Correlation Clustering and Biclustering with Locally Bounded Errors

Gregory J. Puleo PULEO@ILLINOIS.EDU
Coordinated Science Lab, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA

Olgica Milenkovic MILENKOV@ILLINOIS.EDU
Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA

Abstract

We consider a generalized version of the correlation clustering problem, defined as follows. Given a complete graph G whose edges are labeled with $+$ or $-$, we wish to partition the graph into clusters while trying to avoid errors: $+$ edges between clusters or $-$ edges within clusters. Classically, one seeks to minimize the total number of such errors. We introduce a new framework that allows the objective to be a more general function of the number of errors at each vertex (for example, we may wish to minimize the number of errors at the *worst* vertex) and provide a rounding algorithm which converts “fractional clusterings” into discrete clusterings while causing only a constant-factor blowup in the number of errors at each vertex. This rounding algorithm yields constant-factor approximation algorithms for the discrete problem under a wide variety of objective functions.

1. Introduction

Correlation clustering is a clustering model first introduced by Bansal, Blum, and Chawla (Bansal et al., 2002; 2004). The basic form of the model is as follows. We are given a collection of objects and, for some pairs of objects, we are given a judgment of whether the objects are *similar* or *dissimilar*. This information is represented as a labeled graph, with edges labeled $+$ or $-$ according to whether the endpoints are similar or dissimilar. Our goal is to cluster the graph so that $+$ edges tend to be within clusters and $-$ edges tend to go across clusters. The number of clusters is not specified in advance; determining the optimal number of clusters is instead part of the optimization problem.

Given a solution clustering, an *error* is a $+$ edge whose endpoints lie in different clusters or a $-$ edge whose endpoints lie in the same cluster. In the original formulation of the correlation clustering, the goal is to minimize the total number of errors; this formulation of the optimization problem is called MINDISAGREE. Finding an exact optimal solution is NP-hard even when the input graph is complete (Bansal et al., 2002; 2004). Furthermore, if the input graph is allowed to be arbitrary, the best known approximation ratio is $O(\log n)$, obtained by (Charikar et al., 2003; 2005; Demaine et al., 2006). Assuming the Unique Games Conjecture of Khot (Khot, 2002), no constant-factor approximation for MINDISAGREE on arbitrary graphs is possible; this follows from the results of (Chawla et al., 2006; Steurer & Vishnoi, 2009) concerning the minimum multicut problem and the connection between correlation clustering and minimum multicut described in (Charikar et al., 2003; 2005; Demaine et al., 2006).

Since theoretical barriers appear to preclude constant-factor approximations on arbitrary graphs, much research has focused on special graph classes such as complete graphs and complete bipartite graphs, which are the graph classes we consider here. Ailon, Charikar, and Newman (Ailon et al., 2005; 2008) gave a very simple randomized 3-approximation algorithm for MINDISAGREE on complete graphs. This algorithm was derandomized by van Zuylen and Williamson (van Zuylen & Williamson, 2009), and a parallel version of the algorithm was studied by Pan, Papailiopoulos, Recht, Ramchandran, and Jordan (Pan et al., 2015). More recently, a 2.06-approximation algorithm was announced by Chawla, Makarychev, Schramm and Yaroslavtsev (Chawla et al., 2014). Similar results have been obtained for complete bipartite graphs. The first constant approximation algorithm for correlation clustering on complete bipartite graphs was described by Amit (Amit, 2004), who gave an 11-approximation algorithm. This ratio was improved by Ailon, Avigdor-Elgrabli, Liberty and van Zuylen (Ailon et al., 2012), who obtained a 4-approximation algorithm. Chawla, Makarychev, Schramm and Yaroslavtsev (Chawla et al., 2014) announced a 3-approximation algorithm for correlation clustering on complete k-partite graphs, for arbitrary k, which includes the complete bipartite case. Bipartite clustering has also been studied, outside the

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

correlation-clustering context, by Lim, Chen, and Xu (Lim et al., 2015).

We depart from the classical correlation-clustering literature by considering a broader class of objective functions which also cater to the need of many community-detection applications in machine learning, social sciences, recommender systems and bioinformatics (Cheng & Church, 2000; Symeonidis et al., 2007; Kriegel et al., 2009). The technical details of this class of functions can be found in Section 2. As a representative example of this class, we introduce *minimax correlation clustering*.

In minimax clustering, rather than seeking to minimize the *total* number of errors, we instead seek to minimize the number of errors at the *worst-off vertex* in the clustering. Put more formally, if for a given clustering each vertex v has y_v incident edges that are errors, then we wish to find a clustering that minimizes $\max_v y_v$.

Minimax clustering, like classical correlation clustering, is NP-hard on complete graphs, as we prove in the extended version of this paper (Puleo & Milenkovic, 2016). To design approximation algorithms for minimax clustering, it is necessary to bound the growth of errors *locally* at each vertex when we round from a fractional clustering to a discrete clustering; this introduces new difficulties in the design and analysis of our rounding algorithm. These new technical difficulties cause the algorithm of (Ailon et al., 2005; 2008) to fail in the minimax context, and there is no obvious way to adapt that algorithm to this new context; this phenomenon is explored further in the extended version of this paper (Puleo & Milenkovic, 2016).

