Abstract

In this paper, we propose and study a semi-random model for the Correlation Clustering problem. We give an approximation algorithm that given an instance from this model finds a solution of value $(1 + \delta) \text{opt-cost} + O(\delta n \log^3 n)$ with high probability, where opt-cost is the value of the optimal solution (for every $\delta > 0$).

1 Introduction

One of the most commonly used algorithmic tools in data analysis and machine learning is clustering — partitioning a corpus of data into groups based on similarity. Correlation clustering tackles the problem of clustering objects when we are given qualitative information about similarity between some pairs of these objects. This captures many scenarios in machine learning where we do not have the actual representations of the objects, but we instead know the relationships between them.

This qualitative information is represented in the form of a graph $G(V, E, c)$ in which edges $E$ have non-negative costs given by $c : E \to [0, 1]$ and are labeled with signs $\{+, -\}$ (we represent the ‘+’ edges by $E_+$ and the ‘−’ edges by $E_-$ respectively). An edge $(u, v) \in E_+$ denotes that vertices $u$ and $v$ are similar, while an edge $(u, v) \in E_-$ denotes that $u$ and $v$ are dissimilar. This qualitative information, however, may not be consistent; hence, we want to find a clustering in which most of the edges in $E_+$ lie within clusters and most of the edges in $E_-$ go between clusters. Specifically, our goal is to find a clustering that minimizes the cost of disagreements: the cost of edges in $E_-$ between different clusters, and the cost of edges $E_+$ within the clusters. See Section 2 for a formal definition.

One interesting feature about this clustering problem is that we do not specify the number of clusters, unlike in most other clustering problems such as the $k$-means or $k$-median [WS11]. Hence, correlation clustering is particularly useful when we do not have prior knowledge of the number of clusters that we wish to divide the data into. Correlation clustering can be also viewed as an agnostic learning problem where each edge is an example, and the target function is represented by groups of vertices given by the clustering.

Correlation clustering was introduced in [BBC04], and implicitly in [BDSY99] as Cluster editing. Bansal, Blum and Chawla [BBC04] showed that the problem is NP-hard even on complete graphs, i.e., when we are given similarity information between every pair of objects. The state-of-the-art approximation algorithm [CGW05] [DEFI06] achieves an $O(\log n)$ approximation for minimizing disagreements in the worst-case. The complementary objective of maximizing agreements is easier from the approximability standpoint and a 0.766 factor approximation is known [CGW05] [Swa04]. For
the special case of complete graphs (with unit costs on edges), small constant factor approximations are known due to a series of works \[BBC04\, ACN08\].

On the inapproximability side, the problem is known to be APX-hard even on the complete graph with unit cost edges \[CGW05\]. Furthermore, there is a gap-preserving reduction \[CGW05\, DEFI06\] from the classic minimum Multicut problem, for which the current state-of-the-art algorithm gives a \(\Theta(\log n)\) factor approximation \[GVY93\]. This also implies that assuming the Unique Games Conjecture, we cannot obtain a constant factor approximation in the worst-case \[KV05\]. To summarize, despite our best efforts we only know logarithmic factor approximation algorithms for the problem of minimizing disagreements, and assuming the Unique Games Conjecture, we cannot get a constant factor approximation for worst-case instances.

However, our primary interest in solving Correlation Clustering comes from its numerous applications, and the instances that we encounter in these applications are not worst-case instances. This motivates the study of the average-case complexity of the problem and raises the question:

Can we design algorithms with better provable approximation guarantees for realistic average-case models for Correlation Clustering?

Natural average-case models have been studied for Correlation Clustering on complete graphs (with unit costs) in the presence of random noise \[BDSY99\, MS10\, ES09\]. Ben-Dor, Shamir and Yakhini \[BDSY99\] consider a model in which there is an arbitrary partitioning (clustering) of vertices of a graph. There is an edge of unit cost between every pair of vertices. Initially, edges inside a cluster belong to \(E^+_\ast\) (labeled +) and edges between clusters belong to \(E^\ast\) (labeled −). This constitutes a correlation clustering instance with a perfect solution. Then, each of the edge labels is flipped (+ is flipped to −, and − to +) with probability \(\frac{1}{2} - \varepsilon\) independently at random.

Mathieu and Schudy \[MS10\] consider a generalization of this model where there is an adversary. In their model, for each edge, with probability \(\frac{1}{2} + \varepsilon\) we do not flip the initial edge label, and with probability \(\frac{1}{2} - \varepsilon\) the adversary decides whether to flip the edge label or not. Mathieu and Schudy give an algorithm that finds a clustering of cost at most \(1 + O(n^{-1/6})\) times the cost of the optimal clustering, as long as \(\varepsilon \geq n^{-1/3}\).

However, these average-case models and algorithms deal only with the special case of \emph{complete graphs}, with all edges having unit cost. It may be unrealistic to assume that every pair of vertices have the same amount of similarity or dissimilarity (all unit costs) or that we have similarity information between \emph{every} pair of vertices.

