Fast Combinatorial Algorithms for Min Max Correlation Clustering

Sami Davies [*1] **Benjamin Moseley** [*2] **Heather Newman** [*3]

Abstract

We introduce fast algorithms for correlation clustering with respect to the Min Max objective that provide constant factor approximations on complete graphs. Our algorithms are the *first* purely combinatorial approximation algorithms for this problem. We construct a novel semi-metric on the set of vertices, which we call the correlation metric, that indicates to our clustering algorithms whether pairs of nodes should be in the same cluster. The paper demonstrates empirically that, compared to prior work, our algorithms sacrifice little in the objective quality to obtain significantly better run-time. Moreover, our algorithms scale to larger networks that are effectively intractable for known algorithms.

1. Introduction

In correlation clustering, a graph $G = (V, E)$ is given as input, where each edge is labeled either positive $(+)$ or negative $(-)$. If two vertices are connected by a positive edge, this indicates they are similar and want to being clustered together. Alternatively, a negative edge indicates vertices that are dissimilar and want to be in different clusters. We make the common assumption that G is complete; that is, there is a labelled edge between each pair of vertices.

An edge is in *disagreement* with respect to a clustering if it is a negative edge and connects two vertices contained inside the same cluster, or if it is a positive edge and connects two vertices in different clusters. Note that among three or more vertices, the positive and negative labels may be such that *any* clustering has some edges in disagreement, e.g. three vertices where the three labels between them are two positives and one negative.

[*]Equal contribution [1]Department of Computer Science, Northwestern University, Evanston IL, USA [2]Tepper School of Business, Carnegie Mellon University, Pittsburgh PA, USA [3]Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh PA, USA. Correspondence to: Heather Newman <hanewman@andrew.cmu.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

The goal in correlation clustering is to find a partition of the vertices into clusters (the number of which is *not* specified) that minimizes an objective capturing the edges' disagreements. The most widely considered objective is to minimize the ℓ_p-norm of the disagreements over the vertices. Here, each vertex u has some number, $y(u)$, of disagreements adjacent to it and the goal is to minimize $\sqrt[p]{\sum_{u \in V} y(u)^p}$. The most well-studied objective is when $p = 1$, which minimizes the total number of disagreements; the Pivot algorithm is a popular, combinatorial algorithm in this setting (Ailon et al., 2008). The case where $p = \infty$ minimizes the maximum number of disagreements at any vertex, which captures a nice notion of fairness in the clustering.

Our results focus on the ℓ_∞-norm objective, which we will refer to as the *Min Max* objective[1]. Min Max correlation clustering was first motivated by applications in community detection that are antagonist free, i.e., there are no members that are largely inconsistent in their community (Puleo & Milenkovic, 2016). Such problems arise in recommender systems, bioinformatics, and social sciences (Cheng & Church, 2000; Kriegel et al., 2009; Symeonidis et al., 2008; Puleo & Milenkovic, 2016).

Milenkovic and Puleo (2016) initiated the study of Min Max correlation clustering, as well as other ℓ_p-norms. For all $p \geq 1$, they give a 48-approximation. Charikar, Gupta, and Schwartz (2017) improved this to a 7-approximation, and Kalhan, Makarychev, and Zhou (2019) further improved this to the best known approximation[2] of 5. Milenkovic and Puleo (2016) observed that Pivot fails for the ℓ_∞ norm, even though it gives a 3-approximation in expectation for the ℓ_1 norm.

Previous algorithms on Min Max correlation clustering rounded SDP or LP solutions (Puleo & Milenkovic, 2016; Charikar et al., 2017; Ahmadi et al., 2019; Kalhan et al., 2019). The relaxations are large, requiring at least $|V|^3$ constraints and $|V|^2$ variables. Thus solving them is not practical on even modest sized graphs (e.g. 300-500 vertices). The question looms: does there exist a fast algorithm for Min Max correlation clustering with strong theoretical guarantees? Moreover, an intriguing direction both towards

[1]Other names for this are the minimax or ℓ_∞ objective.

[2]Note all of these results are still only for complete graphs.

this question and in its own right is to develop combinatorial algorithms. A key challenge is that it is not clear how to compare to the optimal solution without the LP or SDP, as there are no non-obvious lower bounds known.

1.1. Our contributions

The paper provides fast algorithms for Min Max correlation clustering. Our algorithms are the first combinatorial algorithms for the problem with theoretical guarantees. Our key technical insight that enables these algorithms is the introduction of a semi-metric on the set of vertices, which we call the *correlation metric*. The correlation metric can be used to set variables in the problem's linear program so that the resulting solution is feasible and provably close to the optimal number of disagreements for the Min Max objective. Thus, we use the LP to compare to the optimal like prior work, but we do not need to solve it since we only use it in the analysis. Our correlation metric gives new insights into both correlation clustering and the linear program.

Let ω denote the constant for the $O(n^\omega)$ run-time of matrix multiplication on an $n \times n$ matrix, where $n = |V|$.

Theorem 1.1. *There is an algorithm that obtains a 40-approximation for Min Max correlation clustering on complete graphs in time $O(n^\omega)$.*

The run-time can be substantially improved when the positive edges of the graph form a sparse graph, which is commonly the case in practice.

Corollary 1.2. *Suppose all vertices in V have $+$ degree at most Δ, and for each vertex we are given a list of its positive neighbors. There is an algorithm that obtains a 40-approximation for Min Max correlation clustering on complete graphs in time $O(n\Delta^2 \log n)$.*

Thus, we have a near-linear time algorithm for sparse graphs. This substantially improves on prior work, which (to the best of our knowledge) has run-time no better than $O(n^{2\omega})$ even on sparse graphs (see Appendix A).

Next, we show that we can approximate the correlation metric to improve the run-time, even when the graph is dense, though at some loss in the approximation factor.

Theorem 1.3. *Fix $\varepsilon > 0$ sufficiently small. For some constant $c(\varepsilon)$ depending only on ε, there is a randomized algorithm that obtains a $c(\varepsilon)$-approximation for Min Max correlation clustering on complete graphs with probability $1 - O(1/n)$ in time $O(n^2 \log n/\varepsilon^2)$.*

For the definition of $c(\varepsilon)$, see Appendix E.5. Finally, we show that our theory is predictive of practice (see Section 6). Our algorithm performs similarly in terms of objective value to the algorithm of Kalhan, Makarychev, and Zhou (2019), while improving the runtime so substantially that it can feasibly scale to graphs with about 10,000 vertices[3]. Moreover, the clusters found by our algorithm are often meaningful in that they partially discover "ground truth" clusters in real-world and synthetic instances.

1.2. Related work

In correlation clustering, the most commonly studied objective is minimizing the total number of edges in disagreement, which is equivalent to minimizing the ℓ_1 norm of the disagreement vector $y \in \mathbb{R}^{|E|}_{\geq 0}$. Note that minimizing the ℓ_1 norm can be studied when the edges are weighted. The model was introduced by Bansal, Blum, and Chawla (2004) and is motivated by many applications such as image segmentation, natural language processing, clustering gene expression patterns, and location area planning (Wirth, 2017; McCallum & Wellner, 2004; Ben-Dor & Yakhini, 1999; Demaine & Immorlica, 2003). The celebrated Pivot algorithm of Ailon, Charikar, and Newman (2008) obtains a 3-approximation in expectation. Until recently, the best approximation algorithm for ℓ_1 correlation clustering on complete, unweighted graphs was a $(2.06 + \epsilon)$-approximation due to Chawla, Makarychev, Schramm, and Yaroslavtsev (2015), but the threshold of 2 was broken (they achieved a $(1.994 + \epsilon)$-approximation) in the work of Cohen-Addad, Lee, and Newman (2022), who successfully took advantage of $O(1/\epsilon^2)$ rounds of the Sherali-Adams hierarchy.

Min Max correlation clustering was introduced by Puleo and Milenkovic (2016). They show correlation clustering for the Min Max objective is NP-hard even on complete graphs (see Appendix C in their paper), and algorithmically, they obtain 48-approximation algorithms for complete graphs and complete bipartite graphs. Shortly after, Charikar, Gupta, and Schwartz (2017) gave a 7-approximation for minimizing the ℓ_p norm on the same graphs. For the Min Max objective, i.e. $p = \infty$, they obtain a $O(\sqrt{n})$-approximation algorithm on general, weighted graphs. Kalhan, Makarychev, and Zhou (2019) further generalize the known results for correlation clustering with local objectives by showing approximation algorithms that minimize the ℓ_p norm on general, weighted graphs. This work also shows a 5-approximation for the ℓ_p (including $p = \infty$) objective on complete graphs and on complete bipartite graphs.

In previous works that study any ℓ_p norm objective for correlation clustering, the run-time relies on solving an LP with at least $\Omega(n^2)$ many variables and $\Omega(n^3)$ constraints (Puleo & Milenkovic, 2016; Charikar et al., 2017; Kalhan et al., 2019). We see no clear way to use the structure of the LP to guarantee solving it would take time less than $O(n^{2\omega})$, even on sparse graphs (see Appendix A).

[3]For comparison, the algorithm by Kalhan, Makarychev, and Zhou takes more than 10 minutes on graphs with ≈ 200 vertices.

Several other objectives that are local or capture some notion of fairness have also been studied (Ahmadian et al., 2020; Bateni et al., 2022; Ahmadi et al., 2020; Friggstad & Mousavi, 2021; Jafarov et al., 2021; Ahmadi et al., 2019). For instance, Ahmadi, Khuller, and Saha (2019) seek to find a clustering where for every point, the average distance to the points in its own cluster is no more than the average distance to those in another cluster. Their algorithm rounds the solution to an SDP, which was an idea based on work on min-max k balanced partitioning. Further, Jafarov et al. (2021) studied the ℓ_p objective, but additionally make assumptions on the weights based on whether edges are similar.

2. Notation and Preliminaries

Let $G = (V, E)$ be the complete graph on n vertices with self-loops[4], where each edge has a positive $(+)$ or negative $(-)$ label. Let E^+ be the set of positive edges of G and let E^- be the set of negative edges. For vertex $u \in V$, define $N_u^- = \{v \in V \mid (u, v) \in E^-\}$ to be the *negative neighborhood* of u, and $N_u^+ = \{v \in V \mid (u, v) \in E^+\}$ to be the *positive neighborhood* of u.

We say an edge $e = (u, v) \in E$ is in *disagreement* according to a clustering if $e \in E^+$ and u and v are in different clusters or if $e \in E^-$ and u and v are in the same cluster. For ease in calculations, we assume that every vertex has a positive self-loop, i.e. $(v, v) \in E^+$ for all $v \in V$. Note that doing so will not change the set of disagreeing edges. The following holds for G:

Fact 1. For any $u \in V$, $n = |N_u^+| + |N_u^-|$.

Fact 2. Let $u, v \in V$, possibly with $u = v$. Then

$$n = |N_u^+ \cap N_v^+| + |N_u^- \cap N_v^-| + |N_u^+ \cap N_v^-| + |N_u^- \cap N_v^+|.$$

Proof of Fact 2. Observe $N_u^+ = (N_u^+ \cap N_v^+) \cup (N_u^+ \cap N_v^-)$ and $N_u^- = (N_u^- \cap N_v^+) \cup (N_u^- \cap N_v^-)$. Substituting into Fact 1 gives Fact 2. □

Let $C = (C_1, \ldots, C_k)$ denote a partition of V into k clusters, which we will call a clustering. (Note that in correlation clustering, there is no limit on the number of clusters; k is *not* an input parameter.) We say $C(u) = C_i$ if u is in cluster C_i. Let $\overline{C(u)}$ be the set of all vertices in a different cluster from u. For a given clustering C, let $y_C \in \mathbb{Z}_+^n$ be the *disagreement vector* of C, indexed by V. For $u \in V$, the coordinate $y_C(u)$ is the number of edges incident to u that are disagreements in C. We drop the subscript when C is clear from context.

The correlation metric. We introduce a novel semi-metric based on the positive and negative neighborhoods of nodes.

[4]One could avoid the self-loops by slightly changing the definition of neighborhood intersections.

We will prove that the correlation metric, d, satisfies the triangle inequality, cementing that it is a semi-metric.

Definition 2.1. For all $u, v \in V$, the distance between u and v with respect to the *correlation metric* is

$$\begin{aligned} d_{uv} &= 1 - \frac{|N_u^+ \cap N_v^+|}{n - |N_u^- \cap N_v^-|} \\ &= 1 - \frac{|N_u^+ \cap N_v^+|}{|N_u^+ \cap N_v^+| + |N_u^+ \cap N_v^-| + |N_u^- \cap N_v^+|} \qquad (1) \end{aligned}$$

where (1) follows from Fact 2. Observe that d_{uv} is well-defined as u, v have positive self-loops, so $0 \le d_{uv} \le 1$. Throughout the paper define $\widehat{d}_{uv} = d_{uv}$ if $(u, v) \in E^+$ and $\widehat{d}_{uv} = 1 - d_{uv}$ if $(u, v) \in E^-$.

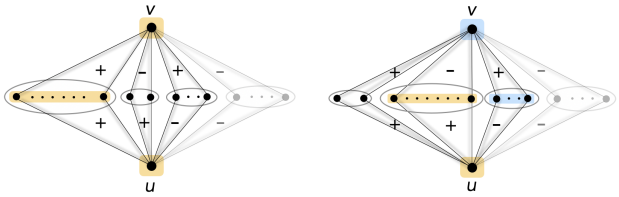


Figure 1. **Left:** $|N_u^+ \cap N_v^+|$ is much larger than $|(N_u^- \cap N_v^+) \cup (N_u^+ \cap N_v^-)|$. If v and u are in the same cluster, edges (u, w) and (w, v) need not form disagreements for all $w \in N_u^+ \cap N_v^+$. **Right:** $|(N_u^- \cap N_v^+) \cup (N_u^+ \cap N_v^-)|$ is much larger than $|N_u^+ \cap N_v^+|$. If v and u are in different clusters, edges (u, w) and (w, v) need not form disagreements for all $w \in (N_u^- \cap N_v^+) \cup (N_u^+ \cap N_v^-)$.

We give some intuition on the definition of d_{uv}. See Figure 1. Fix three distinct vertices $u, v, w \in V$. We will examine how the clustering of u and v affects the disagreements incident to them. First, note that if $w \in N_u^- \cap N_v^-$, as long as w is assigned to a different cluster than both u and v, the edges $(u, w), (v, w)$ are not in disagreement, regardless of whether or not u or v are in the same cluster. I.e., w should not impact whether u and v are in the same cluster. If we do not subtract off $|N_u^- \cap N_v^-|$ in the normalization, we will not correctly identify perfect clusterings, whereas here, $d_{uv} = \mathbf{1}_{\{C(u) \neq C(v)\}}$ when C is a perfect clustering.

