - user guide. *CoRR*, abs/1311.1714, 2013.

[TLM10] Michele Tumminello, Fabrizio Lillo, and Rosario N Mantegna. Correlation, hierarchies, and networks in financial markets. *Journal of Economic Behavior & Organization*, 75(1):40–58, 2010.

[Tul19] Madhur Tulsiani, 2019. Personal Communication.

[VD16] Sharad Vikram and Sanjoy Dasgupta. Interactive bayesian hierarchical clustering. In *International Conference on Machine Learning*, pages 2081–2090, 2016.

[WC00] Kiri Wagstaff and Claire Cardie. Clustering with instance-level constraints. *AAAI/IAAI*, 1097:577–584, 2000.

[WCR$^+$01] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584, 2001.

[WDX15] Chenchen Wu, Donglei Du, and Dachuan Xu. An improved semidefinite programming hierarchies rounding approximation algorithm for maximum graph bisection problems. *Journal of Combinatorial Optimization*, 29(1):53–66, 2015.

[WJ63] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[YV17] Grigory Yaroslavtsev and Adithya Vadapalli. Massively parallel algorithms and hardness for single-linkage clustering under *ell_p*-distances. *arXiv preprint arXiv:1710.01431*, 2017.