1.1 Our Semi-random model

In this paper, we propose and study new semi-random models for generating general instances of Correlation Clustering, which we believe capture many properties of real world instances. Our basic semi-random model generalizes the model of Mathieu and Schudy \[MS10\] to arbitrary graphs \(G(V, E, c)\) with costs. A semi-random instance \(\{G(V, E, c), (E_\ast, E^\ast)\}\) is generated as follows:

1. The adversary chooses an undirected graph \(G(V, E, c)\) and a partition \(P^\ast\) of the vertex set \(V\).

2. Every edge is \(E\) is included in set \(E_R\) independently with probability \(\varepsilon\).

3. Every edge \((u, v) \in E \setminus E_R\) with \(u\) and \(v\) in the same cluster of \(P^\ast\) is included in \(E_\ast\). Similarly, every edge \((u, v) \in E \setminus E_R\), with \(u\) and \(v\) in different clusters of \(P^\ast\) is included in \(E^\ast\).

4. The adversary adds every edge from \(E_R\) either to \(E_\ast\) or to \(E^\ast\) (but not to both sets).
This model can be further generalized to an adaptive semi-random model as we describe in Section 2.1.

As an illustration, consider a document classification system where vertices represent different documents. We have qualitative similarity information from a classifier about some pairs of documents. Documents in the same cluster are supposed to be related, and documents in different clusters are supposed to be unrelated. Hence, in the presence of a perfect classifier, we would have a Correlation Clustering instance with zero cost. However, the classifier may be noisy, and it may flip the labels of the edges with some probability \( \left( \frac{1}{2} - \varepsilon \right) \). This results in an instance from our semi-random model.

1.2 Our Results

We develop an algorithm for semi-random instances of Correlation Clustering which gives a PTAS for instances from our semi-random model.

**Theorem 1.1.** For every \( \delta > 0 \), there is a polynomial-time algorithm that given a semi-random instance \( \{G(V, E, c), (E_\pm, E_\mp)\} \) of Correlation Clustering (with noise probability \( \varepsilon < 1/4 \)), finds a clustering that has disagreement cost \( (1 + \delta) \text{opt-cost} + O((1 - 2\varepsilon)^{-1} \delta^{-3} n \log^3 n) \) w.h.p., where opt-cost is the cost of disagreements of the optimal solution for the instance.

The approximation additive term is much smaller than the cost of the planted solution if the average degree \( \Delta \gg \varepsilon^{-1} \text{polylog } n \). Note that we compare the performance of our algorithm with the cost of the optimal solution. Thus we get a true \( 1 + \delta \) approximation scheme (with an extra additive approximation term). This is in contrast to previous semi-random model results [MMV12, MMV13], which compare to the planted solution.

1.3 Related Work on Semi-random Models

Over the last two decades, there has been extensive research on average-case complexity of many important combinatorial optimization problems. In semi-random models, an instance is generated in a series of steps: some of them are random and some are adversarial. Because of that, semi-random instances may have much more structure than completely random instances.

Research on semi-random models was initiated by Blum and Spencer [BS95], who introduced and investigated semi-random models for \( k \)-coloring. Semi-random models have also been studied for graph partitioning problems [FK98, CSX12, MMV12, MMV14], Independent Set [FK98], Maximum Clique [FK00], Unique Games [KMM11], and other problems. Most related to our current work, both in the nature of the model and in the techniques used, is our recent result [MMV13] on semi-random instances of Minimum Feedback Arc Set. While the techniques used in both papers are conceptually similar, the semidefinite (SDP) relaxation for Correlation Clustering that we use in this paper is very different from the SDP relaxation for Minimum Feedback Arc Set used in [MMV13]. We use a standard SDP relaxation for Correlation Clustering that was previously used in [Swa04, MS10, ES09].

As mentioned earlier, Mathieu and Schudy [MS10] recently considered a semi-random model for Correlation Clustering on complete graphs with unit edge costs. They give a PTAS for the problem using the SDP relaxation that was introduced by Swamy [Swa04]. Later, Elsner and Schudy [ES09] conducted an empirical evaluation of this and other algorithms on instances with qualitative information for all pairs of vertices (i.e. in the complete graph setting).
1.4 Techniques

The general approach of our algorithm for semi-random instances of Correlation Clustering is influenced by [KMM11], [MMV12] and [MMV13]. At a very high level, this approach can be outlined as follows. We write an SDP relaxation for the problem and find a small family of representative SDP solutions. Now for a fixed planted solution, we show that every fixed feasible SDP solution is far from the optimal SDP solution unless it has certain structural properties. Finally, using the union bound and the fact that there are much fewer representative SDP solutions than semi-random instances, we prove that every feasible SDP solution is far from the optimal SDP solution unless it has the required structural properties. Thus the optimal solution has these structural properties with high probability.

In the SDP relaxation, there is a unit vector \( \bar{u} \) associated with each vertex, and the inner product \( \langle \bar{u}, \bar{v} \rangle \in [0,1] \) tries to capture whether \( u \) and \( v \) are in the same cluster (a large inner product implies they are likely to be in the same cluster). Our structural theorem shows that with high probability in the optimal SDP solution, \( \langle \bar{u}, \bar{v} \rangle \geq 1 - \delta \) for almost all random edges whose label was flipped from + to −, and \( \langle \bar{u}, \bar{v} \rangle \leq \delta \) for almost all the random edges whose label was flipped from − to +. So first we remove all − labeled edges \((u,v)\) with \( \langle \bar{u}, \bar{v} \rangle \geq 1 - \delta \) and all + labeled edges with \( \langle \bar{u}, \bar{v} \rangle \leq \delta \). We obtain an instance with a solution of cost \( \text{opt-cost} / (\log n \log \log n) \). Then we just run the “off-the-shelf” algorithm for minimizing disagreements [CGW05, DEFI06] and get a clustering.