On the other hand, if $w \notin N_u^- \cap N_v^-$, then whether or not u and v are in the same cluster directly impacts whether either of (u, w) or (v, w) are in disagreement. By the proof of Fact 2, the edges that are affected are (u, w) and (v, w) for $w \in (N_u^+ \cap N_v^+) \cup (N_u^- \cap N_v^+) \cup (N_u^+ \cap N_v^-)$. If u and v are in different clusters, then for $w \in (N_u^- \cap N_v^+) \cup (N_u^+ \cap N_v^-)$, both (u, w) and (v, w) may still be in agreement. However, if $w \in N_u^+ \cap N_v^+$, then at least one of (u, w) and (v, w) are in disagreement, so the number of disagreements incident to u and v (in total) is at least $|N_u^+ \cap N_v^+|$. An analogous point can be seen when u and v are in the same cluster, but now there are at least $|(N_u^- \cap N_v^+) \cup (N_u^+ \cap N_v^-)|$ disagreements incident to u and v. Overall, if $|N_u^+ \cap N_v^+|$ is large relative to $|(N_u^- \cap N_v^+) \cup (N_u^+ \cap N_v^-)|$ then (roughly

speaking) there are fewer disagreements incident to u and v if they are assigned to the same cluster than if they were assigned to different clusters. The larger $|N_u^+ \cap N_v^+|$ is relative to $|(N_u^- \cap N_v^+) \cup (N_u^+ \cap N_v^-)|$, the closer u and v are with respect to the correlation metric, by (1).

3. Technical Overviews

3.1. LP relaxation and KMZ algorithm

We consider the standard LP relaxation for the problem. It is not hard to see that the LP's constraints induce a semi-metric space on the vertices. Intuitively, the LP semi-metric guides the solution, where vertices close together are more likely to end up in the same cluster.

The LP consists of variables x_{uv} and y_u. In an integral solution, $x_{uv} = 1$ indicates u and v are in different clusters and $x_{uv} = 0$ indicates they are in the same cluster. The disagreement vector is y.

LP 3.1
$$\begin{aligned} \min \max_{u \in V} y_u & \\ \text{s.t. } y_u = \sum_{v \in N_u^+} x_{uv} + \sum_{v \in N_u^-} (1 - x_{uv}) & \quad \forall u \in V \\ x_{uv} \le x_{uw} + x_{vw} & \quad \forall u, v, w \in V \\ 0 \le x_{uv} \le 1 & \quad \forall u, v \in V. \end{aligned}$$

We will refer to the algorithm by Kalhan, Makarychev, and Zhou as the KMZ algorithm. The *KMZ algorithm* has two phases: first it solves LP 3.1 above, and then it rounds the LP solution in the *rounding algorithm* (Algorithm A). We justify in Appendix A that the run-time of the rounding algorithm is $O(n^2)$, so the run-time of the KMZ algorithm is dominated by the time it takes to solve the LP, which to the best of our knowledge has run-time no better than $O(n^{2\omega})$.

3.2. Combinatorial algorithms

We begin with an outline for the proof of Theorem 1.1; details can be found in Section 4. Our first algorithm consists of two steps. First, we compute the correlation metric d_{uv} for all $u, v \in V$, which produces a hand-crafted solution to LP 3.1 by setting $x_{uv} = d_{uv}$. Then, we use the d_{uv} values as input to the rounding algorithm (Algorithm A). Our analysis to prove that this procedure gives a constant factor approximation has three key steps:

1. The correlation metric (perhaps surprisingly) satisfies the triangle inequality (see Appendix C), i.e. $x = d$ is feasible for LP 3.1.
2. The objective to LP 3.1 from setting $x = d$ (fractional cost) is at most a factor 8 more than an optimal integral solution (Section 4.1). This is the heart of Theorem 1.1, and is tricky due to asymmetries in the fractional cost that force us to use non-local charging arguments.
3. The rounding algorithm (Algorithm A) of Kalhan, Makarychev, and Zhou (2019) can be used to round any feasible solution to LP 3.1 to an integer solution, while losing a factor of at most 5 in the objective.

Our run-time is dominated by the time to compute d_{uv} for all $u, v \in V$, which can be done in time $O(n^\omega)$. For run-time proofs, see Section 4.2 and Appendix A.

For sparse graphs, instead of computing the correlation metric for all pairs $u, v \in V$, we need only compute d_{uv} when $d_{uv} < 1$ (there are at most $n\Delta^2$ such pairs). Other pairs are implicitly distance 1. Then we can again use the rounding algorithm (Algorithm A). Given a feasible solution to LP 3.1, the rounding algorithm runs in time $O(n\Delta^2 \log n)$ for sparse graphs. See Appendix D for the complete proof for Corollary 1.2.

Lastly, in Section 5 we show that we can estimate the d_{uv} via sampling, giving an algorithm that trades the guarantee on the (still constant) approximation factor for a faster run-time, leading to Theorem 1.3. To compute the estimates, we sample $O(\log n)$ vertices from the positive neighborhood of each vertex. Then, we use the samples for u and v to obtain estimates of the various terms in (1). However, it is not clear a priori that sampling will work. First, not every term in (1) will be well-concentrated, so we will need to exploit the special structure of d_{uv}. Second, the fractional cost of the initial estimates will not be controlled. We will show a post-processing phase, in which we push some estimates down and others up, takes care of the issue. We use the post-processed estimates as input to the rounding algorithm. The proofs for Section 5 are in Appendix E.

4. Proof of Theorem 1.1

This section is focused on proving Theorem 1.1. We refer to $\max_{u \in V}\{\sum_{v \in V} \hat{d}_{uv}\}$ as the *fractional cost*, which is the objective value of the LP variables we set. The fractional cost is bounded in Section 4.1, and any missing proofs are in Appendix B. Appendix C establishes that the correlation metric is a semi-metric, and therefore is feasible for LP 3.1. We derive Theorem 1.1 in Section 4.2 and prove Corollary 1.2 in Appendix D.

4.1. Bounding the fractional cost

Throughout this section, y_C denotes the (integral) disagreement vector of the clustering C; we drop the subscript C when it is clear from context. The following identity is helpful for bounding the cost.

Proposition 4.1. *Let C be a clustering. For any $u \in V$, let $v \in N_u^+ \cap C(u)$. Then $|N_u^+ \cap N_v^-| + |N_u^- \cap N_v^+| \le y_C(u) + y_C(v)$.*

Our main technical result is Lemma 4.2. Recall $\widehat{d}_{uv} = d_{uv}$

if $(u,v) \in E^+$ and $\widehat{d}_{uv} = 1 - d_{uv}$ if $(u,v) \in E^-$.

Lemma 4.2. *Let y be the disagreement vector for an optimal (integral) solution. For $\mathsf{OPT} = \max_{z \in V} y(z)$ and for $u \in V$, we have $\sum_{v \in V} \widehat{d}_{uv} \le 8 \cdot \mathsf{OPT}$.*

Proof of Lemma 4.2. Expanding the summation, we have:

$$\sum_{v \in V} \widehat{d}_{uv} = \underbrace{\sum_{v \in N_u^+} \frac{|N_u^+ \cap N_v^-| + |N_u^- \cap N_v^+|}{n - |N_u^- \cap N_v^-|}}_{S^+} + \underbrace{\sum_{v \in N_u^-} \frac{|N_u^+ \cap N_v^+|}{n - |N_u^- \cap N_v^-|}}_{S^-}$$

We use Fact 2 to rewrite the first summation. To prove the lemma, it suffices to prove the following two claims.

Claim 1. $S^+ \le 3 \cdot \mathsf{OPT}$.

Claim 2. $S^- \le 5 \cdot \mathsf{OPT}$. □

The proof of Claim 1 is much cleaner than that of Claim 2, and therefore is deferred to Appendix B.

Proof of Claim 2. Let C denote an optimal clustering. Throughout, let $a_{uv} = n - |N_u^- \cap N_v^-|$. We have

$$\begin{aligned} S^- &= \sum_{v \in N_u^- \cap C(u)} (1 - d_{uv}) + \sum_{v \in N_u^- \cap \overline{C(u)}} \frac{|N_u^+ \cap N_v^+|}{a_{uv}} \\ &\le \sum_{v \in N_u^- \cap C(u)} 1 + \sum_{v \in N_u^- \cap \overline{C(u)}} \frac{|N_u^+ \cap N_v^+|}{n - |N_u^- \cap N_v^-|} \\ &\le y(u) + \underbrace{\sum_{v \in N_u^- \cap \overline{C(u)}} \frac{|N_u^+ \cap N_v^+|}{n - |N_u^- \cap N_v^-|}}_{S}, \end{aligned}$$

where the last inequality follows from the fact that if $(u,v) \in E^-$ and u, v are in the same cluster, then (u,v) is a disagreement incident to u.

It remains to bound the last summation, call it S. While we might expect the argument to mimic the bounding of the sum $\sum_{v \in N_u^+ \cap C(u)} d_{uv}$ in the proof of Claim 1, i.e., via an averaging argument, this will not work. Overall, for $v \in N_u^+$ we can compare the sum $\sum_{v \in N_u^+} d_{uv}$ only to disagreements incident to u and to v, but for $v \in N_u^- \cap \overline{C(u)}$ we will have to use a non-local charging argument, that is, charge the cost of S to disagreements incident to vertices besides u and v. Intuitively, this is because putting u and v in different clusters means (u,v) is not in disagreement according to the clustering, but necessarily any vertex $w \in N_u^+ \cap N_v^+$ is incident to an edge in disagreement. Further, this disagreement can be charged to *another* vertex—a carefully chosen $v^*(w) \in C(w)$.

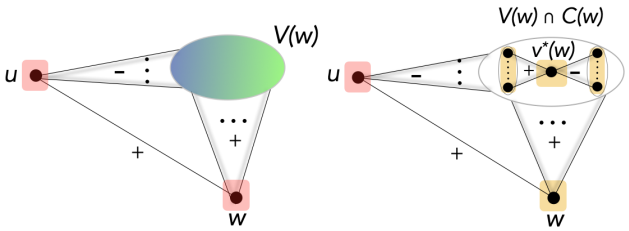


Figure 2. **Left:** A representation of bounding the sum S_1. Here, w is in the same cluster as u (red), while the vertices in $V(w)$ are in various other clusters (blue/green). **Right:** A representation of bounding the sum $S_2(w)$, specifically the right-hand side in line (5). Yellow vertices are in the same cluster. The vertex $v^*(w)$ is carefully chosen as in the text.

We begin by "flipping" the sum S. In particular, we would like to view the sum as being taken over elements in $N_u^+ \cap N_v^+$, subject to scaling by $n - |N_u^- \cap N_v^-|$.

$$S = \sum_{v \in N_u^- \cap \overline{C(u)}} \frac{|N_u^+ \cap N_v^+|}{a_{uv}} = \sum_{w \in N_u^+} \sum_{\substack{v:\ v \in N_w^+, \\ v \in N_u^- \cap \overline{C(u)}}} \frac{1}{a_{uv}}.$$

For $w \in N_u^+$, let $V(w) := \{v \in V | v \in N_u^- \cap N_w^+ \cap \overline{C(u)}\}$.

Note that $V(u) = \emptyset$, since $N_u^- \cap N_u^+ = \emptyset$, so the outer sum need not include the self-loop from u. Thus we have:

$$S = \underbrace{\sum_{\substack{w \in N_u^+ \setminus u, \\ w \in C(u)}} \sum_{v \in V(w)} \frac{1}{a_{uv}}}_{S_1} + \underbrace{\sum_{\substack{w \in N_u^+, \\ w \in \overline{C(u)}}} \sum_{v \in V(w)} \frac{1}{a_{uv}}}_{S_2}.$$

First we will bound S_1:

$$\begin{aligned} S_1 &\le \sum_{\substack{w \in N_u^+ \setminus u, \\ w \in C(u)}} \sum_{v \in V(w)} \frac{1}{|N_u^+|} && (2) \\ &= \sum_{\substack{w \in N_u^+ \setminus u, \\ w \in C(u)}} \frac{|V(w)|}{|N_u^+|} \le \sum_{\substack{w \in N_u^+ \setminus u, \\ w \in C(u)}} \frac{y(w)}{|N_u^+|} && (3) \\ &\le \frac{\mathsf{OPT}}{|N_u^+|} \cdot |(N_u^+ \setminus u) \cap C(u)| \le \mathsf{OPT}. \end{aligned}$$

Line (2) follows from Fact 1 along with the fact that $|N_u^- \cap N_v^-| \le |N_u^-|$. The inequality in Line (3) follows from the fact that if $w \in C(u) \setminus u$, then every member of $V(w)$ is in a different cluster than w, and has a positive edge to w; so $|V(w)| \le y(w)$.

Bounding S_2 is more involved. For $w \in N_u^+ \cap \overline{C(u)}$, define

$$S_2(w) := \sum_{v \in V(w)} \frac{1}{n - |N_u^- \cap N_v^-|},$$

so that we can write

$$S_2 = \sum_{\substack{w \in N_u^+, \\ w \in \overline{C(u)}}} S_2(w).$$

Fix $w \in N_u^+ \cap \overline{C(u)}$. We will bound $S_2(w)$.

$$\begin{aligned} S_2(w) &= \sum_{\substack{v \in V(w), \\ v \in \overline{C(w)}}} \frac{1}{a_{uv}} + \sum_{\substack{v \in V(w), \\ v \in C(w)}} \frac{1}{a_{uv}} \\ &\leq \frac{|V(w) \cap \overline{C(w)}|}{|N_u^+|} + \sum_{\substack{v \in V(w), \\ v \in C(w)}} \frac{1}{a_{uv}} \qquad (4) \\ &\leq \frac{y(w)}{|N_u^+|} + \sum_{\substack{v \in V(w), \\ v \in C(w)}} \frac{1}{n - |N_u^- \cap N_v^-|}. \qquad (5) \end{aligned}$$

In line (4) we have used that $n - |N_u^- \cap N_v^-| \geq |N_u^+|$, and in line (5) we have used that the edges from w to $V(w) \cap \overline{C(w)}$ are positive and in disagreement.

Define $v^*(w) = \arg\max_{v \in V(w) \cap C(w)} |N_u^- \cap N_v^-|$. Continuing from line (5), let $a_{uv} = n - |N_u^- \cap N_v^-|$, $a_{uv^*} = n - |N_u^- \cap N_{v^*(w)}^-|$ for shorthand, and see that:

$$\begin{aligned} S_2(w) &\leq \frac{y(w)}{|N_u^+|} + \sum_{\substack{v \in V(w), \\ v \in C(w), \\ v \in N_{v^*(w)}^+}} \frac{1}{a_{uv}} + \sum_{\substack{v \in V(w), \\ v \in C(w), \\ v \in N_{v^*(w)}^-}} \frac{1}{a_{uv}} \qquad (6) \\ &\leq \frac{y(w)}{|N_u^+|} + \sum_{\substack{v \in V(w), \\ v \in C(w), \\ v \in N_{v^*(w)}^+}} \frac{1}{a_{uv^*}} + \sum_{\substack{v \in V(w), \\ v \in C(w), \\ v \in N_{v^*(w)}^-}} \frac{1}{|N_u^+|} \qquad (7) \\ &\leq \frac{y(w)}{|N_u^+|} + \sum_{\substack{v \in V(w), \\ v \in C(w), \\ v \in N_{v^*(w)}^+}} \frac{1}{a_{uv^*}} + \frac{y(v^*(w))}{|N_u^+|} \qquad (8) \\ &= \frac{y(w)}{|N_u^+|} + \frac{|V(w) \cap C(w) \cap N_{v^*(w)}^+|}{a_{uv^*}} + \frac{y(v^*(w))}{|N_u^+|} \\ &\leq \frac{y(w)}{|N_u^+|} + \frac{|N_u^- \cap N_{v^*(w)}^+|}{a_{uv^*}} + \frac{y(v^*(w))}{|N_u^+|} \qquad (9) \\ &\leq \frac{y(w)}{|N_u^+|} + 1 + \frac{y(v^*(w))}{|N_u^+|} \leq 1 + \frac{2 \cdot \mathsf{OPT}}{|N_u^+|}. \qquad (10) \end{aligned}$$

Line (6) follows from the fact that $v^*(w) \in N_{v^*(w)}^+$. Line (7) follows from the definition of $v^*(w)$ along with the previously used bound of $n - |N_u^- \cap N_v^-| \geq |N_u^+|$. Line (8) follows from the fact that $v^*(w)$ and $N_{v^*(w)}^- \cap C(w)$ are in the same cluster $C(w)$, so the edges between $v^*(w)$ and $N_{v^*(w)}^- \cap C(w)$ are disagreements incident to $v^*(w)$. Line (9) follows from the fact that every vertex in $V(w)$ is in N_u^-, and line (10) follows from Fact 2.