Minimax correlation clustering on graphs is relevant in detecting communities, such as gene, social network, or voter communities, in which no *antagonists* are allowed. Here, an antagonist refers to an entity that has properties inconsistent with a large number of members of the community. Alternatively, one may view the minimax constraint as enabling individual vertex quality control within the clusters, which is relevant in biclustering applications such as collaborative filtering for recommender systems, where minimum quality recommendations have to be ensured for each user in a given category. As an illustrative example, one may view a complete bipartite graph as a preference model in which nodes on the left represent viewers and nodes on the right represent movies. A positive edge between a user and a movie indicates that the viewer likes the movie, while a negative edge indicates that they do not like or have not seen the movie. We may be interested in finding communities of viewers for the purpose of providing them with joint recommendations. Using a minimax objective function here allows us to provide a uniform quality of recommendations, as we seek to minimize the number of errors for the user who suffers the most errors.

A minimax objective function for a graph partitioning problem different from correlation clustering was previously studied by (Bansal et al., 2011). In that paper, the problem under consideration was to split a graph into k roughly-equal-sized parts, minimizing the total number of edges leaving any part. Thus, the minimum in (Bansal et al., 2011) is being taken over the parts of the solution, rather than minimizing over vertices as we do here.

Another idea slightly similar to minimax clustering has previously appeared in the literature on fixed-parameter tractability of the CLUSTER EDITING problem, which is an equivalent formulation of Correlation Clustering. In particular, Komusiewicz and Uhlmann (Komusiewicz & Uhlmann, 2012) proved that the following problem is fixed-parameter tractable for the combined parameter (d, t):

> **(d, t)-Constrained-Cluster Editing**
> **Input:** A labeled complete graph G, a function $\tau : V(G) \to \{0, \ldots, t\}$, and nonnegative integers d and k.
> **Question:** Does G admit a clustering into at most d clusters with at most k errors such that every vertex v is incident to at most $\tau(v)$ errors?

(Here, we have translated their original formulation into the language of correlation clustering.) Komusiewicz and Uhlmann also obtained several NP-hardness results related to this formulation of the problem. While their work involves a notion of local errors for correlation clustering, their results are primarily focused on fixed-parameter tractability, rather than approximation algorithms, and are therefore largely orthogonal to the results of this paper.

The contributions of this paper are organized as follows. In Section 2, we introduce and formally express our framework for the generalized version of correlation clustering, which includes both classical clustering and minimax clustering as special cases. In Section 3, we give a rounding algorithm which allows the development of constant-factor approximation algorithms for the generalized clustering problem. In Section 4, we give a version of this rounding algorithm for complete bipartite graphs.

2. Framework and Formal Definitions

In this section, we formally set up the framework we will use for our broad class of correlation-clustering objective functions.

Definition 1. Let G be an edge-labeled graph. A *discrete clustering* (or just a *clustering*) of G is a partition of $V(G)$. A *fractional clustering* of G is a vector x indexed by $\binom{V(G)}{2}$ such that $x_{uv} \in [0, 1]$ for all $uv \in \binom{V(G)}{2}$ and such that $x_{vz} \leq x_{vw} + x_{wz}$ for all distinct $v, w, z \in V(G)$.

If x is a fractional clustering, we can view x_{uv} as a "distance" from u to v; the constraints $x_{vz} \leq x_{vw} + x_{wz}$ are therefore referred to as *triangle inequality* constraints. We also adopt the convention that $x_{uu} = 0$ for all u.

In the special case where all coordinates of x are 0 or 1, the triangle inequality constraints guarantee that the relation defined by $u \sim v$ iff $x_{uv} = 0$ is an equivalence relation. Such a vector x can therefore naturally be viewed as a discrete clustering, where the clusters are the equivalence classes under $\sim$. By viewing a discrete clustering as a fractional clustering with integer coordinates, we see that fractional clusterings are a continuous relaxation of discrete clusterings, which justifies the name. This gives a natural notion of the total weight of errors at a given vertex.

Definition 2. Let G be an edge-labeled complete graph, and let x be a fractional clustering of G. The *error vector* of x with respect to G, written $\mathrm{err}(x)$, is a real vector indexed by $V(G)$ whose coordinates are defined by

$$\mathrm{err}(x)_v = \sum_{w \in N^+(v)} x_{vw} + \sum_{w \in N^-(v)} (1 - x_{vw}).$$

If $\mathcal{C}$ is a clustering of G and $x^{\mathcal{C}}$ is the natural associated fractional clustering, we define $\mathrm{err}(\mathcal{C})$ as $\mathrm{err}(x^{\mathcal{C}})$.

We are now prepared to formally state the optimization problem we wish to solve. Let $\mathbb{R}^n_{\geq 0}$ denote the set of vectors in $\mathbb{R}^n$ with all coordinates nonnegative. Our problem is parameterized by a function $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}$.

> **f-Correlation Clustering**
> **Input:** A labeled graph G.
> **Output:** A clustering $\mathcal{C}$ of G.
> **Objective:** Minimize $f(\mathrm{err}(\mathcal{C}))$.

In order to approximate f-Correlation Clustering, we introduce a relaxed version of the problem.

> **Fractional f-Correlation Clustering**
> **Input:** A labeled graph G.
> **Output:** A fractional clustering x of G.
> **Objective:** Minimize $f(\mathrm{err}(x))$.

If f is convex on $\mathbb{R}^n_{\geq 0}$, then using standard techniques from convex optimization (Boyd & Vandenberghe, 2004), the Fractional f-Correlation Clustering problem can be approximately solved in polynomial time, as the composite function $f \circ \mathrm{err}$ is convex and the constraints defining a fractional clustering are linear inequalities in the variables x_e. When G is a complete graph, we then employ a rounding algorithm based on the algorithm of Charikar, Guruswami, and Wirth (Charikar et al., 2003; 2005) to transform the fractional clustering into a discrete clustering. Under rather modest conditions on f, we are able to obtain a constant-factor bound on the error growth, that is, we can produce a clustering $\mathcal{C}$ such that $f(\mathrm{err}(\mathcal{C})) \leq cf(\mathrm{err}(x))$, where c is a constant not depending on f or x. In particular, we require the following assumptions on f.