2 Preliminaries

In the Correlation Clustering Problem, we are given a graph \( G(V,E,c) \) with the edge cost function \( c : E \rightarrow [0,1] \), and the edges are divided into two disjoint groups \( E = E_\approx \cup E_\neq \). Each edge \( (u,v) \) in \( E_\approx \) indicates that \( u \) and \( v \) are similar (the amount of similarity being \( c(u,v) \)); and each edge \( (u,v) \in E_\neq \) indicates that \( u \) and \( v \) are dissimilar. We want to find a partition \( \mathcal{P} \) of \( G \) that minimizes the cost of violated constraints imposed by the edges:

\[
\min_{\mathcal{P}} \sum_{(u,v) \in E_\approx : \mathcal{P}(u) \neq \mathcal{P}(v)} c(u,v) + \sum_{(u,v) \in E_\neq : \mathcal{P}(u) = \mathcal{P}(v)} c(u,v).
\]

This is the cost of cut edges in \( E_\approx \) and uncut edges in \( E_\neq \). Above, \( \mathcal{P}(u) \) denotes the cluster that contains the vertex \( u \).

Given an instance of the Correlation Clustering Problem and a partition \( \mathcal{P} \), we say that an edge \((u,v) \in E\) is consistent with \( \mathcal{P} \), if \((u,v) \in E_\approx \) and \( \mathcal{P}(u) = \mathcal{P}(v) \), or \((u,v) \in E_\neq \) and \( \mathcal{P}(u) \neq \mathcal{P}(v) \).

For any subset of edges \( F \subseteq E \), let \( c(F) \) represent the cost of the edges in \( F \) i.e. \( c(F) = \sum_{e \in F} c(e) \).

2.1 Adaptive Model

We study a more general “adaptive” semirandom model. A semirandom instance is generated as follows. We start with a graph \( G_0(V,\emptyset) \) on \( n \) vertices with no edges and a partition \( \mathcal{P}^* \) of \( V \) into disjoint sets, which we call the planted partition. The adversary adds edges one by one. We denote the edge chosen at step \( t \) by \( e_t \) and its cost \( c(e_t) \in [0,1] \). After the adversary adds an edge \( e_t \) to the set of edges, the nature flips a coin and with probability \( \varepsilon \) adds \( e \) to the set of random edges \( E_R \). The next edge \( e_{t+1} \) chosen by the adversary may depend on whether \( e_t \) belongs to

---

\(^2\)To obtain some of their results, [KMM11] and [MMV12] use C-SDP or local SDP relaxations.
The adversary stops the semirandom process at a stopping time $T$. Thus, we obtain a graph $G^*(V, \{e_1, \ldots, e_T\}, c)$ and a set of random edges $E_R$. We denote the set of all edges by $E^* = \{e_1, \ldots, e_T\}$.

Once the graph $G^*(V, E^*)$ and the set $E_R$ is generated, we define sets $E_\text{=}^*$ and $E_\text{\neq}^*$: $E_\text{=}^* = \{(u, v) : \mathcal{P}^*(u) = \mathcal{P}^*(v)\}$, and $E_\text{\neq}^* = \{(u, v) : \mathcal{P}^*(u) \neq \mathcal{P}^*(v)\}$. Note that $\mathcal{P}^*$ is the solution of cost 0 for the instance $\{G^*, (E_\text{=}^*, E_\text{\neq}^*)\}$. The adversary picks a set of edges $E \subseteq E^*$ that contains all edges in $E^* \setminus E_R$ (i.e., $E^* \setminus E_R \subseteq E \subseteq E^*$), and splits them into sets $E_\text{=}^*$, and $E_\text{\neq}^*$ such that $E_\text{=}^* \Delta E_\text{\neq}^* \subseteq E_R$. The semirandom instance of the problem is $\{G(V, E, c), (E_\text{=}^*, E_\text{\neq}^*)\}$. This is the instance the algorithm gets. Of course, the algorithm does not get the set of random edges $E_R$. Note, that the cost of the planted solution $\mathcal{P}^*$ is at most the cost of the edges $E_R \cap E$ i.e. $c(E_R \cap E)$, since all edges in $E \setminus E_R$ are consistent with $\mathcal{P}^*$.

Note that this Adaptive Model is more general than the Basic Semi-random model we introduced earlier. The basic semi-random model corresponds to the case when the whole set of edges $E^*$ is fixed in advance independent of the random choices made in $E_R$. However, in the adaptive model the edge $e_t$ can be chosen based on which of the edges $e_1, \ldots, e_{t-1}$ belong to $E_R$. For instance, the adversary can choose edge $e_t$ from the portion of the graph where many of the previously chosen edges belong to $E_R$.