Having a bound for $S_2(w)$, we can bound S_2 and then S and S^-, which will finish the proof of Claim 2.

$$\begin{aligned} S_2 = \sum_{\substack{w \in N_u^+, \\ w \in \overline{C(u)}}} S_2(w) &\leq \sum_{\substack{w \in N_u^+, \\ w \in \overline{C(u)}}} \left(\frac{2 \cdot \mathsf{OPT}}{|N_u^+|} + 1 \right) \\ &\leq 2 \cdot \mathsf{OPT} + \sum_{\substack{w \in N_u^+, \\ w \in \overline{C(u)}}} 1 \leq 3 \cdot \mathsf{OPT}. \end{aligned}$$

Finally, we finish the proof of the claim because

$$S^- \leq \mathsf{OPT} + S_1 + S_2 \leq \mathsf{OPT} + \mathsf{OPT} + 3 \cdot \mathsf{OPT} = 5 \cdot \mathsf{OPT}.$$

□

4.2. Completing the proof of Theorem 1.1

Tying it all together, we prove our first combinatorial approximation algorithm for Min Max correlation clustering.

Proof of Theorem 1.1. For the clustering $\mathcal{C}$ output by the rounding algorithm (Algorithm A) of Kalhan, Makarychev, and Zhou with the correlation metric as input, let $\mathsf{ALG}(u, v) = \mathbb{1}((u, v)$ is in disagreement in $\mathcal{C})$ and $\mathsf{ALG}(u) = \sum_{v \in V} \mathsf{ALG}(u, v)$. We see that

$$\mathsf{ALG}(u) = \sum_{v \in V} \mathsf{ALG}(u, v) \underset{*}{\leq} 5 \cdot \sum_{v \in V} \widehat{d}_{uv} \leq 40 \cdot \mathsf{OPT}. \qquad (11)$$

The first inequality (*) follows because the correlation metric is a feasible solution to LP 3.1, since all $0 \leq d_{uv} \leq 1$ and by Lemma C.1 d satisfies the triangle inequality. The second inequality follows by Lemma 4.2.

Next, we analyze the run-time. Our algorithm has two phases: in phase (1) it computes the correlation metric for all $u, v \in V$, i.e. d_{uv}, and in phase (2) it uses the rounding algorithm (Algorithm A) with input d. Phase (1) of our algorithm takes $O(n^\omega)$ time, where ω is the matrix multiplication constant. To see this, observe that if P is an adjacency matrix, P^2 counts paths of length 2 between each pair of vertices, so we may compute $|N_u^+ \cap N_v^+|$ for all u, v by taking P to be the adjacency matrix of positive edges. (Note $n - |N_u^- \cap N_v^-| = |N_u^+| + |N_v^+| - |N_u^+ \cap N_v^+|$, so we can reuse the computation of $|N_u^+ \cap N_v^+|$ here.) Phase (2) takes time $O(n^2)$ (see Appendix A). □

5. Faster Algorithm via Sampling

In this section, we show that instead of computing the correlation metric exactly for each pair u, v, it suffices to es-

timate these via samples of the graph. In doing so, we can improve the run-time of our algorithm from $O(n^\omega)$ to $O(n^2 \log n)$. To do this, we show: (1) the fractional cost of the estimates is a constant factor away from OPT w.h.p. (Proposition 5.3), (2) the estimates satisfy an approximate triangle inequality w.h.p. (Proposition 5.2), and (3) inputting a function that approximately satisfies the triangle inequality to the rounding algorithm (Algorithm A) is sufficient for obtaining a constant factor in $(*)$ of line (11) (Lemma E.9). These three steps imply Theorem 1.3.

We will denote the initial estimates for d_{uv} by $\bar{d}_{uv}$ and prove properties of the $\bar{d}_{uv}$ in Section 5.1. Then, we will have a post-processing phase, in which we round some d_{uv} down to 0 and some up to 1; this step, while non-obvious, is needed to control the fractional cost. We examine the post-processing phase in Section 5.2. Our final inputs, which we use as input to the rounding algorithm, will be denoted $\widetilde{d}_{uv}$.

5.1. Initial estimates for the correlation metric

We defer all technical proofs in this section to Appendix E.

Fix $0 < \varepsilon < 1$. For every vertex u, sample $\lceil C(\varepsilon) \cdot \log n \rceil$ vertices from $|N_u^+|$, where $C(\varepsilon) = 32/\varepsilon^2$. Assume first that $|N_u^+| \geq \lceil C(\varepsilon) \cdot \log n \rceil$, as otherwise, the estimate is exact. Call this sample S_u. We use these samples to estimate terms in (1), e.g., counting how many vertices in S_u have a positive edge to v, and then scaling up this number appropriately, gives an estimate for $|N_u^+ \cap N_v^+|$.

For two vertices u, v, define the random variables:

$$X_+^{(u,v)} = \sum_{w \in N_u^+ \cap N_v^+} \mathbf{1}_{\{w \in S_u\}},$$

$$X_-^{(u,v)} = \sum_{w \in N_u^+ \cap N_v^-} \mathbf{1}_{\{w \in S_u\}},$$

$$W^{(u,v)} = \frac{|N_u^+|}{\lceil C(\varepsilon) \log n \rceil} \cdot X_+^{(u,v)}$$

$$Y^{(u,v)} = \frac{|N_u^+|}{\lceil C(\varepsilon) \log n \rceil} \cdot X_-^{(u,v)}$$

The superscripts are *ordered* pairs. If $|N_u^+| < \lceil C(\varepsilon) \cdot \log n \rceil$, set $W^{(u,v)} = |N_u^+ \cap N_v^+|$ and $Y^{(u,v)} = |N_u^+ \cap N_v^-|$. Observe that $X_+^{(u,v)} + X_-^{(u,v)} = |S_u| = \lceil C(\varepsilon) \cdot \log n \rceil$, so

$$Y^{(u,v)} = |N_u^+| - W^{(u,v)}. \tag{12}$$

$Y^{(u,v)}$ will serve as the estimate for $|N_u^+ \cap N_v^-|$ and $W^{(u,v)}$ will serve as an estimate for $|N_u^+ \cap N_v^+|$.

Flipping the order of u and v in the superscripts gives that $Y^{(v,u)}$ is an estimate of $|N_u^- \cap N_v^+|$, and $W^{(v,u)}$ is a second estimate for $|N_u^+ \cap N_v^+|$. Also, observe that

$$\mathbf{E}[X_+^{(u,v)}] = |N_u^+ \cap N_v^+| \cdot \frac{\lceil C(\varepsilon) \log n \rceil}{|N_u^+|} \tag{13}$$

$$\mathbf{E}[X_-^{(u,v)}] = |N_u^+ \cap N_v^-| \cdot \frac{\lceil C(\varepsilon) \log n \rceil}{|N_u^+|} \tag{14}$$

and similar statements for when the order of u, v is flipped in the superscripts.

Let u, v be labelled so that $|N_v^+| \geq |N_u^+|$. We define the *initial* estimate for d_{uv} (before post-processing) as

$$\bar{d}_{uv} = \frac{Y^{(u,v)} + Y^{(v,u)}}{|N_u^+| + Y^{(v,u)}} = \frac{|N_u^+| - W^{(u,v)} + |N_v^+| - W^{(v,u)}}{|N_u^+| + |N_v^+| - W^{(v,u)}}, \tag{15}$$

where the second equality holds by equation (12). Note this is well-defined since $|N_u^+| \geq 1$ due to u's positive self-loop. Also note that in the denominator, we use the estimate $W^{(v,u)}$ of $|N_u^+ \cap N_v^+|$ from v's sample, rather than u's.

Fact 3. For every $u, v \in V$, $W^{(u,v)}, Y^{(u,v)} \geq 0$. Also, $0 \leq \bar{d}_{uv} \leq 1$.

Since $|N_u^+| = |N_u^+ \cap N_v^+| + |N_u^+ \cap N_v^-|$, at least one of the two summands is at least $\frac{1}{2}|N_u^+|$. As a result, we can show that at least one of the estimates of $|N_u^+ \cap N_v^+|$ and $|N_u^+ \cap N_v^-|$ is well-concentrated (see Propositions E.2 and E.3) using Chernoff-Hoeffding bounds (Theorem E.1). This concentration ensures that we are able to approximate $\bar{d}_{uv}$ with d_{uv} from both above and below (see Propositions E.4 and E.5). In total, we obtain the following approximate triangle inequality for $\bar{d}$; the proof is in Appendix E.1.

Proposition 5.1. *Fix $\varepsilon > 0$. An approximate triangle inequality holds for all triplets u, v, w simultaneously with probability $1 - O\left(\frac{1}{n}\right)$:*

$$\bar{d}_{uv} \leq c_3(\varepsilon) \cdot (\bar{d}_{uw} + \bar{d}_{vw}) + h_3(\varepsilon)$$

where $c_3(\varepsilon), h_3(\varepsilon)$ (defined in Appendix E.1) approach 1 and 0, resp., as $\varepsilon \to 0$.

5.2. Post-processed estimates of correlation metric

Now we define our final estimates $\widetilde{d}_{uv}$:

$$\widetilde{d}_{uv} = \begin{cases} 0 & \bar{d}_{uv} \leq 2h_3(\varepsilon) \text{ and } (u,v) \in E^+ \\ 1 & \bar{d}_{uv} \geq 1/(2c_3(\varepsilon)+1) \text{ and } (u,v) \in E^- \\ \bar{d}_{uv} & \text{otherwise} \end{cases}.$$

In order to show that our algorithm is successful using $\widetilde{d}_{uv}$ in place of the exact distances d_{uv}, we need to demonstrate that w.h.p. (1) $\widetilde{d}_{uv}$ satisfy an approximate triangle inequality (Proposition 5.2), and (2) the fractional cost of $\widetilde{d}_{uv}$ can be compared to OPT (Proposition 5.3). In particular, rounding the estimates to $\widetilde{d}$ allows us to trade a small loss in the approximate triangle inequality that $\bar{d}$ satisfied for the ability to bound the fractional cost.

The proofs of the next two propositions are deferred to Appendix E.2. We then finish the proof of Theorem 1.3 in Appendix E.5.

Proposition 5.2. *Fix $\varepsilon > 0$. An approximate triangle inequality holds for all triplets u, v, w simultaneously with probability at least $1 - O(1/n)$:*

$$\widetilde{d}_{uv} \leq c_4(\varepsilon) \cdot \left(\widetilde{d}_{uw} + \widetilde{d}_{vw}\right) + h_4(\varepsilon),$$

where $c_4(\varepsilon), h_4(\varepsilon)$ (defined in Appendix E.2) approach 3 and 0, resp., as $\varepsilon \to 0$.

Proposition 5.3. *Fix $0 < \varepsilon < 0.03$. For $D(\varepsilon) = 2c_3(\varepsilon)$, the following holds with probability at least $1 - O(1/n)$:*

$$\sum_{v \in N_u^+} \widetilde{d}_{uv} + \sum_{v \in N_u^-} (1 - \widetilde{d}_{uv})$$
$$\leq D(\varepsilon)\left(\sum_{v \in N_u^+} d_{uv} + \sum_{v \in N_u^-} (1 - d_{uv})\right) \leq 8 \cdot D(\varepsilon)\mathsf{OPT}.$$

6. Experiments

In this section, we describe experiments supporting our theoretical results.[5] We demonstrate:

- The guarantees of Theorem 1.1 are predictive of our algorithm's performance on real-world and synthetic datasets. Our solution's quality is similar to the KMZ algorithm.
- The fractional cost of the correlation metric is similar to that of the objective value of LP 3.1.
- Our algorithm is *scalable*: we can handle large graphs (up to $\approx 10,000$ vertices), whereas the KMZ algorithm is only practical for graphs with up to ≈ 300 vertices due to the bottleneck of solving the LP.
- The large clusters found by our algorithm can be meaningful, in that the algorithm partially discovers "ground truth" clusters in real and synthetic instances.

Our experiments focus on the exact algorithm. We observe that empirically the exact algorithm is sufficiently fast.

Real dataset description. We obtained datasets representing social networks from the Stanford Large Network Dataset Collection (Leskovec & Mcauley, 2012).[6] Specifically, we used the *ego-Facebook* dataset containing 10 graphs that are subgraphs of a social network from Facebook. Each subgraph, or *ego-network*, represents a specific user's friend list and the connections within it. We converted this to a complete, signed graph by representing a connection between users as a positive edge, and a non-connection as a negative edge. (See Tables 4 and 5 in Appendix F for statistics on these graphs.) Each ego-network is accompanied by "ground truth" circles; each circle is a collection of vertices that the user has labelled as a community. Note the circles are not necessarily partitions of the friend list.

For each Facebook ego-network, we applied our exact algorithm using the matrix multiplication implementation. For five of the ego-networks that were of small enough size, we also solved the LP in order to bound the approximation ratio of our algorithm. The LP solver used was Gurobi. For the latter datasets, we also applied the KMZ algorithm as an additional means of comparison. Let r_1 be the radius in $L_t(\cdot)$ and let r_2 be the radius used to cut out C_t in Algorithm A. While Theorem 1.1 holds for $r_1 = 1/5$ and $r_2 = 2/5$, in practice these radii may give an objective value near the maximum positive degree in a sparse graph. We can obtain even better results than those guaranteed by Theorem 1.1 by setting the hyperparameters less conservatively. We did parameter sweeps (Figure 3 in Appendix F.2), and found that across the datasets, $r_1 = r_2 = 0.7$ work well for our algorithm, and $r_1 = r_2 = 0.4$ work well for the KMZ algorithm, so we report the results using these parameters. (See Appendix F.3 for the best radii for each dataset.) Finally, we applied the Pivot algorithm (described in Appendix F.1) to all datasets for an additional comparison (Ailon et al., 2008). See Tables 1 and 2 and Table 3.

Quality of approximation. For the five small datasets in Table 1, the cost of our algorithm is at most 2 times the cost of the LP, and thus at most twice optimal. Our algorithm and the KMZ algorithm performed similarly in terms of objective value. We note that we do not claim our performance is better than that of the KMZ algorithm (e.g., compare Tables 1 and 6), but that they perform similarly (our algorithm always gives an objective value at most 10% more than that of KMZ). In addition, the fractional cost of the correlation metric and the cost of the LP consistently differ by a factor of around 2. Finally, the objective value of our algorithm is typically slightly less than its fractional cost. For the large data sets (Table 2) for which it was prohibitive to run the LP, we do not have a lower bound on optimal due to the LP not scaling, so we cannot bound the approximation ratio. For these datasets, we compare to Pivot, which our algorithm outperforms by a substantial margin.

	frac. cost	LP obj.	our obj.	KMZ obj.	Pivot obj.
FB 348	74.37	39.13	72	89	85.03
FB 414	35.53	19.66	34	38	50.73
FB 686	58.59	30.48	47	69	65.72
FB 698	22.31	10.64	20	18	23.51
FB 3980	14.31	7.34	12	13	16.36

Table 1. Comparison of the fractional values and objective values of our algorithm and the KMZ algorithm for the five small Facebook datasets. The column "frac. cost" records the fractional cost of the correlation metric. We also run the Pivot algorithm; the recorded value is the objective value averaged over 500 trials.