Assumption A. *We assume that $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}$ has the following properties.*

(1) $f(cy) \leq cf(y)$ for all $c \geq 0$ and all $y \in \mathbb{R}^n$, and

(2) If $y, z \in \mathbb{R}^n_{\geq 0}$ are vectors with $y_i \leq z_i$ for all i, then $f(y) \leq f(z)$.

Under Assumption A, the claim that $f(\mathrm{err}(\mathcal{C})) \leq cf(\mathrm{err}(x))$ follows if we can show that $\mathrm{err}(\mathcal{C})_v \leq c\,\mathrm{err}(x)_v$ for every vertex $v \in V(G)$. This is the property we prove for our rounding algorithms.

We will slightly abuse terminology by referring to the constant c as an *approximation ratio* for the rounding algorithm; this notation is motivated by the fact that when f is linear, the Fractional f-Correlation Clustering problem can be solved exactly in polynomial time, and applying a rounding algorithm with constant c to the fractional solution yields a c-approximation algorithm to the (discrete) f-Correlation Clustering problem. In contrast, when f is nonlinear, we may only be able to obtain a $(1 + \epsilon)$-approximation for the Fractional f-Correlation Clustering problem, in which case applying the rounding algorithm yields a $c(1 + \epsilon)$-approximation algorithm for the discrete problem.

A natural class of convex objective functions obeying Assumption A is the class of ℓ^p norms. For all $p \geq 1$, the ℓ^p-norm on $\mathbb{R}^n$ is defined by

$$\ell^p(x) = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}.$$

As p grows larger, the ℓ^p-norm puts more emphasis on the coordinates with larger absolute value. This justifies that definition of the ℓ^∞-norm as

$$\ell^\infty(x) = \max\{x_1, \ldots, x_n\}.$$

Classical correlation clustering is the case of f-Correlation Clustering where $f(x) = \frac{1}{n}\ell^1(x)$, while minimax correlation clustering is the case of f-Correlation Clustering where $f(x) = \ell^\infty(x)$.

Our emphasis on convex f is due to the fact that convex programming techniques allow the Fractional f-Correlation Clustering problem to be approximately solved in polynomial time when f is convex. However, the correctness of our rounding algorithm does not depend on the convexity of f, only on the properties listed in Assumption A. If f is nonconvex and obeys Assumption A, and we produce a "good" fractional clustering x by some means, then our algorithm still produces a discrete clustering $\mathcal{C}$ with $f(\mathrm{err}(\mathcal{C})) \leq cf(\mathrm{err}(x))$.

3. A Rounding Algorithm for Complete Graphs

We now describe a rounding algorithm to transform an arbitrary fractional clustering x of a labeled complete graph G into a clustering $\mathcal{C}$ such that $\mathrm{err}(\mathcal{C})_v \leq c\,\mathrm{err}(x)_v$ for all $v \in V(G)$.

Our rounding algorithm is based on the algorithm of Charikar, Guruswami, and Wirth (Charikar et al., 2003; 2005) and is shown in Algorithm 1. The main difference between Algorithm 1 and the algorithm of (Charikar et al., 2003; 2005) is the new strategy of choosing a pivot vertex that maximizes $|T_u^*|$; in (Charikar et al., 2003; 2005), the pivot vertex is chosen arbitrarily. Furthermore, the algorithm of (Charikar et al., 2003; 2005) always uses $\alpha = 1/2$ as a cutoff for forming "candidate clusters", while we express α as a parameter which we later choose in order to optimize the approximation ratio.

Under the classical objective function, an optimal fractional clustering is the solution to a linear program, which motivates the following notation for the more general case.

Definition 3. If uv is an edge of a labeled graph G, we define the *LP-cost* of uv relative to a fractional clustering x to be x_{uv} if $uv \in E^+$, and $1 - x_{uv}$ if $uv \in E^-$. Likewise, the *cluster-cost* of an edge uv is 1 if uv is an error in the clustering produced by Algorithm 1, and 0 otherwise.

Our general strategy for obtaining the constant-factor error bound for Algorithm 1 is similar to that of (Charikar et al., 2003; 2005). Each time a cluster is output, we pay for the cluster-cost of the errors incurred by "charging" the cost of these errors to the LP-costs of the fractional clustering. The main difference between our proof and the proof of (Charikar et al., 2003; 2005) is that we must pay for errors *locally*: for each vertex v, we must pay for *all* clustering errors incident to v by charging to the LP cost incident to v. In particular, every clustering error must now be paid for at *each* of its endpoints, while in (Charikar et al., 2003; 2005), it was enough to pay for each clustering error at *one* of its endpoints. For edges which cross between a cluster and its complement, this requires a different analysis at each endpoint, a difficulty which was not present in (Charikar et al., 2003; 2005). Our proof emphasizes the solutions to these new technical problems; the parts of the proof that are technically nontrivial but follow earlier work are omitted due to space constraints but can be found in the extended version of this paper (Puleo & Milenkovic, 2016).

Observation 4. *Let x be a fractional clustering of a graph G, and let $w, z \in V(G)$. For any vertex u, we have $x_{wz} \geq x_{uz} - x_{uw}$ and $1 - x_{wz} \geq 1 - x_{uz} - x_{uw}$.*

Theorem 5. *Let G be a labeled complete graph, let α and γ be parameters with $0 < \gamma < \alpha < 1/2$, and let x be any fractional clustering of G. If $\mathcal{C}$ is the clustering produced by Algorithm 1 with the given input, then for all $v \in V(G)$ we have $\mathrm{err}(\mathcal{C})_v \leq c\,\mathrm{err}(x)_v$, where c is a constant depending only on α and γ.*

Proof. Let k_1, k_2, k_3 be constants to be determined, with $1/2 < k_1 < 1$ and $0 < 2k_2 \leq k_3 < 1/2$. Also assume that $k_1\alpha > \gamma$ and that $k_2\alpha \leq 1 - 2\alpha$.