\section{Algorithm}

**SDP relaxation.** We use the standard SDP relaxation for the problem\cite{Swa04, MS10, ES09}. For every vertex $u$, we have a unit vector $\bar{u}$. For two vertices $u$ and $v$, we interpret the inner product $\langle \bar{u}, \bar{v} \rangle \in [0, 1]$ as the indicator of the event: $u$ and $v$ lie in the same partition. The objective function is to minimize

$$\min_{\mathcal{P}} \sum_{(u,v) \in E_\text{=}^*} c(u,v)(1 - \langle \bar{u}, \bar{v} \rangle) + \sum_{(u,v) \in E_\text{\neq}^*} c(u,v)\langle \bar{u}, \bar{v} \rangle.$$  

The first term is the fractional number of cut edges in $E_\text{=}^*$; the second term is the fractional number of uncut edges in $E_\text{\neq}^*$. The obtained semidefinite program is as follows.

$$\min_{\mathcal{P}} \sum_{(u,v) \in E_\text{=}^*} c(u,v)(1 - \langle \bar{u}, \bar{v} \rangle) + \sum_{(u,v) \in E_\text{\neq}^*} c(u,v)\langle \bar{u}, \bar{v} \rangle.$$  

subject to: for all $u, v \in V$,

$$\langle \bar{u}, \bar{v} \rangle \in [0, 1];$$
$$\|u\|^2 = 1.$$  

Let us explain why this SDP is a relaxation. If $\mathcal{P}$ is a clustering, then the intended solution is as follows: $\bar{u}_P$ is a mapping from $\mathcal{P}$ to $\{0, 1\}$:

$$\bar{u}_P = \begin{cases} 1, & \text{if } u \in P; \\ 0, & \text{otherwise}. \end{cases}$$

Since $\bar{u}_P$ equals 1 only for one $P \in \mathcal{P}$, we have $\|\bar{u}\|^2 = 1$, and

$$\langle \bar{u}, \bar{v} \rangle = \sum_{P \in \mathcal{P}} \bar{u}_P \bar{v}_P = \begin{cases} 1, & \text{if } \mathcal{P}(u) = \mathcal{P}(v); \\ 0, & \text{otherwise.} \end{cases}$$
Hence, the SDP constraints are satisfied, and the SDP objective function exactly equals the objective function of the Correlation Clustering problem.

**Algorithm.** We now describe the algorithm. Fix a parameter $\delta = o(1) \in (0, 1/2)$. To simplify the notation, denote by $f(u, v)$ (for $(u, v) \in E$),

$$f(u, v) = \begin{cases} 1 - \langle \bar{u}, \bar{v} \rangle, & \text{if } (u, v) \in E_\varepsilon; \\ \langle \bar{u}, \bar{v} \rangle, & \text{if } (u, v) \in E_\bar{\varepsilon}. \end{cases} \tag{1}$$

Note, that the SDP objective function can be rewritten as

$$SDP = \sum_{(u, v) \in E} c(u, v)f(u, v).$$

The algorithm solves the SDP and removes all edges $(u, v)$ with $f(u, v) \geq (1-\delta)$. Then the algorithm runs the $O(\log n)$ approximation algorithm of Charikar et al. [CGW05] or Demaine et al. [DEFI06] on the remaining graph.

### 3.1 Analysis

We need to bound the number of edges removed at the first step and the number of edges cut by the $O(\log n)$ approximation algorithm at the second step. The SDP contribution of every edge $(u, v)$ removed is at least $c(u, v)(1-\delta)$. Thus the cost of edges removed is bounded by $SDP/(1-\delta) \leq (1 + 2\delta)OPT$. To bound the cost of the solution produced by the approximation algorithm at the second step, we need to bound the cost of the optimal solution for the remaining instance i.e., the instance with the set of edges $\{(u, v) \in E : f(u, v) \leq 1 - \delta\}$.

We define a function $f^*(u, v)$, which slightly differs from $f(u, v)$. For all $(u, v) \in E$,

$$f^*(u, v) = \begin{cases} 1 - \langle \bar{u}, \bar{v} \rangle, & \text{if } P^*(u) = P^*(v); \\ \langle \bar{u}, \bar{v} \rangle, & \text{if } P^*(u) \neq P^*(v). \end{cases} \tag{2}$$

Here, $P^*$ is the planted partition. Note that $P^*$ and $f^*(u, v)$ are not known to the algorithm. Observe that $f(u, v) = f^*(u, v)$ if the edge $(u, v)$ is consistent with the planted partition $P^*$, and $f(u, v) = 1 - f^*(u, v)$ otherwise. Our goal is to show that the algorithm removes all but very few edges inconsistent with $P^*$, i.e., edges $(u, v)$ with $f(u, v) = 1 - f^*(u, v)$. We prove the following theorem in Section 6. The proof relies on Theorem 4.4 presented in Section 4.

**Theorem 3.1.** Let $\{G = (V, E, c), (E_\varepsilon, E_\bar{\varepsilon})\}$ be a semirandom instance of the correlation clustering problem. Let $E_R$ be the set of random edges, and $P^*$ be the planted partition. Denote by $Q \subset E_R$ the set of random edges not consistent with $P^*$. Then, for $\lambda = C(1 - 2\varepsilon)^{-2}\gamma^{-2}\delta^{-3}n\log n$, for some universal constant $C$,

$$\Pr \left[ \sum_{(u, v) \in Q : f(u, v) \leq 1 - \delta} c(u, v) \geq \lambda + \frac{6\gamma}{1 - 2\varepsilon}c(Q) \right] = o(1).$$

where $f$ corresponds to any feasible SDP solution of cost at most $OPT$.