[5] The code for our experiments can be found at https://github.com/hanewman/MinMax-Correlation-Clustering-

[6] https://snap.stanford.edu/data/ego-Facebook.html

	frac. cost	our obj.	Pivot obj.	our run-time	# vertices
FB 0	64.02	49	71.78	0.20	333
FB 107	181.49	152	216.65	1.76	1034
FB 1684	103.99	93	130.71	0.98	786
FB 1912	227.74	220	259.01	0.93	747
FB 3437	98.36	107	99.1	0.47	534

Table 2. Columns are as in Table 1. The size of each dataset here makes the LP run-time prohibitive. Run-time is listed in seconds.

	our run-time	KMZ run-time	#vertices
FB 348	0.10	1847.99	224
FB 414	0.06	207.92	150
FB 686	0.06	337.9	168
FB 698	0.02	3.42	61
FB 3980	0.01	2.03	52

Table 3. A comparison of the run-times (in seconds) of our algorithm and the KMZ algorithm on the five small Facebook datasets.

Run-time and scalability. The run-time of our algorithm is significantly faster than that of the KMZ algorithm (Table 3). For instance, on FB 348, which contains only 224 vertices, the KMZ algorithm took over 30 minutes, whereas our algorithm took a tenth of a second.[7] In fact, we can quickly handle very large graphs; on a social network with 12,008 nodes[8], our algorithm ran in just over 4 minutes. See Appendix F.5 for more details on scalability.

Comparison to ground truth clusters. We also analyzed whether the clusters found by our algorithm discovered the ground truth circles identified by users. For each dataset, we considered "large" clusters of size at least 10, since the small clusters (including several singleton clusters) are less meaningful. For each large cluster, we identified the ground truth circle containing the largest number of vertices from that cluster. The results are plotted in Figure 4 in Appendix F.2. We find that for datasets FB 348, 414, 686, 1684, and 1912, almost every large cluster is almost entirely contained in its best ground truth circle (i.e., between 80% and 100% of each large cluster is contained in its best ground truth circle). For other datasets, there is less evidence that ground truth circles are being discovered.

Synthetic datasets. We considered synthetic datasets for two reasons. The first is that running the LP and the KMZ algorithm are prohibitive on many real-world datasets. The second is to further test whether our algorithm discovers ground truth clusters. We took a graph with 100 vertices and 10 positive cliques of size 10 (so the graph admits a perfect clustering) and introduced 20 levels of noise. At each level i, we randomly flipped $45i$ edges to the opposite sign. We then applied our algorithm using $r_1 = r_2 = 0.7$ as before. We found that for up to 495 flips ($i = 11$), the original clusters of size 10 were almost entirely preserved by our algorithm (in some cases, up to three vertices popped out into singleton clusters). We also found that for all levels of noise we considered, almost every cluster we found was at least 88% contained in a ground truth cluster (only 6 clusters were an exception to this). See Appendix F.4 for additional plots of these experiments.

[7] Even if we terminate the LP early, e.g., when the primal and dual are within 1 of each other, this still takes at least 10 minutes. Note however that doing so would affect the distances outputted.

[8] https://snap.stanford.edu/data/feather-lastfm-social.html

7. Conclusion

This paper presents a fast, efficient, and scalable combinatorial $O(1)$-approximation algorithm for Min Max correlation clustering. The algorithm constructs a provably good fractional solution to a LP, and rounds this solution using a previous work's LP rounding algorithm.

A future direction is to extend our algorithmic techniques to other ℓ_p norms for correlation clustering. More generally, given an LP, when can one use combinatorial properties of the underlying instance's structure to form a provably good fractional solution to the LP? This general framework could lead to run-time improvements for other problems. While these directions are theoretically interesting in their own right, there is practical motivation for finding fast algorithms for other ℓ_p norms, since $p \in (1, \infty)$ interpolates between the competing objectives of local fairness ($p = \infty$) and global optimality ($p = 1$).

Acknowledgements

Sami Davies was supported by an NSF Computing Innovation Fellowship. Benjamin Moseley and Heather Newman were supported in part by a Google Research Award, an Informs Research Award, a Carnegie Bosch Junior Faculty Chair, and NSF grants CCF-2121744 and CCF-1845146.

References

Ahmadi, S., Khuller, S., and Saha, B. Min-max correlation clustering via multicut. In *International Conference on Integer Programming and Combinatorial Optimization*, pp. 13–26. Springer, 2019.

Ahmadi, S., Galhotra, S., Saha, B., and Schwartz, R. Fair correlation clustering. *arXiv preprint arXiv:2002.03508*, 2020.

Ahmadian, S., Epasto, A., Kumar, R., and Mahdian, M. Fair correlation clustering. In *International Conference on Artificial Intelligence and Statistics*, pp. 4195–4205. PMLR, 2020.

Ailon, N., Charikar, M., and Newman, A. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):1–27, 2008.

Bansal, N., Blum, A., and Chawla, S. Correlation clustering. *Machine Learning*, 56(1):89–113, 2004.

Bateni, M., Cohen-Addad, V., Epasto, A., and Lattanzi, S. Scalable and improved algorithms for individually fair clustering. In *Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS 2022*, 2022. URL https://openreview.net/forum?id=gdPiOBu99Y.

Ben-Dor, A. and Yakhini, Z. Clustering gene expression patterns. In *Proceedings of the Third Annual International Conference on Computational Molecular Biology*, pp. 33–42, 1999.

Charikar, M., Gupta, N., and Schwartz, R. Local guarantees in graph cuts and clustering. In *International Conference on Integer Programming and Combinatorial Optimization*, pp. 136–147. Springer, 2017.

Chawla, S., Makarychev, K., Schramm, T., and Yaroslavtsev, G. Near optimal lp rounding algorithm for correlation clustering on complete and complete k-partite graphs. In *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, pp. 219–228, 2015.

Cheng, Y. and Church, G. M. Biclustering of expression data. In *Intelligent Systems for Molecular Biology*, volume 8, pp. 93–103, 2000.

Cohen, M. B., Lee, Y. T., and Song, Z. Solving linear programs in the current matrix multiplication time. *Journal of the ACM (JACM)*, 68(1):1–39, 2021.

Cohen-Addad, V., Lee, E., and Newman, A. Correlation clustering with sherali-adams. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 651–661. IEEE, 2022.

Demaine, E. D. and Immorlica, N. Correlation clustering with partial information. In *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques: APPROX 2003 and RANDOM 2003, Princeton, NJ*, pp. 1–13. Springer, 2003.

Friggstad, Z. and Mousavi, R. Fair correlation clustering with global and local guarantees. In *Workshop on Algorithms and Data Structures*, pp. 414–427. Springer, 2021.

Jafarov, J., Kalhan, S., Makarychev, K., and Makarychev, Y. Local correlation clustering with asymmetric classification errors. In *International Conference on Machine Learning*, pp. 4677–4686. PMLR, 2021.

Kalhan, S., Makarychev, K., and Zhou, T. Correlation clustering with local objectives. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/785ca71d2c85e3f3774baaf438c5c6eb-Paper.pdf.

Kriegel, H.-P., Kröger, P., and Zimek, A. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(1):1–58, 2009.

Leskovec, J. and Mcauley, J. Learning to discover social circles in ego networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/7a614fd06c325499f1680b9896beedeb-Paper.pdf.

Leskovec, J., Kleinberg, J., and Faloutsos, C. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1 (1):2–es, 2007.

McCallum, A. and Wellner, B. Conditional models of identity uncertainty with application to noun coreference. In Saul, L., Weiss, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004. URL https://proceedings.neurips.cc/paper_files/paper/2004/file/1680829293f2a8541efa2647a0290f88-Paper.pdf.

Puleo, G. and Milenkovic, O. Correlation clustering and biclustering with locally bounded errors. In *International Conference on Machine Learning*, pp. 869–877. PMLR, 2016.

Rozemberczki, B. and Sarkar, R. Characteristic Functions on Graphs: Birds of a Feather, from Statistical Descriptors to Parametric Models. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*, pp. 1325–1334. ACM, 2020.

Symeonidis, P., Nanopoulos, A., Papadopoulos, A. N., and Manolopoulos, Y. Nearest-biclusters collaborative filtering based on constant and coherent values. *Information Retrieval*, 11(1):51–75, 2008.

Wirth, A. Correlation clustering. In Sammut, C. and Webb, G. I. (eds.), *Encyclopedia of Machine Learning and Data Mining*, pp. 280–284. Springer, 2017. doi: 10.1007/

978-1-4899-7687-1_176. URL https://doi.org/10.1007/978-1-4899-7687-1_176.

A. Rounding Algorithm and Run-time

Rounding algorithm We detail the rounding algorithm that we will leverage (Kalhan et al., 2019). For vertex $u \in V$ let the ball of radius ρ around u with respect to a semi-metric x on V be defined as $\mathsf{Ball}(u, \rho) = \{v \in V \mid x_{uv} \leq \rho\}$. The algorithm is iterative, where vertices are clustered in each iteration. At step t, the set of unclustered vertices is denoted $V_t \subseteq V$. From V_t, a special vertex is chosen to be the cluster center, specifically the vertex that maximizes $L_t(u) = \sum_{v \in \mathsf{Ball}(u,r) \cap V_t} r - x_{uv}$ which indicates how packed towards the center the vertices in $\mathsf{Ball}(u, r) \cap V_t$ are.

Algorithm A (Rounding Algorithm)

Input: Semi-metric x on V.
Output: Clustering $\mathcal{C}$.

1. Let $V_0 = V$, $r = 1/5$, $t = 0$.
2. **while** ($V_t \neq \emptyset$)
   - Find $u_t^* = \arg\max_{u \in V_t} L_t(u) = \arg\max_{u \in V_t} \sum_{v \in \mathsf{Ball}(u,r) \cap V_t} r - x_{uv}$.
   - Create $C_t = \mathsf{Ball}(u_t^*, 2r) \cap V_t$.
   - Set $V_{t+1} = V_t \setminus C_t$ and $t = t + 1$.
3. Return $\mathcal{C} = (C_0, \ldots, C_{t-1})$.

Let $\mathsf{LP}(u, v)$ be the cost of edge (u, v) to the LP in its objective value, so one can set $x_{uv} = \mathsf{LP}(u, v)$ if $(u, v) \in E^+$ and $x_{uv} = 1 - \mathsf{LP}(u, v)$ if $(u, v) \in E^-$. For the output of the algorithm above, let $\mathsf{ALG}(u, v) = \mathbb{1}((u, v)$ is in disagreement) and $\mathsf{ALG}(u) = \sum_{v \in V} \mathsf{ALG}(u, v)$. Kalhan, Makarychev, and Zhou show that using an optimal LP solution x as input,

$$\mathsf{ALG}(u) = \sum_{v \in V} \mathsf{ALG}(u, v) \underset{*}{\leq} 5 \cdot \sum_{v \in V} \mathsf{LP}(u, v) \leq 5y(u), \tag{16}$$

which leads to a 5 approximation algorithm for any ℓ_p norm for complete graphs. The technical work in their result is showing the inequality marked with * in Equation 16. (The second inequality is due to the LP being a relaxation.)

Run-time We will first justify that the run-time of the rounding algorithm is $O(n^2)$, and can be obtained through the following procedure:

- For each $u \in V$, precompute $L_t(u) = \sum_{v \in Ball(u,r)} r - d_{uv}$. This takes $O(n^2)$ time.
- At iteration t, it takes $O(n)$ time to find the max of $L_t(u)$ over all unclustered vertices, i.e., over $u \in V_t$, and $O(n)$ time to create the cluster. This contributes $O(n^2)$ time over all iterations.
- When a vertex is removed, its contribution to $L_t(u)$ for remaining vertices u must be removed as well. This takes $O(n)$ time for each vertex that is removed, as it may be a member of $O(n)$ balls. Further, each vertex is only removed once. So the updates to the L_t values take $O(n^2)$ time overall.

In all, the run-time of the full KMZ algorithm is dominated by the time it takes to solve the LP to obtain the semi-metric x. The LP contains $O(n^2)$ many variables and $O(n^3)$ many constraints, for $n = |V|$. Given the current best algorithms for solving linear programs, this obliviously gives a run-time that is no better than $O(n^{2\omega})$ for solving the LP (Cohen et al., 2021). We justify that this is a reasonable theoretical benchmark to compare against for run-time. On sparse networks, i.e. when the graph on the $+$ edges is sparse, one might wonder whether it is possible to reduce the number of variables that the solver must solve for by fixing the values of the LP variables for pairs of vertices whose positive neighborhoods have empty intersection. We believe it would be interesting to try this experimentally to see whether the solution is optimal/ near-optimal and whether it is obtained more quickly by the solver. If empirically it were the case that the solution is near-optimal, then one could try to back these findings with theoretical guarantees. Lastly, it might be possible to more quickly obtain an LP solution which is provably approximately optimal by solving the LP, but such study would be quite technical and a completely different approach.

B. Omitted Proofs from Section 4.1

Proof of Proposition 4.1. Take $S = (N_u^+ \cap N_v^-) \cup (N_u^- \cap N_v^+)$. Note that $N_u^+ \cap N_v^-$ and $N_u^- \cap N_v^+$ are disjoint, so $|S| = |N_u^+ \cap N_v^-| + |N_u^- \cap N_v^+|$. Fix $w \in S$. Since u and v are in the same cluster, exactly one of (u, w) and (v, w) is a disagreement, and contributes to one of $y(u)$ or $y(v)$. □

Proof of Claim 1. Observe that by Fact 2, for every $v \in V$,

$$|N_u^+| = |N_u^+ \cap N_v^-| + |N_u^+ \cap N_v^+| \leq n - |N_u^- \cap N_v^-|.$$

Now we can bound S^+ by partitioning the sum based one whether or not u and v are in the same cluster:

$$\begin{aligned}
S^+ &= \sum_{v \in N_u^+ \cap C(u)} \frac{|N_u^+ \cap N_v^-| + |N_u^- \cap N_v^+|}{n - |N_u^- \cap N_v^-|} + \sum_{v \in N_u^+ \cap \overline{C(u)}} d_{uv} \\
&\leq \frac{1}{|N_u^+|} \cdot \sum_{v \in N_u^+ \cap C(u)} (|N_u^+ \cap N_v^-| + |N_u^- \cap N_v^+|) + \sum_{v \in N_u^+ \cap \overline{C(u)}} 1 \\
&\leq \frac{1}{|N_u^+|} \cdot \sum_{v \in N_u^+ \cap C(u)} (y(u) + y(v)) + \sum_{v \in N_u^+ \cap \overline{C(u)}} 1 && (17) \\
&\leq y(u) + \max_z y(z) + \sum_{v \in N_u^+ \cap \overline{C(u)}} 1 && (18) \\
&\leq y(u) + \max_z y(z) + y(u) && (19) \\
&\leq 3 \cdot \mathsf{OPT}.
\end{aligned}$$

Line (17) follows from Proposition 4.1, line (18) follows from an averaging argument (we sum over $|N_u^+ \cap C(u)|$ terms and then divide by $|N_u^+|$), and line (19) follows from the fact that if $(u, v) \in E^+$ and u, v are in different clusters, then (u, v) is a disagreement incident to u. □

C. Correlation metric satisfies triangle inequality

We will show that the correlation metric satisfies the triangle inequality, i.e., for distinct $u, v, w \in V$, $d_{uv} + d_{vw} \geq d_{uw}$.