To prove the approximation ratio, we consider the cluster-costs incurred as each cluster is output, splitting into cases according to the type of cluster. In our analysis, as the algorithm runs, we will mark certain vertices as "safe", representing the fact that some possible future clustering costs have been paid for in advance. Initially, no vertex is marked as safe.

Case 1: *A Type 1 cluster is output.* Let $X = S \cap N^+(u)$, with S as in Algorithm 1. The new cluster-cost incurred at u is $|X|$, and for each $v \in X$, a new cluster-cost of 1 is incurred at v.

First we pay for the new cluster cost incurred at u. For each edge uv with $v \in T$, we have $x_{uv} \leq \alpha$ and so $1 - x_{uv} \geq 1 - \alpha \geq x_{uv}$. Thus, the total LP cost of edges uv with $v \in T$ is at least $\sum_{v \in T} x_{uv}$, which is at least $\alpha\,|T|\,/2$ since $\{u\}$ is output as a Type 1 cluster. Thus, charging each edge uv with $v \in T$ a total of $2/\alpha$ times its LP-cost pays for the cluster-cost of any positive edges from u to T. On the other hand, if uv is a positive edge with $v \in S - T$, then since $v \notin T$, we have $x_{uv} \geq \alpha$. Hence, the LP-cost of uv is at least α, and charging $1/\alpha$ times the LP-cost of uv pays for the cluster-cost of this edge.

Now let $v \in X$; we must pay for the new cluster cost at v. If $x_{uv} \geq k_2\alpha$, then the edge uv already incurs LP cost at least $k_2\alpha$, so the new cost at v is only $1/(k_2\alpha)$ times the LP-cost of the edge uv. So assume $x_{uv} < k_2\alpha$. In this case, we say that u is a *bad pivot* for v.

First suppose that v is not safe (as is initially the case). We will make a single charge to the edges incident to v that is large enough to pay for both the edge uv and for all possible *future* bad pivots, and then we will mark v as safe to indicate that we have done this. The basic idea is that if v has many possible bad pivots, then since x_{uv} is "small", all of these possible bad pivots are also close to u, thus included in T_u. Since $\sum_{w \in T_u} x_{uw} \geq \alpha\,|T_u|\,/2$, there is a large set $B \subseteq T_u$ of vertices that are "moderately far" from u, and therefore moderately far from v. The number of these vertices grows with the number of bad pivots, so

Algorithm 1 Round fractional clustering x to obtain a discrete clustering, using threshold parameters α, γ with $0 < \gamma < \alpha < 1/2$.

```
Let $S = V(G)$.
while $S \neq \emptyset$ do
  For each $u \in S$, let $T_u = \{w \in S - \{u\}\colon\ x_{uw} \leq \alpha\}$ and let $T_u^* = \{w \in S - \{u\}\colon\ x_{uw} \leq \gamma\}$.
  Choose a pivot vertex $u \in S$ that maximizes $|T_u^*|$.
  Let $T = T_u$.
  if $\sum_{w \in T} x_{uw} \geq \alpha\,|T|\,/2$ then
    Output the cluster $\{u\}$. {Type 1 cluster}
    Let $S = S - \{u\}$.
  else
    Output the cluster $\{u\} \cup T$. {Type 2 cluster}
    Let $S = S - (\{u\} \cup T)$.
  end if
end while
```

charging all the edges vz for $z \in B$ is sufficient to pay for all bad pivots.

We now make this argument rigorous. Let P_v be the set of potential bad pivots for v, defined by

$$P_v = \{p \in S\colon\ x_{vp} < k_2\alpha\}.$$

Note that $u \in P_v$. Since $k_2 < 1/4$, we have $x_{up} \leq x_{uv} + x_{vp} < \alpha/2$ for all $p \in P_v$; hence $P_v \subseteq T$. Define the vertex set B by

$$B = \{z \in T\colon\ x_{uz} > k_3\alpha\}.$$

Since $x_{uz} \leq \alpha$ for all $z \in T$, we see that

$$\sum_{z \in T} x_{uz} \leq k_3\alpha\,|T - B| + \alpha\,|B|\,.$$

On the other hand, since $\{u\}$ is output as a Type 1 cluster, we have

$$\sum_{z \in T} x_{uz} \geq \alpha\,|T|\,/2.$$

Combining these inequalities and rearranging, we obtain $|B| \geq (1 - 2k_3)\,|T - B|$. For each vertex $z \in B$, we have $x_{vz} \geq x_{uz} - x_{uv} \geq (k_3 - k_2)\alpha$; in particular, since $k_3 \geq 2k_2$, we have $x_{vz} \geq k_2\alpha$, so that $z \notin P_v$. Hence $|T - B| \geq |P_v|$, and we have $|B| \geq (1 - 2k_3)\,|P_v|$.