**Remark 3.2.** In the statement of Theorem 3.1, $c(Q)$ is the value of the solution given by the planted solution $P^*$. If $OPT = c(Q)$, then the planted solution $P^*$ is indeed an optimal clustering. The function $f(u, v)$ in the theorem that corresponds to the SDP contribution of edge $(u, v)$ could come from any (not necessarily optimal) SDP solution of cost at most $OPT$. This will be useful in Lemma 3.3.
Let $D = O(\log n)$ be the approximation algorithms of Charikar et al. [CGW05] or Demaine et al. [DEP06]. We apply Theorem 3.1 with

$$\gamma = \frac{\delta(1 - 2\varepsilon)}{6D}.$$  

The cost of edges in $\{ (u, v) \in Q : f(u, v) \leq 1 - \delta \}$ is bounded by

$$\lambda + \frac{6\gamma}{1 - 2\varepsilon} c(Q) \leq \lambda + D^{-1}\delta c(Q),$$

w.h.p., where $\lambda = O((1 - 2\varepsilon)^{-4}\delta^{-3}n \log^3 n)$. Thus, after removing edges with $f(u, v) \geq (1 - \delta)$, the cost of the optimal solution is at most $\lambda$ w.h.p. The approximation algorithm finds a solution of cost at most $D$ times of (3). Thus, the total cost of the solution returned by the algorithm is bounded by

$$(1 + 2\delta)OPT + D \times (\lambda + D^{-1}\delta \cdot c(Q)) = (1 + 3\delta)c(Q) + D\lambda = (1 + 3\delta)c(Q) + O((1 - 2\varepsilon)^{-4}\delta^{-3}n \log^3 n).$$

The above argument shows that the solution has small cost compared to the cost of the planted solution $P^*$. We can in fact use Theorem 3.1 to give a true approximation i.e., compared to the cost of the optimal solution $OPT$. This follows from the following lower bound on $OPT$ in terms of $c(Q)$ for semi-random instances.

**Lemma 3.3.** In the notation of Theorem 3.1, with probability $1 - o(1)$, $c(Q) \leq (1 + 2\delta)OPT + O((1 - 2\varepsilon)^{-4}\delta^{-3}n \log^3 n)$.

**Proof.** Let $f_{OPT}$ correspond to the “integral” SDP solution corresponding to the optimal solution $OPT$. In this solution, $f_{OPT}(u, v) = 1$ for positive edges $(u, v)$ which are across different clusters and negative edges $(u, v)$ which are in the same cluster. This SDP solution has cost $OPT$ and satisfies the conditions of Theorem 3.1. Hence, w.h.p., $c(Q \setminus (Q \cap OPT)) \leq \frac{\delta}{D} \cdot c(Q) + \lambda$. Hence,

$$c(Q) - OPT \leq \frac{\delta}{D} c(Q) + \lambda$$

and

$$OPT \geq (1 - \frac{\delta}{D}) \cdot c(Q) - \lambda.$$  

**Proof of Theorem 1.1** From Theorem 3.1, we get the total cost of the solution is bounded by

$$(1 + 2\delta)OPT + D \times (\lambda + D^{-1}\delta \cdot c(Q)) = (1 + 2\delta)OPT + D \times \lambda + \frac{\delta}{1 - \delta/D} \cdot (OPT + \lambda) \leq (1 + 4\delta)OPT + 2D\lambda = (1 + 4\delta)OPT + O((1 - 2\varepsilon)^{-4}\delta^{-3}n \log^3 n).$$

This finishes the analysis of the algorithm.  

7
4 Betting with Stakes Depending on the Outcome

We first informally describe the theorem we prove in this section. Consider the following game. Assume that we are given a set of vectors \( W \subset [0, 1]^m \). At every step \( t \), the player (adversary) picks an arbitrary not yet chosen coordinate \( e_t \in \{1, \ldots, m\} \), and the casino (nature) flips a coin such that with probability \( \varepsilon < 1/2 \), the player wins, and with probability \( 1 - \varepsilon > 1/2 \), the player looses. In the former case, we set \( X_t = 1 \); and in the latter case we set \( X_t = -1 \). At some point \( T \leq m \) the player stops the game. At that point, he picks a vector \( w \in W \) and declares that at time \( t \) his stake was \( w(e_t) \) dollars. We stress that the vector \( w \) may depend on the outcomes \( X_t \). Then, the player’s payoff equals

\[
\sum_{t=1}^{T} X_t w(e_t).
\]

If the player could pick an arbitrary \( w \) after the outcomes \( X_t \) are revealed, then clearly he could get a significant payoff by letting \( w(e_t) = 1 \), for \( X_t = 1 \), and \( w(e_t) = 0 \), otherwise. However, we assume that the set \( W \) of possible bets is relatively small. Then, we show that with high probability the payoff is negative unless the total amount of bets \( \sum_t w(e_t) \) is very small. The precise statement of the theorem (see below) is slightly more technical.

The main idea of the proof is that for any \( w \in W \) fixed in advance, the player is expected to lose with high probability, since the coin is not fair \( (\varepsilon < 1/2) \), and thus the casino has an advantage. In fact, the probability that the player wins exponentially small if the coordinates of \( w \) are sufficiently large. Now we union bound over all \( w \)’s in \( W \) and conclude that with high probability for every \( w \in W \), the player’s payoff is negative.