Lemma C.1. *The correlation metric satisfies the triangle inequality.*

Proof. Fix three distinct vertices $u, v, w \in V$. We see from the definition of d that $d_{uv} + d_{vw} \geq d_{uw}$ if and only if

$$1 - \frac{|N_u^+ \cap N_v^+|}{n - |N_u^- \cap N_v^-|} + 1 - \frac{|N_w^+ \cap N_v^+|}{n - |N_w^- \cap N_v^-|} \geq 1 - \frac{|N_u^+ \cap N_w^+|}{n - |N_u^- \cap N_w^-|},$$

which after rearranging is equivalent to

$$\frac{|N_u^+ \cap N_w^+|}{n - |N_u^- \cap N_w^-|} + 1 \geq \frac{|N_u^+ \cap N_v^+|}{n - |N_u^- \cap N_v^-|} + \frac{|N_w^+ \cap N_v^+|}{n - |N_w^- \cap N_v^-|}.$$

For shorthand, allow $a_{uv} = n - |N_u^- \cap N_v^-|$, and analogously define a_{uw} and a_{vw}. Then we can multiply both sides of the above inequality by $a_{uw} \cdot a_{uv} \cdot a_{vw}$ to rewrite it as

$$a_{uv} a_{wv} \left(|N_u^+ \cap N_w^+| + n - |N_u^- \cap N_w^-|\right) \geq a_{uw} a_{vw} |N_u^+ \cap N_v^+| + a_{uw} a_{uv} |N_w^+ \cap N_v^+|. \tag{20}$$

We can rewrite the left-hand side of Equation 20 using Fact 2 and expand it into the intersection of all three vertices as

$$\begin{aligned}
a_{uv} a_{vw} \left(|N_u^+ \cap N_w^+| + a_{uw}\right) &= a_{uv} a_{vw} (|N_u^+ \cap N_w^+| + |N_u^+ \cap N_w^+| + |N_u^- \cap N_w^+| + |N_u^+ \cap N_w^-|) \\
&= a_{uv} a_{vw} (2|N_u^+ \cap N_v^+ \cap N_w^+| + 2|N_u^+ \cap N_v^- \cap N_w^+| \\
&\quad + |N_u^- \cap N_v^+ \cap N_w^+| + |N_u^- \cap N_v^- \cap N_w^+| \\
&\quad + |N_u^+ \cap N_v^+ \cap N_w^-| + |N_u^+ \cap N_v^- \cap N_w^-|).
\end{aligned}$$

We introduce more shorthand to compactly notate the intersections of 3 neighborhoods. Let the subscripts of the variable b denote the vertices whose positive intersection we examine, so $b_{uvw} = |N_u^+ \cap N_v^+ \cap N_w^+|$, $b_{uv} = |N_u^+ \cap N_v^+ \cap N_w^-|$, $b_u = |N_u^+ \cap N_v^- \cap N_w^-|$, etc. Therefore we can more compactly write the left-hand side of Equation 20 as

$$a_{uv} a_{vw} \left(|N_u^+ \cap N_w^+| + a_{uw}\right) = a_{uv} a_{vw} \left(2b_{uvw} + 2b_{uw} + b_{uv} + b_{vw} + b_u + b_w\right). \tag{21}$$

We next write the factors a_{uv}, a_{uw}, and a_{vw} in terms of the b variables, where we let $B = b_{uvw} + b_{uv} + b_{uw} + b_{vw}$:

$$\begin{aligned} a_{uv} &= |N_u^+ \cap N_v^+| + |N_u^+ \cap N_v^-| + |N_u^- \cap N_v^+| = b_{uv} + b_{uvw} + b_u + b_{uw} + b_v + b_{vw} \\ &= B + b_u + b_v \end{aligned}$$

Similarly, $a_{uw} = B + b_u + b_w$ and $a_{vw} = B + b_v + b_w$. Plugging these into Equation 21 the full left-hand side of Equation 20 is

$$a_{uv}a_{vw}\left(|N_u^+ \cap N_w^+| + a_{uw}\right) = (B + b_u + b_v)(B + b_v + b_w)\left(B + b_{uvw} + b_{uw} + b_u + b_w\right). \tag{22}$$

Now we move onto rewriting the right-hand side of Equation 20:

$$\begin{aligned} &a_{uw}a_{vw}|N_u^+ \cap N_v^+| + a_{uw}a_{uv}|N_w^+ \cap N_v^+| \\ &\quad = (B + b_u + b_w)(B + b_v + b_w)(b_{uvw} + b_{uv}) + (B + b_u + b_w)(B + b_u + b_v)(b_{uvw} + b_{vw}). \end{aligned} \tag{23}$$

Using Equations 22 and 23, we rewrite the condition in Equation 20 as

$$\begin{aligned} &(B + b_u + b_v)(B + b_v + b_w)\left(b_{uvw} + b_{uw} + B + b_u + b_w\right) \\ &\qquad \geq (B + b_u + b_w)(B + b_v + b_w)(b_{uvw} + b_{uv}) \\ &\qquad\quad + (B + b_u + b_w)(B + b_u + b_v)(b_{uvw} + b_{vw}), \end{aligned}$$

which one can verify is true as every term on the right-hand side is on the left-hand side too.

□

D. Proof of Corollary 1.2

Proof of Corollary 1.2. For each $u \in V$, there are at most Δ^2 vertices v such that $|N_u^+ \cap N_v^+| > 0$, so there are at most Δ^2 vertices v with $d_{uv} < 1$. We only need to compute d_{uv} for pairs $\{u, v\}$ such that v in the 2-hop neighborhood $N^2(u)$ of u. For fixed u, computing $N^2(u)$ as well as $|N_u^+ \cap N_v^+|$ for each $v \in N^2(u)$ can be done in $O(\Delta^2)$ time, so for all vertices u this takes a total of $O(n\Delta^2)$ time.

Now we need to compute d_{uv} for every $v \in N^2(u)$ (for $v \notin N^2(u)$, $d_{uv} = 1$). This takes constant time for each pair $\{u, v\}$, since

$$d_{uv} = 1 - |N_u^+ \cap N_v^+|/(n - |N_u^- \cap N_v^-|) = 1 - |N_u^+ \cap N_v^+|/\left(|N_u^+| + |N_v^+| - |N_u^+ \cap N_v^+|\right).$$

Now that we have d computed for all relevant pairs of vertices, it remains to argue the rounding algorithm (Algorithm A) also can be run faster on sparse graphs. First, initialize $L_0(u)$ for each $u \in V$. Since $r = 1/5$, we only have to check whether $d_{uv} \leq 1/5$ for $v \in N^2(u)$ (otherwise, $d_{uv} = 1$). Since $|N^2(u)| \leq \Delta^2$, initializing $L_0(\cdot)$ takes $O(n\Delta^2)$ time total. We will store $L_t(u)$ for each $u \in V_t$ in a binary heap; inserting these takes $O(n \log n)$ time total. In each phase t, finding u_t^* maximizing $L_t(u)$ will take $O(1)$ time. Then, we will have to remove all v such that $d_{u_t^* v} \leq 2/5$. Since these v must belong to $N^2(u_t^*)$, and each deletion is $O(\log n)$ time, this will take $O(\Delta^2 \log n)$ time. We will also have to decrease $L_t(u)$ to $L_{t+1}(u)$ for $u \in V_{t+1}$: For each vertex v removed during phase t, there are at most Δ^2 elements in $N^2(v)$, so v induces at most Δ^2 updates to $L_t(\cdot)$. Since each vertex is only removed once, the key update operations contribute $O(n\Delta^2 \log n)$ time total. □

E. Omitted Proofs for Section 5

E.1. Proofs: initial estimates for the correlation metric

We will repeatedly use the following tail bounds for sums of random variables.

Theorem E.1 (Chernoff-Hoeffding)**.** *Let $X = X_1 + \cdots + X_m$ where $\{X_1, \ldots, X_m\}$ is a set of negatively correlated random variables. Define $\mu = \mathbb{E}[X]$. For $0 < \varepsilon < 1$, the following tail bounds hold:*

$$\mathbb{P}(X \geq (1 + \varepsilon)\mu) \leq e^{-\varepsilon^2 \mu/4}$$

$$\mathbb{P}(X \leq (1 - \varepsilon)\mu) \leq e^{-\varepsilon^2 \mu/4}.$$

Since $|N_u^+| = |N_u^+ \cap N_v^+| + |N_u^+ \cap N_v^-|$, at least one of the two summands is at least $\frac{1}{2}|N_u^+|$. Therefore, we can show that at least one of the estimates of $|N_u^+ \cap N_v^+|$ and $|N_u^+ \cap N_v^-|$ is well-concentrated.

Proposition E.2. *Let $0 < \varepsilon < 1$. If $|N_u^+ \cap N_v^+| \geq \frac{1}{2}|N_u^+|$, then w.h.p. $W^{(u,v)} \geq (1-\varepsilon)|N_u^+ \cap N_v^+|$ and $W^{(u,v)} \leq (1+\varepsilon)|N_u^+ \cap N_v^+|$. Likewise, if $|N_u^+ \cap N_v^+| \geq \frac{1}{2}|N_v^+|$, then w.h.p. $W^{(v,u)} \geq (1-\varepsilon)|N_u^+ \cap N_v^+|$ and $W^{(v,u)} \leq (1+\varepsilon)|N_u^+ \cap N_v^+|$. In particular, each of the four events fails with probability at most $\frac{1}{n^{\varepsilon^2 C(\varepsilon)/8}}$.*

Proposition E.3. *Let $0 < \varepsilon < 1$. If $|N_u^+ \cap N_v^-| \geq \frac{1}{2}|N_u^+|$, then w.h.p. $Y^{(u,v)} \geq (1-\varepsilon)|N_u^+ \cap N_v^+|$ and $Y^{(u,v)} \leq (1+\varepsilon)|N_u^+ \cap N_v^+|$. Likewise, if $|N_u^- \cap N_v^+| \geq \frac{1}{2}|N_v^+|$, then w.h.p. $Y^{(v,u)} \geq (1-\varepsilon)|N_u^+ \cap N_v^+|$ and $Y^{(v,u)} \leq (1+\varepsilon)|N_u^+ \cap N_v^+|$. In particular, each of the four events fails with probability at most $\frac{1}{n^{\varepsilon^2 C(\varepsilon)/8}}$.*

Proof of Proposition E.2. If $|N_u^+| \leq \lceil C(\varepsilon)\log n \rceil$, then the first two bounds hold automatically. So assume $|N_u^+| \geq \lceil C(\varepsilon)\log n \rceil$. By Theorem E.1,

$$\begin{aligned}
\mathbf{P}\left(W^{(u,v)} \leq (1-\varepsilon)|N_u^+ \cap N_v^+|\right) &= \mathbf{P}\left(\frac{|N_u^+|}{\lceil C(\varepsilon)\log n\rceil} \cdot X_+^{(u,v)} \leq (1-\varepsilon)|N_u^+ \cap N_v^+|\right) \\
&= \mathbf{P}\left(X_+^{(u,v)} \leq (1-\varepsilon)|N_u^+ \cap N_v^+| \cdot \frac{\lceil C(\varepsilon)\log n\rceil}{|N_u^+|}\right) \\
&= \mathbf{P}\left(X_+^{(u,v)} \leq (1-\varepsilon)\mathbf{E}[X_u^+]\right) \\
&\leq e^{-\frac{\varepsilon^2}{4}\mathbf{E}[X_+^{(u,v)}]} \leq e^{-\frac{\varepsilon^2}{4}\cdot\lceil C(\varepsilon)\log n\rceil\cdot\frac{|N_u^+\cap N_v^+|}{|N_u^+|}} \leq e^{-\frac{\varepsilon^2}{8}\cdot C(\varepsilon)\log n},
\end{aligned}$$

where we have used equation (13) and the assumption that $|N_u^+ \cap N_v^+| \geq \frac{1}{2}|N_u^+|$ in the last two inequalities. Similarly,

$$\mathbf{P}\left(W^{(u,v)} \geq (1+\varepsilon)|N_u^+ \cap N_v^+|\right) \leq e^{-\frac{\varepsilon^2}{8}\cdot C(\varepsilon)\log n}.$$

□

Proof of Proposition E.3. The proof is similar to Proposition E.2, but instead using the assumption that $|N_u^+ \cap N_v^-| \geq (1/2)|N_u^+|$ and the fact that $\mathbf{E}[X_-^{(u,v)}] = \frac{|N_u^+\cap N_v^-|}{|N_u^+|} \cdot \lceil C(\varepsilon)\log n\rceil$ by (14). □

Using the previous two concentration results, we can derive the following two propositions.

Proposition E.4. *For any $u, v \in V$, the following holds with probability at least $1 - \frac{4}{n^{\varepsilon^2 C(\varepsilon)/8}}$:*

$$\bar{d}_{uv} \leq c_1(\varepsilon)\cdot d_{uv} + h_1(\varepsilon)$$

and further, since d_{uv} satisfy the triangle inequality, for any u, v, w,

$$\bar{d}_{uv} \leq c_1(\varepsilon)\cdot(d_{uw} + d_{vw}) + h_1(\varepsilon)$$

where $c_1(\varepsilon) \to 1$ and $h_1(\varepsilon) \to 0$ as $\varepsilon \to 0$. In particular, take $c_1(\varepsilon) = (1+\varepsilon)/(1-\varepsilon)$ and $h_1(\varepsilon) = 2\varepsilon/(1-\varepsilon)$.

Proposition E.5. *For any $u, v \in V$, the following holds with probability at least $1 - \frac{4}{n^{\varepsilon^2 C(\varepsilon)/8}}$:*

$$d_{uv} \leq c_2(\varepsilon)\cdot \bar{d}_{uv} + h_2(\varepsilon)$$

where $c_2(\varepsilon) \to 1$ and $h_2(\varepsilon) \to 0$ as $\varepsilon \to 0$. In particular, take $c_2(\varepsilon) = (1+\varepsilon)/(1-\varepsilon)$ and $h_2(\varepsilon) = 2\varepsilon/(1-\varepsilon)$.

Proof of Proposition E.4. Assume WLOG that $|N_v^+| \geq |N_u^+|$. Let A_{uv} denote the numerator of d_{uv} and B_{uv} denote the denominator, that is:

$$\begin{aligned}
A_{uv} &= |N_u^+ \cap N_v^-| + |N_u^- \cap N_v^+| \\
B_{uv} &= |N_u^+ \cap N_v^-| + |N_u^- \cap N_v^+| + |N_u^+ \cap N_v^+|
\end{aligned}$$

We case on whether or not $|N_u^+ \cap N_v^+| \geq (1/2)|N_u^+|$ and $|N_u^+ \cap N_v^+| \geq (1/2)|N_v^+|$. Note there are three possible cases instead of four, since $|N_u^+| \leq |N_v^+|$ makes it impossible that both $|N_u^+ \cap N_v^+| < (1/2)|N_u^+|$ and $|N_u^+ \cap N_v^+| \geq (1/2)|N_v^+|$ hold.