On the other hand, for $z \in B$ we also have $1 - x_{vz} \geq 1 - x_{uv} - x_{uz} \geq 1 - (1 + k_2)\alpha$. It follows that each edge vz for $z \in B$ has LP-cost at least $\min((k_3 - k_2)\alpha,\ 1 - (1 + k_2)\alpha)$, independent of whether vz is positive or negative. It is easy to check that since $\alpha < 1/2$ and $k_3 < 1$, this minimum is always achieved by $(k_3 - k_2)\alpha$. Therefore, we can pay for the (possible) Type-1-cluster cost of all edges vp for $p \in P_v$ by charging each edge vz with $z \in B$ a total of

$$\frac{1}{(1 - 2k_3)(k_3 - k_2)\alpha}$$

times its LP-cost. We make all these charges when the cluster $\{u\}$ is created and put them in a "bank account" to pay for later Type-1-cluster costs for v. Then we mark v as safe. The total charge in the bank account is at least $|P_v|$, which is enough to pay for all bad pivots for v.

We have just described the case where u is a bad pivot and v is not safe. On the other hand, if u is a bad pivot and v is safe, then v already has a bank account large enough to pay for all its bad pivots, and we simply charge 1 to the account to pay for the edge uv.

Case 2: *A Type 2 cluster $\{u\} \cup T$ is output.* The negative edges within $\{u\} \cup T$ are easy to pay for: if vw if a negative edge inside $\{u\} \cup T$, then we have $1 - x_{vw} \geq 1 - x_{uv} - x_{uw} \geq 1 - 2\alpha$, so we can pay for each of these edges by charging a factor of $\frac{1}{1 - 2\alpha}$ times its LP-cost.

Thus, we consider edges joining $\{u\} \cup T$ with $S - (\{u\} \cup T)$. We call these edges *cross-edges* for their endpoints. A standard argument (see the extended version of this paper (Puleo & Milenkovic, 2016)) shows that for $z \in S - (\{u\} \cup T)$, the total cluster-cost of the cross-edges for z is at most $\max\{1/(1 - 2\alpha), 2/\alpha\}$ times the LP-cost of those edges, so the vertices outside $\{u\} \cup T$ can be dealt with easily.

However, we also must bound the cluster-cost at vertices inside $\{u\} \cup T$. This is where we use the maximality of $|T_u^*|$.

Let $w \in \{u\} \cup T$. First consider the positive cross-edges wz such that $x_{wz} \geq \gamma$. Any such edge has cluster-cost 1 and already has LP-cost at least γ, so charging $1/\gamma$ times the LP-cost to such an edge pays for its cluster cost. Now let $X = \{z \in S - (\{u\} \cup T)\colon\ x_{wz} < \gamma\}$; we still must pay for the edges wz with $z \in X$.

If $x_{uw} \leq k_1\alpha$, which includes the case $u = w$, then for all

$z \in X$, we have $x_{wz} \geq x_{uz} - x_{uw} \geq \alpha - k_1\alpha = (1-k_1)\alpha$. Hence, for any positive edge wz with $z \in X$, the LP-cost of wz is at least $(1-k_1)\alpha$, and so the cluster cost of the edge wz is at most $1/((1-k_1)\alpha)$ times the LP cost. Charging this factor to each cross-edge pays for the cluster-cost of each cross-edge.

Now suppose $x_{uw} > k_1\alpha$. Since $k_1\alpha > \gamma$, this implies $w \notin T^*_u$. In this case, it is possible that w may have many positive neighbors $z \in X$ for which x_{wz} is quite small, so we cannot necessarily pay for the cluster-cost of the edges joining w and X by using their LP-cost. Instead, we charge their cluster-cost to the LP-cost of edges within T.

Observe that $X \subseteq T^*_w$, and hence $|T^*_w| \geq |X|$. By the maximality of $|T^*_u|$, this implies that $|T^*_u| \geq |X|$. Now for any $v \in T^*_u$, we have the following bounds:

$$\begin{aligned} x_{wv} &\geq x_{uw} - x_{uv} \geq k_1\alpha - \gamma, \\ 1 - x_{wv} &\geq 1 - x_{uw} - x_{uv} \geq 1 - \alpha - \gamma. \end{aligned}$$

Since $\alpha < 1/2$ and $k_1 \leq 1$, we have $k_1\alpha \leq \alpha < 1-\alpha$, so these lower bounds imply that each edge wv with $v \in T^*_u$ has LP-cost at least $k_1\alpha - \gamma$, independent of whether wv is a positive or negative edge. Thus, the total LP cost of edges joining w to T^*_u is at least $(k_1\alpha - \gamma)\,|T^*_u|$.

Since the total cluster-cost of edges joining w and X is at most $|X|$ and since $|T^*_u| \geq |X|$, we can pay for these edges by charging each edge wv with $v \in T^*_u$ a factor of $\frac{1}{k_1\alpha-\gamma}$ times its LP-cost.

Having paid for all cluster-costs, we now look at the total charge accrued at each vertex. Fix any vertex v and an edge vw incident to v. We bound the total amount charged to vw by v in terms of the LP-cost of vw. There are three distinct possibilities for the edge vw: either vw ended inside a cluster, or v was clustered before w, or w was clustered before v.

Case 1: *vw ended within a cluster.* In this case, v may have made the following charges:

- A charge of $\frac{1}{(1-2k_3)(k_3-k_2)\alpha}$ times the LP-cost, to pay for a "bank account" for v,
- A charge of $\frac{1}{1-2\alpha}$ times the LP-cost, to pay for vw itself if vw is a negative edge,
- A charge of $\frac{1}{k_1\alpha-\gamma}$ times the LP-cost, to pay for positive edges leaving the v-cluster.