When we apply this theorem to a semirandom instance of Correlation Clustering (with unit costs i.e. \( c(e_t) = 1 \)), the stakes are defined by the solution of the SDP: for an edge \( e_t = (u, v) \), \( w(e_t) = f^*(u, v) \). Loosely speaking, we show that since the SDP value is at most \( \text{OPT} \), the game is profitable for the adversary. This implies that most stakes \( f^*(u, v) \) are close to 0. Now, if an edge \( (u, v) \) is consistent with the planted partition \( P^* \), then \( f(u, v) = f^*(u, v) \approx 0 \), and hence we do not remove this edge. On the other hand, if the edge is not consistent with the planted partition, then \( f(u, v) = 1 - f^*(u, v) \approx 1 \), hence we remove the edge.

**Lemma 4.1.** Let \( W \subset [0, 1]^m \) be a set of vectors. Consider a stochastic process \((e_1, X_1, c_1), \ldots, (e_T, X_T, c_T)\). Each \( e_t \in \{1, \ldots, m\} \setminus \{e_1, \ldots, e_{t-1}\} \), \( X_t \in \{\pm 1\} \), \( c_t \in [0, 1] \). Let \( F_t \) be the filtration generated by the random variables \((e_1, X_1, c_1), \ldots, (e_t, X_t, c_t)\), and \( F_t^* \) be the filtration generated by the random variables \((e_1, X_1, c_1), \ldots, (e_{t+1}, X_{t+1}, c_{t+1})\). The random variable \( T \in \{1, \ldots, m\} \) is a stopping time w.r.t. \( F_t \). Each \( X_t \) is a Bernoulli random variable independent of \( F_t^* \).

\[
X_t = \begin{cases} 
1, & \text{with probability } \varepsilon; \\
-1, & \text{with probability } 1 - \varepsilon;
\end{cases}
\]

where \( \varepsilon < 1/2 \). Then, for all \( \lambda > 3(1 - 2\varepsilon)^{-2} \),

\[
\Pr \left( \exists w \in W \text{ s.t. } \sum_{t=1}^{T} X_t w(e_t) c_t + \frac{1 - 2\varepsilon}{2} \sum_{t=1}^{T} w(e_t) c_t \geq 0 \text{ and } \sum_{t=1}^{T} w(e_t) c_t \geq \lambda \right) \leq 2|W|e^{-1/\lambda(1-2\varepsilon)^2} \lambda. \quad (4)
\]
Remark 4.3. If \( W \in \mathbb{R}^m \) is a \( \gamma \)-net for a set \( Z \subset \mathbb{R}^m \) in the \( \ell_\infty \) norm, if for every \( z \in Z \), there exists \( w \in W \) such that \( \|z - w\|_\infty \equiv \max_i \{|z(i) - w(i)|\} \leq \gamma \).

Definition 4.2. We say that a set \( W \subset \mathbb{R}^m \) is a \( \gamma \)-net for a set \( Z \subset \mathbb{R}^m \) in the \( \ell_\infty \) norm, if for every \( z \in Z \), there exists \( w \in W \) such that \( \|z - w\|_\infty \equiv \max_i \{|z(i) - w(i)|\} \leq \gamma \).

Remark 4.3. If \( W \) is a \( \gamma \)-net for \( Z \subset [0, 1]^m \), then there exists \( W' \subset [0, 1]^m \) of the same size as \( W \) (\(|W'| = |W|\)), such that for every \( z \in Z \), there exists \( w' \in W' \) satisfying \( w'(i) \leq z(i) \leq w'(i) + 2\gamma \) for all \( i \). To obtain \( W' \) we simply subtract \( \min(\gamma, w(i)) \) from each coordinate of \( w \) and then truncate each \( w'(i) \) at the threshold of 1.
Theorem 4.4. Consider a stochastic process \((e_1, X_1, c_1), \ldots, (e_T, X_T, c_T)\) such that each \(e_t \in \{1, \ldots, m\} \setminus \{e_1, \ldots, e_{t-1}\}\), \(X_t \in \{\pm 1\}\) and \(c_t \in [0, 1]\). Let \(\mathcal{F}_t\) be the filtration generated by the random variables \((e_1, X_1, c_1), \ldots, (e_t, X_t, c_t)\), and \(\mathcal{F}'_t\) be the filtration generated by the random variables \((e_1, X_1, c_1), \ldots, (e_{t+1}, X_{t+1}, c_{t+1})\). The random variable \(T \in \{1, \ldots, m\}\) is a stopping time w.r.t. \(\mathcal{F}_t\). Each \(X_t\) is a Bernoulli random variable independent of \(\mathcal{F}'_{t-1}\).

\[
X_t = \begin{cases} 1, & \text{with probability } \varepsilon; \\ -1, & \text{with probability } 1 - \varepsilon; \end{cases}
\]

where \(\varepsilon < 1/2\). Let \(Z \subset [0, 1]^m\) be a set of vectors having a \(\gamma\)-net in the \(L_\infty\) norm of size \(N\). Define two random sets depending on \(\{X_t\}\):

\[
Q_+ = \{t : X_t = 1\} \quad \text{and} \quad Q_- = \{t : X_t = -1\}.
\]