Case 1. $|N_u^+ \cap N_v^+| \geq (1/2)|N_u^+|$ and $|N_u^+ \cap N_v^+| \geq (1/2)|N_v^+|$.

Then w.h.p.

$$\begin{aligned}
\bar{d}_{uv} &= \frac{|N_u^+| - W^{(u,v)} + |N_v^+| - W^{(v,u)}}{|N_u^+| + |N_v^+| - W^{(v,u)}} \\
&\leq \frac{|N_u^+| - (1-\varepsilon)|N_u^+ \cap N_v^+| + |N_v^+| - (1-\varepsilon)|N_u^+ \cap N_v^+|}{|N_u^+| + |N_v^+| - (1+\varepsilon)|N_u^+ \cap N_v^+|} \\
&= \frac{|N_u^+ \cap N_v^-| + |N_u^- \cap N_v^+| + 2\varepsilon|N_u^+ \cap N_v^+|}{|N_u^+ \cap N_v^+| + |N_u^+ \cap N_v^-| + |N_u^- \cap N_v^+| - \varepsilon|N_u^+ \cap N_v^+|} \\
&= \frac{A_{uv} + 2\varepsilon(B_{uv} - A_{uv})}{B_{uv} - \varepsilon(B_{uv} - A_{uv})} \\
&\leq \frac{(1-2\varepsilon)A_{uv} + 2\varepsilon B_{uv}}{(1-\varepsilon)B_{uv}} = \frac{1-2\varepsilon}{1-\varepsilon} \cdot d_{uv} + \frac{2\varepsilon}{1-\varepsilon},
\end{aligned}$$

where we have used (15) in the first line and Proposition E.2 in the second line.

Case 2. $|N_u^+ \cap N_v^-| \geq \frac{1}{2}|N_u^+|$ and $|N_u^- \cap N_v^+| \geq \frac{1}{2}|N_v^+|$.

Then w.h.p.

$$\begin{aligned}
\bar{d}_{uv} &= \frac{Y^{(u,v)} + Y^{(v,u)}}{|N_u^+| + Y^{(v,u)}} \\
&\leq \frac{(1+\varepsilon)|N_u^+ \cap N_v^-| + (1+\varepsilon)|N_u^- \cap N_v^+|}{|N_u^+| + (1-\varepsilon)|N_u^- \cap N_v^+|} \\
&\leq \frac{1+\varepsilon}{1-\varepsilon} \cdot d_{uv}
\end{aligned}$$

where we have used (15) in the first line and Proposition E.3 in the second line.

Case 3. $|N_u^+ \cap N_v^+| \geq (1/2)|N_u^+|$ but $|N_u^+ \cap N_v^+| < (1/2)|N_v^+|$ (i.e., $|N_u^- \cap N_v^+| > (1/2)|N_v^+|$).

Then w.h.p.

$$\begin{aligned}
\bar{d}_{uv} &= \frac{|N_u^+| - W^{(u,v)} + Y^{(v,u)}}{|N_u^+| + Y^{(v,u)}} \\
&\leq \frac{|N_u^+| - (1-\varepsilon)|N_u^+ \cap N_v^+| + (1+\varepsilon)|N_u^- \cap N_v^+|}{|N_u^+| + (1-\varepsilon)|N_u^- \cap N_v^+|} \\
&\leq \frac{|N_u^+ \cap N_v^-| + \varepsilon|N_u^+ \cap N_v^+| + (1+\varepsilon)|N_u^- \cap N_v^+|}{(1-\varepsilon)B_{uv}} \\
&\leq \frac{(1+\varepsilon)A_{uv}}{(1-\varepsilon)B_{uv}} + \frac{\varepsilon(B_{uv} - A_{uv})}{(1-\varepsilon)B_{uv}} \\
&\leq \frac{1+\varepsilon}{1-\varepsilon} \cdot d_{uv} + \frac{\varepsilon}{1-\varepsilon}.
\end{aligned}$$

where we have used (15) in the first line and Propositions E.2 and E.3 in the second line.

□

Proof of Proposition E.5. The proof follows as that of Proposition E.4, but using the other side of the Chernoff bound. □

We combine Propositions E.4 and E.5 to show our sampling leads to Proposition E.6.

Proposition E.6. *For any $u, v, w \in V$, the following holds with probability at least $1 - \frac{12}{n^{\varepsilon^2 C(\varepsilon)/8}}$:*

$$\bar{d}_{uv} \leq c_3(\varepsilon) \cdot (\bar{d}_{uw} + \bar{d}_{vw}) + h_3(\varepsilon)$$

where $c_3(\varepsilon) \to 1$ *and* $h_3(\varepsilon) \to 0$ *as* $\varepsilon \to 0$. *In particular, take*

$$c_3(\varepsilon) = \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^2 \quad \textit{and} \quad h_3(\varepsilon) = \left(\frac{2\varepsilon}{1-\varepsilon}\right)\left(1 + \frac{2(1+\varepsilon)}{1-\varepsilon}\right).$$

Proof. This follows directly from Propositions E.4 and E.5. □

Then, we take a union bound to show the approximate triangle inequality holds globally for all pairs with high probability.

Proposition E.7 (Restatement of Proposition 5.1)**.** *Fix* $\varepsilon > 0$. *An approximate triangle inequality* $\bar{d}_{uv} \leq c_3(\varepsilon) \cdot (\bar{d}_{uw} + \bar{d}_{vw}) + h_3(\varepsilon)$ *holds for all triplets* u, v, w ***simultaneously*** *with probability at least* $1 - O\left(\frac{1}{n}\right)$.

Proof. The triangle inequality fails for an arbitrary triplet u, v, w with probability at most $\frac{12}{n^{\varepsilon^2 C(\varepsilon)/8}}$ by Proposition E.6. As there are at most n^3 triplets, the triangle inequality fails on at least one triplet with probability at most $\frac{12}{n^{\varepsilon^2 \frac{C(\varepsilon)}{8} - 3}} = \frac{12}{n}$ since $C(\varepsilon) = 32/\varepsilon^2$. □

E.2. Proofs: post-processed estimates for the correlation metric

Proof of Proposition 5.2. By Proposition 5.1,

$$\bar{d}_{uv} \leq c_3(\varepsilon) \cdot (\bar{d}_{uw} + \bar{d}_{vw}) + h_3(\varepsilon) \tag{24}$$

holds simultaneously for all triplets u, v, w w.h.p. We need to show that a similar inequality holds when we replace the intermediate estimates $\bar{d}(\cdot)$ with the post-processed estimates $\widetilde{d}(\cdot)$. First observe that we round $\bar{d}_{uv}$ up to 1 only when $\bar{d}_{uv} \geq 1/(2c_3(\varepsilon) + 1)$. So in this case

$$\widetilde{d}_{uv} = 1 = (1/(2c_3(\varepsilon) + 1))\,(2c_3(\varepsilon) + 1) \leq \bar{d}_{uv} \cdot (2c_3(\varepsilon) + 1).$$

So it always true that

$$\widetilde{d}_{uv} \leq \bar{d}_{uv} \cdot (2c_3(\varepsilon) + 1) \tag{25}$$

since this inequality also holds for $\widetilde{d}_{uv} = 0$ and for $\widetilde{d}_{uv} = \bar{d}_{uv}$ (using $\bar{d}_{uv} \geq 0$ by Fact 3). Next observe that we round $\bar{d}_{uw}$ down to 0 only when $\bar{d}_{uw} \leq 2h_3(\varepsilon)$. So it is always true that

$$\bar{d}_{uw} - 2h_3(\varepsilon) \leq \widetilde{d}_{uw}. \tag{26}$$

This is because (1) if $\bar{d}_{uw}$ is rounded up to $\widetilde{d}_{uw} = 1$, then the inequality holds since $\bar{d}_{uw} \leq 1$ (Fact 3), (2) if $\widetilde{d}_{uw} = \bar{d}_{uw}$, then the inequality holds automatically, and (3) if $\bar{d}_{uw}$ is rounded down to $\widetilde{d}_{uw} = 0$ then $\bar{d}_{uw} \leq 2h_3(\varepsilon)$, so again the inequality holds.

Putting inequalities (24), (25), and (26) together gives:

$$\begin{aligned}
\widetilde{d}_{uv} &\leq (2c_3(\varepsilon) + 1)\bar{d}_{uv} \\
&\leq (2c_3(\varepsilon) + 1)c_3(\varepsilon)(\bar{d}_{uw} + \bar{d}_{vw}) + (2c_3(\varepsilon) + 1)h_3(\varepsilon) \\
&\leq (2c_3(\varepsilon) + 1)c_3(\varepsilon)\left(\widetilde{d}_{uw} + \widetilde{d}_{vw} + 4h_3(\varepsilon)\right) + (2c_3(\varepsilon) + 1)h_3(\varepsilon) \\
&\leq c_4(\varepsilon)\left(\widetilde{d}_{uw} + \widetilde{d}_{vw}\right) + h_4(\varepsilon)
\end{aligned}$$

where $c_4(\varepsilon) = (2c_3(\varepsilon) + 1)c_3(\varepsilon)$ and $h_4(\varepsilon) = (4c_3(\varepsilon) + 1)(2c_3(\varepsilon) + 1)h_3(\varepsilon)$. In particular, $c_4(\varepsilon) \to 3$ and $h_4(\varepsilon) \to 0$ as $\varepsilon \to 0$.

□

Proof of Proposition 5.3. It suffices to show the bound holds pointwise, e.g., for $v \in N_u^+$, we show that $\widetilde{d}_{uv} \leq D(\varepsilon) \cdot d_{uv}$ and for $v \in N_u^-$, we show that $(1 - \widetilde{d}_{uv}) \leq D(\varepsilon)(1 - d_{uv})$.

First consider $v \in N_u^+$. If $\bar{d}_{uv}$ was rounded down to $\widetilde{d}_{uv} = 0$, then $\widetilde{d}_{uv} \leq D(\varepsilon)d_{uv}$ holds automatically. If $\bar{d}_{uv}$ was not rounded down, then $\widetilde{d}_{uv} = \bar{d}_{uv} \geq 2h_3(\varepsilon)$. By Proposition E.4, w.h.p.

$$2h_3(\varepsilon) \leq \widetilde{d}_{uv} = \bar{d}_{uv} \leq c_3(\varepsilon)d_{uv} + h_3(\varepsilon)$$

where we have used that $c_1(\varepsilon) \leq c_3(\varepsilon)$ and $h_1(\varepsilon) \leq h_3(\varepsilon)$. So $d_{uv} \geq h_3(\varepsilon)/c_3(\varepsilon)$. In turn,

$$\widetilde{d}_{uv} = \bar{d}_{uv} \leq c_3(\varepsilon)d_{uv} + h_3(\varepsilon) \leq c_3(\varepsilon)d_{uv} + c_3(\varepsilon)d_{uv} \leq D(\varepsilon) \cdot d_{uv}$$

as desired.

Next consider when $v \in N_u^-$. This case will be more involved. If $\bar{d}_{uv}$ was rounded up to $\widetilde{d}_{uv} = 1$, then $0 = 1 - \widetilde{d}_{uv} \leq D(\varepsilon) \cdot (1 - d_{uv})$ holds automatically. Otherwise, $\widetilde{d}_{uv} = \bar{d}_{uv} \leq 1/(2c_3(\varepsilon) + 1)$. We consider two cases:

Case 1. $h_3(\varepsilon) \leq \bar{d}_{uv}$.

By Proposition E.5, w.h.p.

$$c_3(\varepsilon) \cdot \bar{d}_{uv} + h_3(\varepsilon) \geq d_{uv}$$

where we have used that $c_2(\varepsilon) \leq c_3(\varepsilon)$ and $h_2(\varepsilon) \leq h_3(\varepsilon)$. Now using the assumptions of this case,

$$\begin{aligned}
\bar{d}_{uv} &\geq \frac{1}{c_3(\varepsilon)} \cdot d_{uv} - \frac{h_3(\varepsilon)}{c_3(\varepsilon)} \\
\bar{d}_{uv} &\geq \frac{1}{c_3(\varepsilon)} \cdot d_{uv} - \frac{\bar{d}_{uv}}{c_3(\varepsilon)} \\
\left(1 + \frac{1}{c_3(\varepsilon)}\right) \cdot \bar{d}_{uv} &\geq \frac{1}{c_3(\varepsilon)} \cdot d_{uv} \\
\bar{d}_{uv} &\geq \frac{1}{c_3(\varepsilon) + 1} \cdot d_{uv} \qquad (27)
\end{aligned}$$

Since $\bar{d}_{uv} \leq 1/(2c_3(\varepsilon) + 1)$, it holds that

$$1 - \bar{d}_{uv} \leq 2\left(1 - (c_3(\varepsilon) + 1)\bar{d}_{uv}\right).$$

Now, using (27),

$$1 - \widetilde{d}_{uv} = 1 - \bar{d}_{uv} \leq 2(1 - d_{uv}) \leq D(\varepsilon) \cdot (1 - d_{uv})$$

as desired.

Case 2. $h_3(\varepsilon) \geq \bar{d}_{uv}$.

As in the previous case we use Proposition E.5. This, along with the assumptions of this case gives, w.h.p.

$$\begin{aligned}
d_{uv} &\leq c_3(\varepsilon) \cdot \bar{d}_{uv} + h_3(\varepsilon) \\
d_{uv} &\leq (c_3(\varepsilon) + 1)h_3(\varepsilon) \\
1 - d_{uv} &\geq 1 - (c_3(\varepsilon) + 1)h_3(\varepsilon).
\end{aligned}$$

Now since $c_3(\varepsilon) \to 1$ and $h_3(\varepsilon) \to 0$, we have that for small enough ε ($\varepsilon < 0.3$ suffices), $(c_3(\varepsilon)+1)h_3(\varepsilon) \leq 1/2$. In turn,

$$\begin{aligned}
1 - d_{uv} &\geq 1/2 \\
1 - d_{uv} &\geq 1/2(1 - \bar{d}_{uv})
\end{aligned}$$

where we have used that $\bar{d}_{uv} \geq 0$ (Fact 3). So

$$1 - \widetilde{d}_{uv} = 1 - \bar{d}_{uv} \leq 2(1 - d_{uv}) \leq D(\varepsilon) \cdot (1 - d_{uv})$$

which concludes the case and the proof. □

E.3. Proof: approximate triangle inequality suffices

In this section, we show that instead of inputting a semi-metric to the rounding algorithm (Algorithm A), one can use as input a function that is almost a semi-metric. We will call such a function d a (δ_1, δ_2)-*semi-metric* if it is a semi-metric except instead of satisfying the triangle inequality, it satisfies a (δ_1, δ_2)-approximate triangle inequality:

Definition E.8. The function $d : V^2 \to \mathbb{R}_{\geq 0}$ satisfies a (δ_1, δ_2)-*approximate triangle inequality* when

$$d_{uv} \leq \delta_1(d_{uw} + d_{wv}) + \delta_2 \quad \forall u, v, w \in V.$$

Algorithm E.3

Input: d a (δ_1, δ_2)-semi-metric on V **Output:** Clustering $\mathcal{C}$.

1. Let $V_0 = V$, $r = r(\delta_1, \delta_2)$, $t = 0$.
2. **while** ($V_t \neq \emptyset$)
   - Find $w_t = \arg\max_{w \in V_t} L_t(w) = \arg\max_{w \in V_t} \sum_{u \in \text{Ball}(w,r) \cap V_t} r - d_{uw}$.
   - Create $C_t = \text{Ball}(w_t, b \cdot r) \cap V_t$, for $b = b(\delta_1, \delta_2)$.
   - Set $V_{t+1} = V_t \setminus C_t$ and $t = t + 1$.
3. Return $\mathcal{C} = (C_0, \ldots, C_{t-1})$.