Thus, in this case the total cost charged to vw by v is at most c_1 times the LP-cost of vw, where

$$c_1 = \frac{1}{(1-2k_3)(k_3-k_2)\alpha} + \frac{1}{1-2\alpha} + \frac{1}{k_1\alpha - \gamma}.$$

Case 2: *v was clustered before w.* In this case, v may have made the following charges:

- A charge of $\frac{1}{(1-2k_3)(k_3-k_2)\alpha}$ times the LP-cost, to pay for a "bank account" for v,
- A charge of at most $\frac{2}{\alpha}$ times the LP-cost, to pay for all cross-edges if v was output as a Type 1 cluster,
- A charge of at most $\max\left\{\frac{1}{(1-k_1)\alpha}, \frac{1}{\gamma}\right\}$ times the LP-cost, to pay for vw if v was output in a Type 2 cluster.

Note that $k_1 > 1/2$ implies that $\frac{1}{(1-k_1)\alpha} \geq \frac{2}{\alpha}$, so we may disregard the case where v is output as a Type 1 cluster. Thus, in this case the total cost charged to vw by v is at most c_2 times the LP-cost of vw, where

$$c_2 = \frac{1}{(1-2k_3)(k_3-k_2)\alpha} + \max\left\{\frac{1}{(1-k_1)\alpha}, \frac{1}{\gamma}\right\}.$$

Case 3: *w was clustered before v.* In this case, v may have made the following charges:

- A charge of at most $\frac{1}{(1-2k_3)(k_3-k_2)\alpha}$ times the LP-cost, to pay for a "bank account" for v,
- A charge of at most $\frac{1}{k_2\alpha}$ times the LP-cost, to pay for the cluster-cost of vw if vw is a positive edge and w was output as a Type 1 cluster,
- A charge of at most

  $$\max\left\{\frac{1}{1-2\alpha}, \frac{2}{\alpha}\right\}$$

  times the LP-cost, to pay for vw if w was output in a Type 2 cluster.

Clearly vw cannot receive both the second and third types of charge. Furthermore, since $k_2 \leq 1/4$, we have $\frac{1}{k_2\alpha} \geq \frac{2}{\alpha}$. Since $k_2\alpha \leq 1-2\alpha$, we see that $\frac{1}{k_2\alpha}$ is the largest charge that vw could receive from either the second or third type of charge. Thus, in this case the total cost charged to vw by v is at most c_3 times the LP-cost, where

$$c_3 = \frac{1}{(1-2k_3)(k_3-k_2)\alpha} + \frac{1}{k_2\alpha.}$$

Thus, the approximation ratio of the algorithm is at most $\max\{c_1, c_2, c_3\}$. We wish to choose the various parameters to make this ratio as small as possible, subject to the various assumptions on the parameters required for the correctness of the proof. It seems difficult to obtain an exact solution to this optimization problem. Solving the problem numerically, we obtained the following values for the parameters:

$$\alpha = 0.465744 \qquad \gamma = 0.0887449$$
$$k_1 = 0.767566 \quad k_2 = 0.117219 \quad k_3 = 0.308433.$$

These parameters yield an approximation ratio of roughly 48. □

4. A Rounding Algorithm for One-Sided Biclustering

In this section, we consider a version of the f-Correlation Clustering problem on complete bipartite graphs. Let G be a complete bipartite graph with edges labeled $+$ and $-$, and let V_1 and V_2 be its partite sets. We will obtain a rounding algorithm that transforms any fractional clustering x into a discrete clustering $\mathcal{C}$ such that $\mathrm{err}(\mathcal{C})_v \le c\,\mathrm{err}(x)_v$ for all $v \in V_1$. Our algorithm is shown in Algorithm 2.

Our algorithm does not guarantee any upper bound on $\mathrm{err}(\mathcal{C})_v$ for $v \in V_2$: as the algorithm treats the sides V_1 and V_2 asymmetrically, it is difficult to control the per-vertex error at V_2. Nevertheless, an error guarantee for the vertices in V_1 suffices for some applications. Our approach is motivated by applications in recommender systems, where vertices in V_1 correspond to users, while vertices in V_2 correspond to objects to be ranked. In this context, quality of service conditions only need to be imposed for users, and not for objects.

Theorem 6. *Let G be a labeled complete bipartite graph with partite sets V_1 and V_2, let α, γ be parameters as described in Algorithm 2, and let x be any fractional clustering of G. If $\mathcal{C}$ is the clustering produced by Algorithm 2 with the given input, then for all $v \in V_1$ we have $\mathrm{err}(\mathcal{C})_v \le c\,\mathrm{err}(x)_v$, where c is a constant depending only on α and γ.*

We note that the proof of Theorem 6 is actually simpler than the proof of Theorem 5, because the focus on errors only at V_1 eliminates the need for the "bad pivots" argument used in Theorem 6. This also leads to a smaller value of c in Theorem 6 than we were able to obtain in Theorem 5.

Proof. As before, we make charges to pay for the new cluster costs at each vertex of V_1 as each cluster is output, splitting into cases according to the type of cluster. Let k_1 be a constant to be determined, with $k_1\alpha > \gamma$.

Case 1: *A Type 1 cluster $\{u\}$ is output.* In this case, the only cluster costs incurred are the positive edges incident to u, all of which have their other endpoint in V_2. The averaging argument used in Case 1 of Section 3 shows that charging every edge incident to u a factor of $2/\alpha$ times its LP cost pays for the cluster cost of all such edges.

Case 2: *A Type 2 cluster $\{u\} \cup T$ is output.* Negative edges within the cluster are easy to pay for: if w_1w_2 is a negative edge within the cluster, with $w_i \in V_i$, then we have

$$1 - x_{w_1w_2} \ge 1 - x_{uw_1} - x_{uw_2} \ge 1 - 2\alpha,$$

so we can pay for the cluster-cost of such an edge by charging it a factor of $1/(1-2\alpha)$ times its LP-cost.