Then, for all \(\lambda > 3(1 - 2\varepsilon)^2\), we have

\[
\Pr \left( \exists z \in Z, \; Q_\oplus \subset Q_+ \text{ s.t. } \sum_{t \in Q_\oplus \cup Q_-} X_t z(e_t) c_t \geq 0 \quad \text{and} \quad \sum_{t \in Q_\oplus} z(e_t) c_t \geq \lambda + \frac{6\gamma}{1 - 2\varepsilon} \sum_{t \in Q_\oplus} c_t \right) \leq 2Ne^{-\lambda/2(1 - 2\varepsilon)^2}. \tag{5}
\]

Proof. Let \(W\) be a \(\gamma\)-net for \(Z\). For simplicity of exposition we subtract \(\min(\gamma, w(i))\) from all coordinates of vectors \(w \in W\). Thus, we assume that for all \(z \in Z\), there exists \(w \in W\) such that \(w(i) \leq z(i) \leq w(i) + 2\gamma\) and \(w(i) \geq 0\) for all \(i\) (see Remark \[43\]).

Suppose that for some \(z \in Z\) and \(Q_\oplus \subset Q_+\), the inequalities

\[
\sum_{t \in Q_\oplus \cup Q_-} X_t z(e_t) c_t \geq 0 \tag{6}
\]

and

\[
\sum_{t \in Q_\oplus} z(e_t) c_t \geq \lambda + \frac{6\gamma}{1 - 2\varepsilon} \sum_{t \in Q_\oplus} c_t \tag{7}
\]

hold. Pick a \(w \in W\), such that \(w(i) \leq z(i) \leq w(i) + 2\gamma\) for all \(i\). We replace \(z(e_t)\) with \(w(e_t)\) in \(\[7\]\

\[
\sum_{t \in Q_\oplus} w(e_t) c_t \geq \sum_{t \in Q_\oplus} (z(e_t) - 2\gamma) c_t \geq \lambda + \frac{4\gamma}{1 - 2\varepsilon} \sum_{t \in Q_\oplus} c_t \tag{8}
\]

Then,

\[
\sum_{t=1}^T X_t w(e_t) c_t + \frac{1 - 2\varepsilon}{2} \sum_{t=1}^T w(e_t) c_t \geq \sum_{t \in Q_\oplus \cup Q_-} X_t w(e_t) c_t + \frac{1 - 2\varepsilon}{2} \sum_{t \in Q_\oplus} w(e_t) c_t \geq \left[ \sum_{t \in Q_\oplus} (z(e_t) - 2\gamma) c_t - \sum_{Q_-} z(e_t) c_t \right] + 2\gamma \sum_{t \in Q_\oplus} c_t = \sum_{t \in Q_\oplus \cup Q_-} X_t z(e_t) c_t \geq 0. \tag{9}
\]

By Lemma \([4, 1]\) there exists a \(w \in W\) satisfying \(\[8\]\) and \(\[9\]\) with probability at most \(2Ne^{-\lambda/2(1 - 2\varepsilon)^2}\). This concludes the proof. \(\square\)
5 Epsilon Net for SDP Solutions

In order to use Theorem 3.1, we need to prove that the set of all SDP solutions to our problem has a small epsilon net. We use the following lemma from [MMVT].

Lemma 5.1 (ITCS, Lemma 2.7). For every graph $G = (V, E)$ on $n$ vertices ($V = \{1, \ldots, n\}$) with the average degree $\Delta = 2|E|/|V|$, real $M \geq 1$, and $\gamma \in (0, 1)$, there exists a set of matrices $W$ of size at most $|W| \leq \exp(O(n^{M \gamma^2} \log n))$ such that: for every collection of vectors $L(1), \ldots, L(n), R(1), \ldots R(n)$ with $\|L(u)\| = M$, $\|R(v)\| = M$ and $\langle L(u), R(v) \rangle \in [0, 1]$, there exists $W \in W$ satisfying for every $(u, v) \in E$:

$$w_{uv} \leq \langle L(u), R(v) \rangle \leq w_{uv} + \gamma;$$

$$w_{uv} \in [0, 1].$$

By letting $G$ be the complete graph, $M = 1$, $L(u) = R(u) = f(u)$, we get the following corollary.

Corollary 5.2. For every $\gamma \in (0, 1)$, there exists a set of matrices $W$ of size at most $|W| \leq \exp(O(n^{\gamma^2} \log n))$ such that: For every collection of vectors $\{f(u)\}$, there exists $W \in W$ satisfying for every $(u, v)$:

$$|w_{uv} - \langle f(u), f(v) \rangle| \leq \gamma.$$

6 Structural Theorem – Proof of Theorem 3.1

Define $f$ and $f^*$ as in (1) and (2). Recall, that the algorithm removes all edges $(u, v) \in E$ with $f(u, v) \geq 1 - \gamma$. We show that the number of edges inconsistent with the planted partition $P^*$ that are remain in the graph after the first step of the algorithm is small with high probability.