Recall as in Section 3 that $\widehat{d}_{uv} = d_{uv}$ if $(u, v) \in E^+$, $\widehat{d}_{uv} = 1 - d_{uv}$ if $(u, v) \in E^-$, and $\mathsf{ALG}(u, v) = \mathbf{1}((u, v)$ is in disagreement in $\mathcal{C})$, for $\mathcal{C}$ the clustering returned by Algorithm E.3.

Let $\varepsilon > 0$ be sufficiently small. By Proposition 5.2, $\widetilde{d}$ satisfies a (δ_1, δ_2)-approximate triangle inequality, where $\delta_1 = \delta_1(\varepsilon) = 3 + h_4(\varepsilon)$ and $\delta_2 = \delta_2(\varepsilon) = h_4(\varepsilon)$. (This follows by noting that $c_4(\varepsilon) \leq 3 + h_4(\varepsilon)$. Thus the (δ_1, δ_2)-approximate triangle inequality is weaker than that in Proposition 5.2, but we use the former for ease of computation.)

Lemma E.9 (Analogue of Theorem B.1 in (Kalhan et al., 2019))**.** *Let $r = r(\delta_1, \delta_2)$ and $b = b(\delta_1, \delta_2)$ be as defined in Appendix E.4.*[9] *Let $\mathcal{C}$ be a clustering returned by Algorithm E.3. Then,*

$$\mathsf{ALG}(u) = \sum_v \mathsf{ALG}(u, v) \leq \frac{1}{r(\delta_1, \delta_2)} \sum_v \widehat{d}_{uv}$$

where $r(\delta_1, \delta_2) \to 1/121$.[10]

Proof of Lemma E.9. We will follow the proof of Theorem B.1. However, the change in the parameter r and in how C_t is created (i.e., the radius to $b \cdot r$ instead of $2r$) will cause the cases to split at different points. Note that $b \cdot r < 1$.

Define $\text{profit}(u) = \sum_v \widehat{d}_{uv} - r \sum_v \mathsf{ALG}(u, v)$ and $\Delta E_t = \{(u, v) \mid u \in C_t \text{ or } v \in C_t\}$. Then,

$$\text{profit}_t(u, v) = \begin{cases} \widehat{d}_{uv} - r\mathsf{ALG}(u, v) & (u, v) \in \Delta E_t \\ 0 & o.w. \end{cases}$$

and

$$\text{profit}_t(u) = \sum_{v \in V_t} \text{profit}_t(u, v) = \sum_{(u,v) \in \Delta E_t, v \in V_t} \widehat{d}_{uv} - r \sum_{(u,v) \in \Delta E_t, v \in V_t} \mathsf{ALG}(u, v).$$

Observe that since ΔE_t are disjoint, $\text{profit}(u) = \sum_t \text{profit}_t(u)$. Note that if $u \notin V_t$, $\text{profit}_t(u) = 0$.

Lemma E.10 (Analogue of Lemma B.2 in (Kalhan et al., 2019))**.** *For every $u \in V_t$, $\text{profit}_t(u) \geq 0$. Consequently, $\text{profit}(u) \geq 0$, giving Lemma E.9.*

Take $c_1 = c_1(\delta_1, \delta_2)$ and $c_2 = c_2(\delta_1, \delta_2)$ as defined in Appendix E.4. Note $c_2 \cdot r < 1$ and $c_1 < c_2$, so the cases in the proof of Lemma E.10 make sense.

[9] For us, $\delta_1(\varepsilon) \to 3$ and $\delta_2(\varepsilon) \to 0$ as $\varepsilon \to 0$. A similar result can be obtained when δ_1 approaches an arbitrary constant, as long as $\delta_2(\varepsilon) \to 0$ as $\varepsilon \to 0$.

[10] If $\delta_1(\varepsilon) \to 1$, then $r(\delta_1, \delta_2) \to 1/5$ as $\varepsilon \to 0$, recovering the rounding result that holds when the exact triangle inequality is satisfied.

Proof. Let $w = w_t$ be the maximizer of L_t.

Case 3. ($d_{uw} \in [0, c_1 \cdot r] \cup [c_2 \cdot r, 1]$)

In this case, it suffice to show that $\text{profit}_t(u, v) \geq 0$ for $v \in V_t$ and $(u, v) \in \Delta E_t$. This follows by Claim 3 and Lemmas E.11 and E.12.

Case 4. ($d_{uw} \in (c_1 \cdot r, c_2 \cdot r)$)

By Lemmas E.13 and E.14, $\text{profit}_t(u) \geq L_t(w) - L_t(u) \geq 0$, since w is the maximizer of L_t.

□

Claim 3 (Analogue of Claim B.3 in (Kalhan et al., 2019)). Let $u, v \in V_t$ and $(u, v) \in E^-$. Then $\text{profit}_t(u, v) \geq 0$.

Proof. Note the claim holds automatically if $\mathsf{ALG}(u, v) = 0$ or $(u, v) \notin \Delta E_t$. So we consider when $\mathsf{ALG}(u, v) = 1$ and $(u, v) \in \Delta E_t$. In this case, it must be that both u and v are in C_t, since $(u, v) \in E^- \cap \Delta E_t$. By definition of C_t, $d_{uw} \leq b \cdot r$ and $d_{vw} \leq b \cdot r$. By the approximate triangle inequality, $d_{uv} \leq 2\delta_1 b \cdot r + \delta_2$. This gives

$$\text{profit}_t(u, v) = \widehat{d}_{uv} - r \cdot \mathsf{ALG}(u, v) = 1 - d_{uv} - r \geq 1 - 2\delta_1 b \cdot r - \delta_2 - r \geq 0,$$

where the last inequality follows by the choice of constants. □

Lemma E.11 (Analogue of Lemma B.4 in (Kalhan et al., 2019))**.** *Let $u \in V_t$ and $d_{uw} \in [0, c_1 \cdot r]$. Then $\text{profit}_t(u, v) \geq 0$ for all $v \in V_t$ and $(u, v) \in \Delta E_t$.*

Proof. Due to Claim 3, we may assume $(u, v) \in E^+$. Since $d_{uw} \leq c_1 \cdot r$ and $c_1 \leq b$, we have $u \in \text{Ball}(w, b \cdot r)$, thus $u \in C_t$. So $(u, v) \in E^+$ is a disagreement if and only if $v \notin C_t$, i.e., if and only if $d_{vw} > b \cdot r$. By our choice of constants

$$\begin{aligned}\text{profit}_t(u, v) &\geq \widehat{d}_{uv} - r\mathsf{ALG}(u, v) \geq d_{uv} - r \geq \frac{1}{\delta_1} d_{vw} - d_{uw} - r - \delta_2/\delta_1 \\ &\geq \frac{1}{\delta_1} b \cdot r - c_1 \cdot r - r - \delta_2/\delta_1 \geq 0.\end{aligned}$$

□

Lemma E.12 (Analogue of Lemma B.5 in (Kalhan et al., 2019))**.** *Let $u \in V_t$ and $d_{uw} \in [c_2 \cdot r, 1]$. Then $\text{profit}_t(u, v) \geq 0$ for all $v \in V_t$ and $(u, v) \in \Delta E_t$.*

Proof. Due to Claim 3, we may assume $(u, v) \in E^+$. Since $d_{uw} \geq c_2 \cdot r$ and $c_2 > b$, we have $u \notin \text{Ball}(w, b \cdot r) \cap V_t = C_t$. In turn $(u, v) \in E^+$ is a disagreement if and only if $v \in C_t$ (note $(u, v) \in \Delta E_t$ by assumption), i.e., $d_{vw} \leq b \cdot r$. By the choice of constants, this gives

$$\text{profit}_t(u, v) = d_{uv} - r \geq \frac{1}{\delta_1} d_{uw} - d_{vw} - r - \delta_2/\delta_1 \geq \frac{1}{\delta_1} c_2 \cdot r - b \cdot r - r - \delta_2/\delta_1 \geq 0.$$

□

Claim 4 (Analogue of Claim B.6 in (Kalhan et al., 2019)). Let $u \in V_t$, $d_{uw} \in (c_1 \cdot r, c_2 \cdot r]$, and $v \in \text{Ball}(w, r) \cap V_t$. Then $\text{profit}_t(u, v) \geq r - d_{vw}$.

Proof.

Case 5. $d_{uw} \in (c_1 \cdot r, b \cdot r]$.

In this case, $u \in C_t$, since $d_{uw} \leq b \cdot r$. Also, since $v \in \text{Ball}(w, r) \cap V_t$ and $b \geq 1$, we have $v \in \text{Ball}(w, b \cdot r) \cap V_t$, so $v \in C_t$ as well. Thus, if $(u, v) \in E^+$, $ALG(u, v) = 0$, and by the choice of constants

$$\text{profit}_t(u, v) = \widehat{d}_{uv} - r\mathsf{ALG}(u, v) = d_{uv} - 0 \geq \frac{1}{\delta_1} d_{uw} - d_{vw} - \delta_2/\delta_1$$

$$\geq \frac{1}{\delta_1} c_1 \cdot r - d_{vw} - \delta_2/\delta_1 = r - d_{vw}.$$

On the other hand, if $(u, v) \in E^-$, then

$$\begin{aligned}\text{profit}_t(u,v) &= \widehat{d}(u,v) - r \cdot \mathsf{ALG}(u,v) \geq 1 - d_{uv} - r \geq 1 - \delta_1 d_{uw} - \delta_1 d_{vw} - r - \delta_2 \\ &= 1 - \delta_1 d_{uw} - r - (\delta_1 - 1) d_{vw} - d_{vw} - \delta_2 \\ &= 1 - (\delta_1 b + \delta_1) r - d_{vw} - \delta_2 \geq r - d_{vw}\end{aligned}$$

Case 6. $d_{uw} \in (b \cdot r, c_2 \cdot r]$.

If $(u, v) \in E^+$, then

$$\begin{aligned}\text{profit}_t(u,v) &= \widehat{d}_{uv} - r\mathsf{ALG}(u,v) \geq d_{uv} - r \geq \frac{1}{\delta_1} d_{uw} - d_{vw} - r - \delta_2/\delta_1 \\ &= (\frac{1}{\delta_1} b - 1) r - d_{vw} - \delta_2/\delta_1 \geq r - d_{vw}.\end{aligned}$$

If $(u, v) \in E^-$, then

$$\begin{aligned}\text{profit}_t(u,v) &= \widehat{d}_{uv} - r\mathsf{ALG}(u,v) \geq 1 - d_{uv} - r \geq 1 - \delta_1 d_{uw} - \delta_1 d_{vw} - r - \delta_2 \\ &\geq 1 - (\delta_1 c_2 + \delta_1) r - d_{vw} - \delta_2 \geq r - d_{vw},\end{aligned}$$

where we have used that $\delta_1 \geq 3$ implies $1 - \delta_1 < 0$. □

Lemma E.13 (Analogue of Lemma B.7 in (Kalhan et al., 2019))**.** *If $d_{uw} \in (c_1 \cdot r, c_2 \cdot r]$, then $P_{high}(u) \geq L_t(w)$.*

Proof. By Claim 4, $\text{profit}_t(u, v) \geq r - d_{vw}$ for all $v \in \text{Ball}(w, r) \cap V_t$. So

$$P_{high}(u) = \sum_{v \in \text{Ball}(w,r) \cap V_t} \text{profit}_t(u,v) \geq \sum_{v \in \text{Ball}(w,r) \cap V_t} r - d_{vw} = L_t(w).$$

□

Claim 5 (Analogue of Claim B.8 in (Kalhan et al., 2019)). Let $u, v \in V_t$. Then $\text{profit}_t(u, v) \geq \min(d_{uv} - r, 0)$.

Proof. If $(u, v) \in E^-$, then $\text{profit}_t(u, v) \geq 0$ by Claim 3. If $(u, v) \notin \Delta E_t$, then $\text{profit}_t(u, v) = 0$ by definition. So we may assume $(u, v) \in E^+ \cap \Delta E_t$. In this case,

$$\text{profit}_t(u,v) = \widehat{d}_{uv} - r\mathsf{ALG}(u,v) \geq d_{uv} - r \geq \min(d_{uv} - r, 0).$$

□

Lemma E.14 (Analogue of Lemma B.9 in (Kalhan et al., 2019))**.** *Let $u \in V_t$. Then $P_{low}(u) \geq -L_t(u)$, where $P_{low}(u) = \sum_{v \in V_t \setminus \text{Ball}(w,r)} \text{profit}_t(u, v)$.*

Proof.

$$\begin{aligned}P_{low}(u) &= \sum_{v \in V_t \setminus \text{Ball}(w,r)} \text{profit}_t(u,v) \geq \sum_{v \in V_t \setminus \text{Ball}(w,r)} \min(d_{uv} - r, 0) \\ &\geq \sum_{v \in V_t} \min(d_{uv} - r, 0) = \sum_{v \in \text{Ball}(u,r) \cap V_t} d_{uv} - r = -L_t(u)\end{aligned}$$

where in the second line we have used Claim 5, in the third line we have used that all terms are non-positive, and in the fourth line we have used that $\min(d_{uv} - r, 0) = 0$ if $v \notin \text{Ball}(u, r)$. □

□

E.4. Choices of constants in Section E.3

Let $\varepsilon > 0$ be sufficiently small, $\delta_1 = 3 + h_4(\varepsilon)$, and $\delta_2 = h_4(\varepsilon)$, where $h_4(\varepsilon)$ is as in Propositon 5.2. Note that $h_4(\varepsilon) \to 0$ as $\varepsilon \to 0$. Take

$$\begin{aligned} r &= \frac{1 - \delta_2 - \delta_1\delta_2 - \delta_1^3\delta_2 - \delta_1^2\delta_2}{\delta_1^2 + \delta_1^3(\delta_1 + 1) + \delta_1 + 1} \\ c_1 &= \delta_1 + \frac{\delta_2}{r} \\ b &= (c_1 + 1)\delta_1 + \frac{\delta_2}{r} \\ c_2 &= \delta_1(b + 1) + \frac{\delta_2}{r}. \end{aligned}$$

One can verify that the following inequalities, as needed in Section E.3, hold.

$$\begin{aligned} & b \geq 1 && c_2 \cdot r < 1 \\ & c_1 \leq b < c_2 && 1 - 2\delta_1 b \cdot r - \delta_2 - r \geq 0 \\ & \frac{1}{\delta_1} b \cdot r - c_1 \cdot r - r - \delta_2/\delta_1 \geq 0 && \frac{1}{\delta_1} c_2 \cdot r - b \cdot r - r - \delta_2/\delta_1 \geq 0 \\ & \frac{1}{\delta_1} c_1 \cdot r - \delta_2/\delta_1 \geq r && 1 - (\delta_1 b + \delta_1) r - \delta_2 \geq r \\ & (\frac{1}{\delta_1} b - 1) r - \delta_2/\delta_1 \geq r && 1 - (\delta_1 c_2 + \delta_1) r - \delta_2 \geq r \end{aligned}$$