We still must pay for positive edges joining the cluster with the rest of S; we call such edges *cross-edges*. Each such edge must be paid for at its endpoint in V_1.

If $z \in V_1$ is a vertex outside the cluster, then a standard argument (see the extended version of this paper (Puleo & Milenkovic, 2016)) shows that the cross-edges for z can be paid for by charging each such edge a factor of $\max\{1/(1-2\alpha), 2/\alpha)\}$ times its LP cost.

Now let $w \in V_1$ be a vertex inside the cluster. We must pay for the cross-edges incident to w using the LP-cost of the edges incident to w. First consider the positive edges from w to vertices z outside the cluster such that $x_{wz} \ge \gamma$. Any such edge has cluster-cost 1 and LP-cost at least γ, so charging each such edge a factor of $1/\gamma$ times its LP-cost pays for its cluster cost. Let $X = \{z \in (S \cap V_2) - T \colon\ x_{wz} < \gamma\}$; we must pay for the edges wz with $z \in X$. Note that $x_{uz} > \alpha$ for all $z \in X$, since $z \in X$ implies $z \notin T$.

If $x_{uw} \le k_1\alpha$, then for all $z \in X$, we have

$$x_{wz} \ge x_{uz} - x_{uw} \ge (1 - k_1)\alpha.$$

Hence, for any positive cross-edge wz with $z \in X$, the LP-cost of wz is at least $(1-k_1)\alpha$, and so we can pay for the cluster-cost of wz by charging wz a factor of $\frac{1}{(1-k_1)\alpha}$ times its LP-cost.

Now suppose $x_{uw} > k_1\alpha$. As before, we pay for the cross-edges by charging the edges inside the cluster. Observe that $|T^*_w| \ge |X|$. Since u was chosen to maximize $|T^*_u|$, this implies that $|T^*_u| \ge |X|$. For any $v \in T^*_u$, we have

$$x_{wv} \ge x_{uw} - x_{uv} \ge k_1\alpha - \gamma.$$

On the other hand, for any $v \in T^*_u$ we also have

$$1 - x_{wv} \ge 1 - x_{uw} - x_{uv} \ge 1 - \alpha - \gamma \ge \alpha - \gamma.$$

Since $k_1 \le 1$, it follows that the edge wv has LP-cost at least $k_1\alpha - \gamma$ independent of whether wv is positive or negative. Thus, the total LP cost of edges joining w to T^*_u is at least $(k_1\alpha - \gamma)\,|T^*_u|$.

Since the total cluster-cost of the cross- edges joining w and X is at most $|X|$ and since $|T^*_u| \ge |X|$, we can pay for

Algorithm 2 Round fractional clustering to obtain a discrete clustering, using threshold parameters α, γ with $\alpha < 1/2$ and $\gamma < \alpha$.

```
Let $S = V(G)$.
while $V_1 \cap S \neq \emptyset$ do
  For each $u \in V_1 \cap S$, let $T_u = \{w \in S - \{u\}\colon\ x_{uw} \leq \alpha\}$ and let $T_u^* = \{w \in V_2 \cap S\colon\ x_{uw} \leq \gamma\}$.
  Choose a pivot vertex $u \in V_1 \cap S$ that maximizes $|T_u^*|$.
  Let $T = T_u$.
  if $\sum_{w \in V_2 \cap T} x_{uw} \geq \alpha \, |V_2 \cap T| \, /2$ then
    Output the singleton cluster $\{u\}$. {Type 1 cluster}
    Let $S = S - \{u\}$.
  else
    Output the cluster $\{u\} \cup T$. {Type 2 cluster}
    Let $S = S - (\{u\} \cup T)$.
  end if
end while
Output each remaining vertex of $V_2 \cap S$ as a singleton cluster.
```

the cross-edges by charging each edge wv with $v \in T_u^*$ a factor of $\frac{1}{k_1\alpha - \gamma}$ times its LP-cost.

Having paid for all cluster-costs, we now look at the total charge accrued at each vertex. Fix a vertex $v \in V_1$ and an edge vw incident to v. We bound the total amount charged to vw by v in terms of the LP-cost of vw. There are three distinct possibilities for the edge vw: either vw ended inside a cluster, or v was clustered before w, or w was clustered before v.

Case 1: *vw ended within a cluster.* In this case, v may have made the following charges:

- A charge of at most $\frac{1}{1-2\alpha}$ times the LP cost, to pay for vw itself if vw is a negative edge,
- A charge of $\frac{1}{k_1\alpha - \gamma}$ times the LP-cost, to pay for positive edges leaving the v-cluster.

Thus, in this case the total cost charged to vw by v is at most c_1 times the LP-cost of vw, where

$$c_1 = \frac{1}{1-2\alpha} + \frac{1}{k_1\alpha - \gamma}.$$

Case 2: *v was clustered before w.* In this case, v may have made the following charges:

- A charge of $2/\alpha$ times the LP cost, to pay for vw if v was output as a singleton,
- A charge of $\max\{\frac{1}{(1-k_1)\alpha}, \frac{1}{\gamma}\}$ times the LP cost, to pay for vw if v was output in a nonsingleton cluster,

Since v makes at most one of the charges above, the total cost charged to vw by v is at most c_2 times the LP-cost of vw, where

$$c_2 = \max\left\{\frac{1}{(1-k_1)\alpha},\ \frac{1}{\gamma},\ \frac{2}{\alpha}\right\}.$$

Case 3: *w was clustered before v.* In this case, v may have made the following charges:

- A charge of at most $\max\{\frac{1}{1-2\alpha}, \frac{2}{\alpha}\}$ times the LP cost, to pay for cross-edges at v if w is output in a nonsingleton cluster.