Proof of Theorem 3.1. For $(u, v) \in E$, let

$$X_{(u,v)} = \begin{cases} 1, & \text{if } (u, v) \in E_R; \\ -1, & \text{otherwise}. \end{cases}$$

Let $Q_+ = E_R$ and $Q_- = E^* \setminus E_R$. Then, $Q \subset Q_+$. Observe, that $f(u, v) = f^*(u, v)$ if $(u, v) \in E \setminus Q = Q_- \text{ and } f(u, v) = 1 - f^*(u, v)$ if $(u, v) \in Q \subset Q_+$. The SDP value is upper bounded by the optimal value $OPT$, which in turn is at most $c(Q)$. Write,

$$\text{SDP} = \sum_{(u, v) \in E} c(u, v) f(u, v) = \sum_{(u, v) \in E \setminus Q} c(u, v) f^*(u, v) + \sum_{(u, v) \in Q} c(u, v)(1 - f^*(u, v)) \leq c(Q).$$

Therefore,

$$\sum_{(u, v) \in E \setminus Q} c(u, v) f^*(u, v) \leq c(Q) - \sum_{(u, v) \in Q} c(u, v)(1 - f^*(u, v)) = \sum_{(u, v) \in Q} c(u, v)f^*(u, v).$$

We rewrite this expression as follows,

$$\sum_{(u, v) \in Q \cup Q_-} X_{(u,v)} c(u, v) f^*(u, v) \geq 0.$$ (10)
Suppose that
\[
\sum_{(u,v) \in Q : f(u,v) \leq 1 - \delta} c(u,v) \geq \lambda + \frac{6\gamma}{1 - 2\epsilon} c(Q).
\]
For \((u, v) \in Q\), \(f(u, v) = 1 - f^*(u, v)\). Thus, \(\{(u, v) \in Q : f(u,v) \leq 1 - \delta\} = \{(u,v) \in Q : f^*(u,v) \geq \delta\}\), and
\[
\sum_{(u,v) \in Q} c(u,v) f^*(u,v) \geq \delta \lambda + \frac{6\delta \gamma}{1 - 2\epsilon} c(Q).
\]
(11)
By Theorem 4.4 and Corollary 5.2, the probability that inequalities (10) and (11) hold is at most
\[
2 \exp\left(O(n\gamma^{-2}\delta^{-2}\log n)\right) \exp\left(-\frac{1}{5}(1 - 2\epsilon)^2 \delta \lambda\right) = o(1),
\]
for an appropriate choice of the constant \(C\) in the bound on \(\lambda\).

References

[ACN08] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. J. ACM, 55(5):23:1–23:27, November 2008.

[BBC04] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. Machine Learning, 56(1-3):89–113, 2004.

[BDSY99] Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. Clustering gene expression patterns. Journal of Computational Biology, 6(3/4):281–297, 1999.

[BS95] Avrim Blum and Joel Spencer. Coloring random and semi-random \(k\)-colorable graphs. J. Algorithms, 19:204–234, September 1995.

[CGW05] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. J. Comput. Syst. Sci., 71(3):360–383, October 2005.

[CSX12] Yudong Chen, Sujay Sanghavi, and Huan Xu. Clustering sparse graphs. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 2204–2212. Curran Associates, Inc., 2012.

[DEFI06] Erik D. Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. Correlation clustering in general weighted graphs. Theoretical Computer Science, 361(23):172 – 187, 2006. Approximation and Online Algorithms.

[ES09] Micha Elsner and Warren Schudy. Bounding and comparing methods for correlation clustering beyond ilp. In Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing, ILP ’09, pages 19–27, 2009.

[FK98] Uriel Feige and Joe Kilian. Heuristics for finding large independent sets, with applications to coloring semi-random graphs. In Proceedings of Symposium on Foundations of Computer Science, pages 674 –683, 1998.

[FK00] Uriel Feige and Robert Krauthgamer. Finding and certifying a large hidden clique in a semirandom graph. Random Struct. Algorithms, 16:195–208, March 2000.

[Fre75] David A Freedman. On tail probabilities for martingales. The Annals of Probability, pages 100–118, 1975.
[GVY93] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Approximate max-flow min-
(multi)cut theorems and their applications. In Proceedings of Symposium on Theory of
Computing, pages 698–707, 1993.

[KMM11] Alexandra Kolla, Konstantin Makarychev, and Yury Makarychev. How to play unique
games against a semi-random adversary: Study of semi-random models of unique games.
In Proceeding of Symposium on Foundations of Computer Science, pages 443–452, 2011.

[KV05] Subhash Khot and Nisheeth K. Vishnoi. The unique games conjecture, integrality gap
for cut problems and embeddability of negative type metrics into $\ell_1$. In Proceedings of
Symposium on Foundations of Computer Science, pages 53–62, 2005.

[MMV12] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Approx-
imation algorithms for semi-random partitioning problems. In Proceedings of Symposium
on Theory of Computing, pages 367–384, 2012.

[MMV13] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Sorting
noisy data with partial information. In Proceedings of the 4th Conference on Innovations
in Theoretical Computer Science, pages 515–528, 2013.

[MMV14] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Constant
factor approximation for balanced cut in the random PIE model. In Proceedings of
Symposium on Theory of Computing, 2014.

[MS10] Claire Mathieu and Warren Schudy. Correlation clustering with noisy input. In Proceed-
ings of Symposium on Discrete Algorithms, pages 712–728, 2010.

[Swa04] Chaitanya Swamy. Correlation clustering: Maximizing agreements via semidefinite pro-
gramming. In Proceedings of Symposium on Discrete Algorithms, pages 526–527, 2004.

[WS11] David P. Williamson and David B. Shmoys. The Design of Approximation Algorithms.
Cambridge University Press, New York, NY, USA, 1st edition, 2011.