E.5. Proof of Theorem 1.3

Proof of Theorem 1.3. Our sampling algorithm for correlation clustering with respect to the ℓ_∞ norm is the following. Compute $\widetilde{d}$ via sampling as described in Section 5.2. Feed $\widetilde{d}$ as input to Algorithm A. Let $\mathsf{ALG}(u, v) = \mathbb{1}((u, v)$ is in disagreement in $\mathcal{C})$ and $\mathsf{ALG}(u) = \sum_{v \in V} \mathsf{ALG}(u, v)$. Define $\widehat{\widetilde{d}}_{uv} = \widetilde{d}_{uv}$ if $v \in N_u^+$ and $\widehat{\widetilde{d}}_{uv} = 1 - \widetilde{d}_{uv}$ if $u \in N_v^-$. Following line (11), we see that, with probability $1 - O(1/n)$,

$$\mathsf{ALG}(u) = \sum_{v \in V} \mathsf{ALG}(u, v) \underset{*}{\leq} \frac{1}{r(\varepsilon)} \cdot \sum_{v \in V} \widehat{\widetilde{d}}_{uv} \underset{**}{\leq} \frac{1}{r(\varepsilon)} \cdot D(\varepsilon) \cdot \mathsf{OPT}. \tag{28}$$

where $D(\varepsilon) = 2 \cdot \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^2$ and $r(\varepsilon) = r(\delta_1, \delta_2)$ (recall that $\delta_1 = \delta_1(\varepsilon)$ and $\delta_2 = \delta_2(\varepsilon)$), as defined in Appendix E.4. The inequality $(*)$ follows from Proposition 5.2 and Lemma E.9, and the inequality $(**)$ follows from Proposition 5.3. The theorem statement is obtained by taking

$$c(\varepsilon) = \frac{1}{r(\varepsilon)} \cdot D(\varepsilon) = \frac{(3 + h_4(\varepsilon))^2 + (3 + h_4(\varepsilon))^3(4 + h_4(\varepsilon)) + 4 + h_4(\varepsilon)}{1 - h_4(\varepsilon) - (3 + h_4(\varepsilon))h_4(\varepsilon) - (3 + h_4(\varepsilon))^3 h_4(\varepsilon) - (3 + h_4(\varepsilon))^2 h_4(\varepsilon)} \cdot 2\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^2,$$

where $h_4(\varepsilon) = \left(4\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^2 + 1\right)\left(2\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^2 + 1\right)\left(\frac{2\varepsilon}{1-\varepsilon}\right)\left(1 + \frac{2(1+\varepsilon)}{1-\varepsilon}\right)$.

Next, we analyze the run-time. As before, there are two phases: (1) computing the estimates $\widetilde{d}_{uv}$, and (2) using the rounding algorithm (Algorithm A) with input $\widetilde{d}$. Phase (1) takes $O(n^2 \log n/\varepsilon^2)$ time. First we compute the sample S_u for each vertex u, which takes $O(n \log n/\varepsilon^2)$ time, since we sample $O(\log n/\varepsilon^2)$ vertices from N_u^+. Then, for each of the $O(n^2)$ pairs u, v, we compute $W^{(u,v)}$ and $Y^{(u,v)}$, which takes $O(\log n/\varepsilon^2)$ time. Thus to compute $\bar{d}_{uv}$ as in (15) for all pairs takes $O(n^2 \log n/\varepsilon^2)$ time. Finally, obtaining $\widetilde{d}$ from $\bar{d}$ via rounding takes $O(n^2)$ time. Phase (2) takes $O(n^2)$ time, by the discussion in Appendix A. Together the two phases contribute a total run-time of $O(n^2 \log n/\varepsilon^2)$, completing the proof of Theorem 1.3. □

F. Supplementary Experiment Information

F.1. Description of Pivot algorithm

The Pivot algorithm of Ailon, Charikar, and Newman is a randomized algorithm that gives a 3-approximation in expectation for classic correlation clustering (i.e., ℓ_1 norm) on a complete graph (Ailon et al., 2008). As mentioned in Section 1.2, Pivot may perform poorly for the Min Max objective. The Pivot algorithm is as follows. Sample a uniformly random ordering of the vertices. Visit vertices in this order. Upon visiting a vertex, check whether it has been marked as clustered. If it has, visit the next vertex. If not, call the current vertex a pivot, and open a new cluster consisting of the pivot and all unclustered vertices that are in the positive neighborhood of the pivot. Mark these vertices as clustered.

F.2. Additional plots for Section 6

This section contains additional plots for the experiments discussed in Section 6.

	FB 348	FB 414	FB 686	FB 698	FB 3980
#vertices	224	150	168	61	52
#edges	6384	3386	3312	540	292
max positive degree	100	58	78	30	19

Table 4. Graph statistics for the five small Facebook datasets in Table 1.

	FB 0	FB 107	FB 1684	FB 1912	FB 3437
#vertices	333	1034	786	747	534
#edges	5038	53498	28048	60050	9626
max positive degree	78	254	137	294	108

Table 5. Graph statistics for the five large Facebook datasets in Table 2.

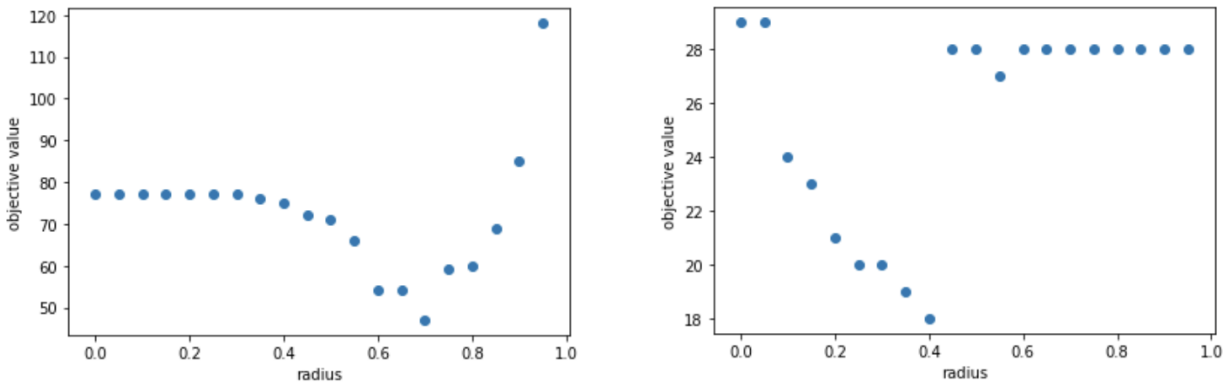


Figure 3. Plots showing how the objective value changes for our algorithm (left) and the KMZ algorithm (right) as we sweep over a common radius $r_1 = r_2$ for datasets FB 686 (left) and FB 698 (right). Plots for other datasets are similar, so we used radii of 0.7 and 0.4 for our algorithm and the KMZ algorithm, respectively, in Tables 1 and 2.

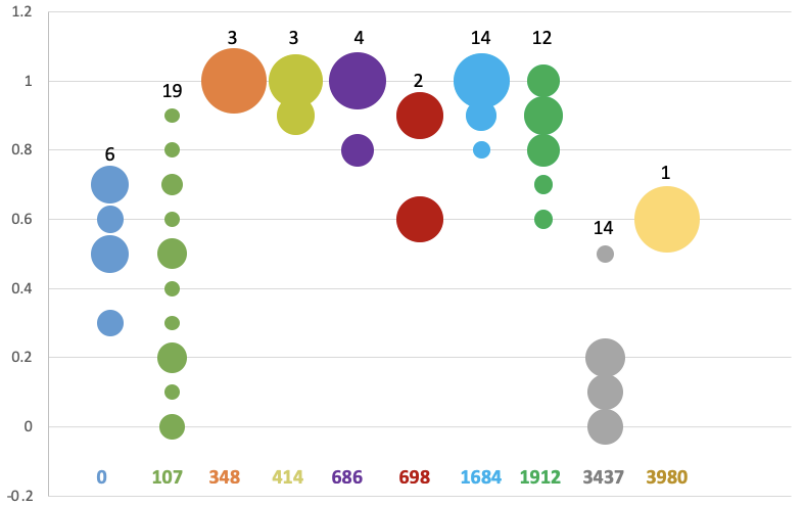


Figure 4. For each dataset (bottom), the bubbles above it quantify the extent to which large clusters (size ≥ 10) found by our algorithm are contained in ground truth clusters. The number of large clusters for each dataset is at the top of the column. Each bubble represents a certain number of large clusters for the corresponding dataset; the size of the bubble is proportional to how many large clusters it represents. For each large cluster, we found its best ground truth cluster, i.e., the one containing the largest number of nodes from that cluster. The y-axis represents the proportion of a large cluster contained in its best ground truth cluster. For example, for dataset FB 698, consider a given large cluster that has between 90% (inclusive) and 100% (exclusive) of its nodes contained in its best ground truth cluster. It is accounted for by the red circle vertically positioned at 0.9. A large cluster that has between 60% and 70% of its nodes contained in its best ground truth cluster, is accounted for by the red circle vertically positioned at 0.6. Since the sizes of these two red circles are equal, half of the large clusters fall into each of these two categories. Notice that for datasets FB 348, 414, 686 1684, and 1912, there are bubbles located at the y-axis position of 1; these bubbles represent large clusters that are *completely* (i.e., 100%) contained in a ground truth cluster. For dataset FB 348 in particular, *every* large cluster is 100% contained in a ground truth cluster.

F.3. Results for double parameter sweep on Facebook datasets

For ease of discussion in Section 6 and Appendix F.2, we applied our algorithm and the KMZ algorithm to the Facebook datasets by enforcing a common radius $r = r_1 = r_2$. We used $r = 0.7$ for our algorithm and $r = 0.4$ for the KMZ algorithm on *all datasets*. In this section, we present a more tailored analysis, where for *each dataset* and *each algorithm*, we find the best r_1 and r_2 for that dataset (without requiring that $r_1 = r_2$). The results are reported in Tables 6 and 7. While the results are similar to those reported in Section 6 and Appendix F.2, we include these results for completeness.

	FB 348	FB 414	FB 686	FB 698	FB 3980
fractional cost	74.37	35.53	58.59	22.31	14.31
LP objective	39.13	19.66	30.48	10.64	7.34
our objective	71	31	43	18	12
KMZ objective	69	28	47	17	13
Pivot objective	85.03	50.73	65.72	23.51	16.36
(r_1, r_2)	(0.45, 0.7)	(0.1, 0.65)	(0.3, 0.75)	(0.8, 0.8)	(0.7, 0.7)
KMZ (r_1, r_2)	(0.2, 0.45)	(0.05, 0.6)	(0.3, 0.6)	(0.05, 0.5)	(0.3, 0.6)

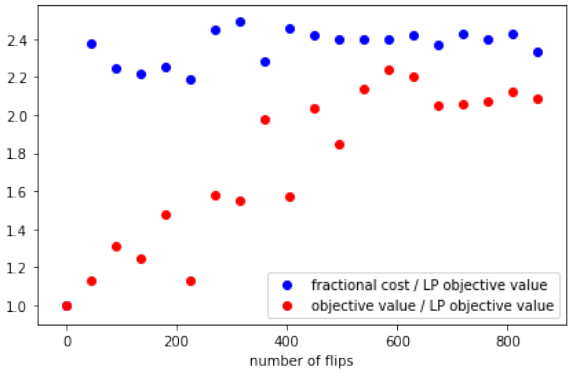
Table 6. Experimental results for the five small Facebook datasets. The pairs (r_1, r_2) and KMZ (r_1, r_2) refer to an optimal pair of radii in a double parameter sweep (instead of setting $r_1 = r_2$ as in Section E.3 and Appendix F), which were used in the rounding phase of each algorithm.

	FB 0	FB 107	FB 1684	FB 1912	FB 3437
fractional cost	64.02	181.49	103.99	227.74	98.36
our objective	49	134	93	187	77
Pivot objective	71.78	216.65	130.71	259.01	99.1
(r_1, r_2)	(0.7, 0.7)	(0.65, 0.65)	(0.7, 0.7)	(0.65, 0.65)	(0.3, 0.7)

Table 7. Experimental results for the five larger Facebook datasets. As in Table 6, (r_1, r_2) refers to the optimal radii in a double parameter sweep.

F.4. Synthetic data: perfect clusterings with noise

Below we show various plots for the synthetic experiments discussed in Section 6.



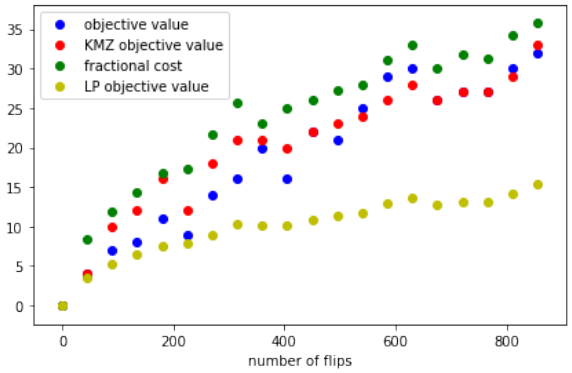


Figure 5. In the above plots, "fractional cost" refers to the fractional cost of our correlation metric. The term "objective value" refers to that of our algorithm, while "KMZ objective value" refers to that of the KMZ algorithm. These objective values are with respect to $r_1 = r_2 = 0.7$ for our algorithm and $r_1 = r_2 = 0.4$ for the KMZ algorithm.

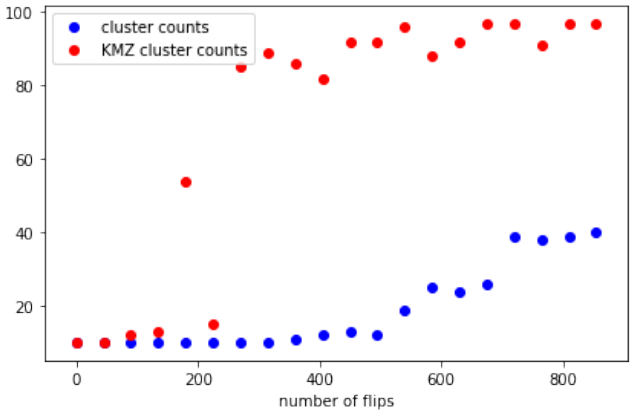


Figure 6. The term "cluster counts" refers to the number of clusters output by our algorithm, and "KMZ cluster counts" for the number output by the KMZ algorithm.

F.5. Scalability of our algorithm

We ran our exact algorithm (using the matrix multiplication implementation) on three large datasets with approximately 10,000 vertices each to show that our algorithm scales. The first dataset, denoted LastFM [11], is a social network of users of

[11] https://snap.stanford.edu/data/feather-lastfm-social.html

the music service LastFM in Asia, where vertices represent users and positive edges represent mutual follower relationship (Rozemberczki & Sarkar, 2020). The other two datasets, denoted ca-HepTh and ca-HepPh [12], are collaboration networks of high energy physics authors on arXiv, where vertices represent authors and positive edges represent co-authors (Leskovec et al., 2007). See Table 8 for our algorithm's run-time on these three datasets. We observe that the algorithm takes approximately two to four minutes on each dataset.

	our run-time	#vertices	#edges
LastFM	102.74	7624	27806
ca-HepTh	165.42	9877	51971
ca-HepPh	250.91	12008	237010

Table 8. Run-times (in seconds) of our exact algorithm on three large datasets, along with their sizes.

[12] https://snap.stanford.edu/data/ca-HepTh.html, https://snap.stanford.edu/data/ca-HepPh.html