Thus, in this case the total cost charged to vw by v is at most c_3 times the LP-cost of vw, where

$$c_3 = \max\left\{\frac{1}{1-2\alpha},\ \frac{2}{\alpha}\right\}.$$

The approximation ratio is $\max\{c_1, c_2, c_3\}$. Numerically, we obtain an approximation ratio of at most 10 by taking the following parameter values:

$$\alpha = 0.377 \qquad \gamma = 0.102 \qquad k_1 = 0.730 \qquad \square$$

Acknowledgments

The authors thank Dimitris Papailiopoulos for helpful discussions. The authors also acknowledge funding from the NSF grants IOS 1339388 and CCF 1527636, 1526875, 1117980. Research of the first author was supported by the IC Postdoctoral Program.

References

Ailon, Nir, Charikar, Moses, and Newman, Alantha. Aggregating inconsistent information: ranking and clustering. In *Proceedings of the thirty-seventh annual ACM* *symposium on Theory of computing*, pp. 684–693. ACM, 2005.

Ailon, Nir, Charikar, Moses, and Newman, Alantha. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):23, 2008.

Ailon, Nir, Avigdor-Elgrabli, Noa, Liberty, Edo, and Van Zuylen, Anke. Improved approximation algorithms for bipartite correlation clustering. *SIAM Journal on Computing*, 41(5):1110–1121, 2012.

Amit, Noga. The bicluster graph editing problem. Master's thesis, Tel Aviv University, 2004.

Bansal, Nikhil, Blum, Avrim, and Chawla, Shuchi. Correlation clustering. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, FOCS '02, pp. 238–, Washington, DC, USA, 2002. IEEE Computer Society. ISBN 0-7695-1822-2.

Bansal, Nikhil, Blum, Avrim, and Chawla, Shuchi. Correlation clustering. *Mach. Learn.*, 56(1-3): 89–113, June 2004. ISSN 0885-6125. doi: 10.1023/B:MACH.0000033116.57574.95.

Bansal, Nikhil, Feige, Uriel, Krauthgamer, Robert, Makarychev, Konstantin, Nagarajan, Viswanath, Naor, Joseph, and Schwartz, Roy. Min-max graph partitioning and small set expansion. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science—FOCS 2011*, pp. 17–26. IEEE Computer Soc., Los Alamitos, CA, 2011. doi: 10.1109/FOCS.2011.79. URL `http://dx.doi.org/10.1109/FOCS.2011.79`.

Boyd, Stephen and Vandenberghe, Lieven. *Convex optimization*. Cambridge University Press, 2004.

Charikar, Moses, Guruswami, Venkatesan, and Wirth, Anthony. Clustering with qualitative information. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '03, pp. 524–, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-2040-5.

Charikar, Moses, Guruswami, Venkatesan, and Wirth, Anthony. Clustering with qualitative information. *J. Comput. Syst. Sci.*, 71(3):360–383, October 2005. ISSN 0022-0000. doi: 10.1016/j.jcss.2004.10.012.

Chawla, Shuchi, Krauthgamer, Robert, Kumar, Ravi, Rabani, Yuval, and Sivakumar, D. On the hardness of approximating multicut and sparsest-cut. *Computational Complexity*, 15(2):94–114, 2006.

Chawla, Shuchi, Makarychev, Konstantin, Schramm, Tselil, and Yaroslavtsev, Grigory. Near optimal lp rounding algorithm for correlation clustering on complete and complete k-partite graphs, 2014.

Cheng, Yizong and Church, George M. Biclustering of expression data. In *ISMB*, volume 8, pp. 93–103, 2000.

Demaine, Erik D, Emanuel, Dotan, Fiat, Amos, and Immorlica, Nicole. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2): 172–187, 2006.

Khot, Subhash. On the power of unique 2-prover 1-round games. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pp. 767–775. ACM, 2002.

Komusiewicz, Christian and Uhlmann, Johannes. Cluster editing with locally bounded modifications. *Discrete Appl. Math.*, 160(15):2259–2270, 2012. ISSN 0166-218X. doi: 10.1016/j.dam.2012.05.019.

Kriegel, Hans-Peter, Kröger, Peer, and Zimek, Arthur. 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, 2009.

Lim, Shiau Hong, Chen, Yudong, and Xu, Huan. A convex optimization framework for bi-clustering. In *Proceedings of The 32nd International Conference on Machine Learning*, pp. 1679–1688, 2015.

Pan, Xinghao, Papailiopoulos, Dimitris, Oymak, Samet, Recht, Benjamin, Ramchandran, Kannan, and Jordan, Michael I. Parallel correlation clustering on big graphs. *arXiv preprint arXiv:1507.05086*, 2015.

Puleo, Gregory J. and Milenkovic, Olgica. Correlation clustering and biclustering with locally bounded errors. *arXiv preprint arXiv:1506.08189*, 2016.

Steurer, D. and Vishnoi, N. Connections between unique games and multcut. Technical Report TR09-125, Electronic Colloquium on Computational Complexity, 2009. Available at `http://eccc.hpi-web.de/report/2009/125`.

Symeonidis, Panagiotis, Nanopoulos, Alexandros, Papadopoulos, Apostolos, and Manolopoulos, Yannis. Nearest-biclusters collaborative filtering with constant values. In *Advances in web mining and web usage analysis*, pp. 36–55. Springer, 2007.

van Zuylen, Anke and Williamson, David P. Deterministic pivoting algorithms for constrained ranking and clustering problems. *Math. Oper. Res.*, 34(3):594–620, 2009. ISSN 0364-765X. doi: 10.1287/moor.1090.0385.