Semirandom Models: Correlation Clustering

Instructor: Yury Makarychev, TTIC

Max Planck ADFOCS 2023

Two approaches to Modelling for Real-life Instances

Assume that an instance satisfies certain structural properties:

- Perturbation Resilience
- Assumptions of the graph, weights, etc

Generative models. Assume that an instance is generated in a certain way:

- Random models: e.g. G is a $G(n, p)$ graph
- Semirandom models: random + adversarial choices

Two Approaches to Modelling Real-life Instances

Assume that an instance satisfies certain structural properties:

- Perturbation Resilience
- Assumptions of the graph, weights, etc

Generative models. Assume that an instance is generated in a certain way:

- Random models: e.g. G is a $G(n,p)$ graph
- Semirandom models: random + adversarial choices

Semirandom Models

There are algorithms for semirandom models of graph partitioning, graph coloring, community detection, sorting noisy data, constraint satisfaction, and other problems.

Today: semirandom instances for correlation clustering

Roadmap

Introduction

- Define Correlation Clustering & a semirandom model
- Review known results

Algorithm

- Solve an SDP relaxation
- Remove edges with high SDP cost
- Prove the Main Structural Algorithm, which claims that the remaining problem is "easy"
- Construct a small set of representative solutions

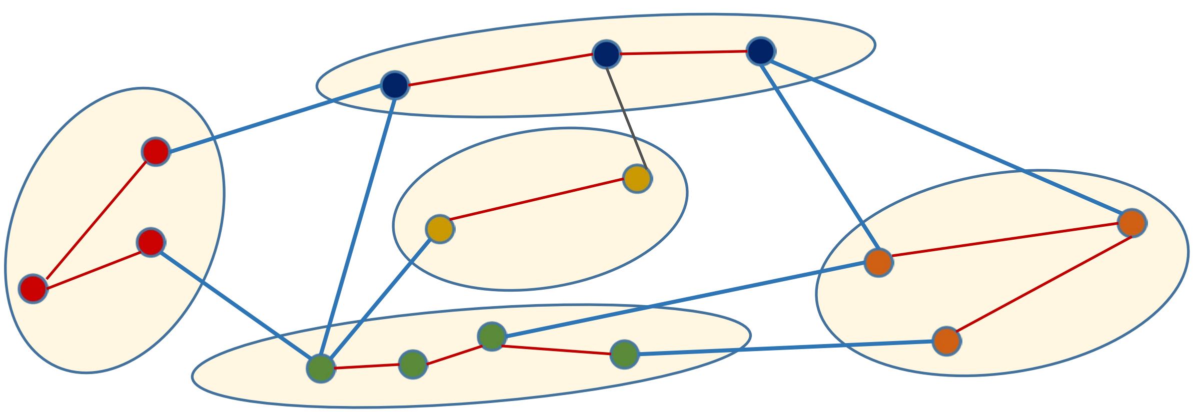
Correlation Clustering

We are given a graph $G = (V, E, c)$ with edge costs c_e and edge labels.

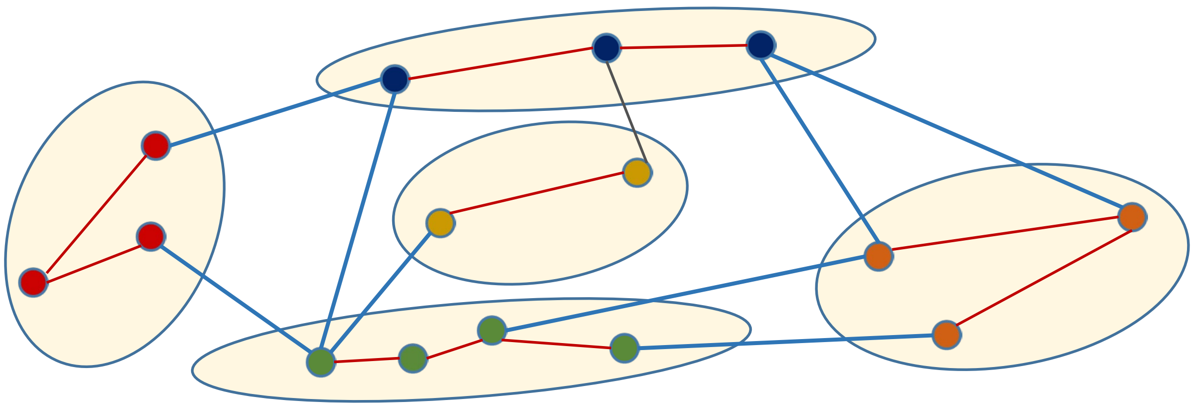
- V is the set of datapoints/vertices
- for $(u, v) \in E$, we are given whether

u and v are similar or dissimilar

and the confidence level $c_e \in [0,1]$



Correlation Clustering



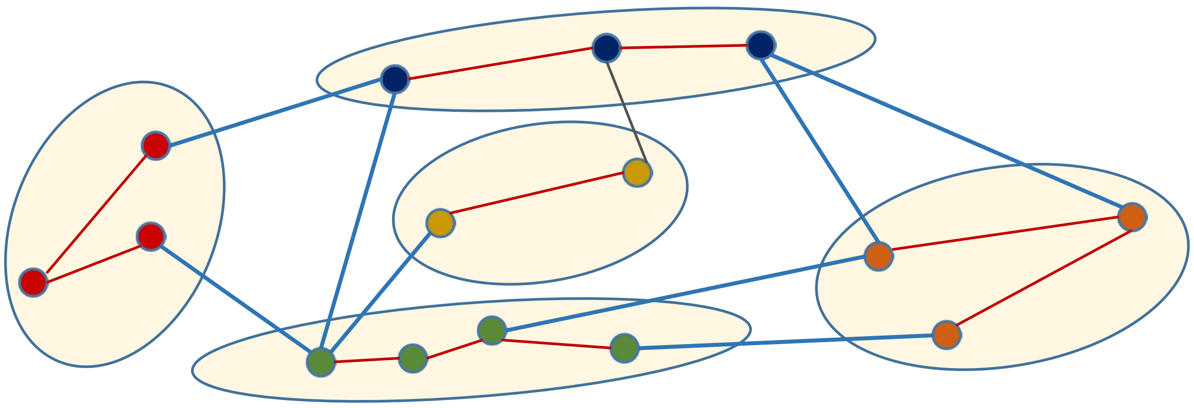
$$E = E_{+} \sqcup E_{-}$$

"+" edges connect similar vertices

"−" edges connect dissimilar vertices

$c_{uv} \in [0,1]$ is the confidence level

Perfect Information

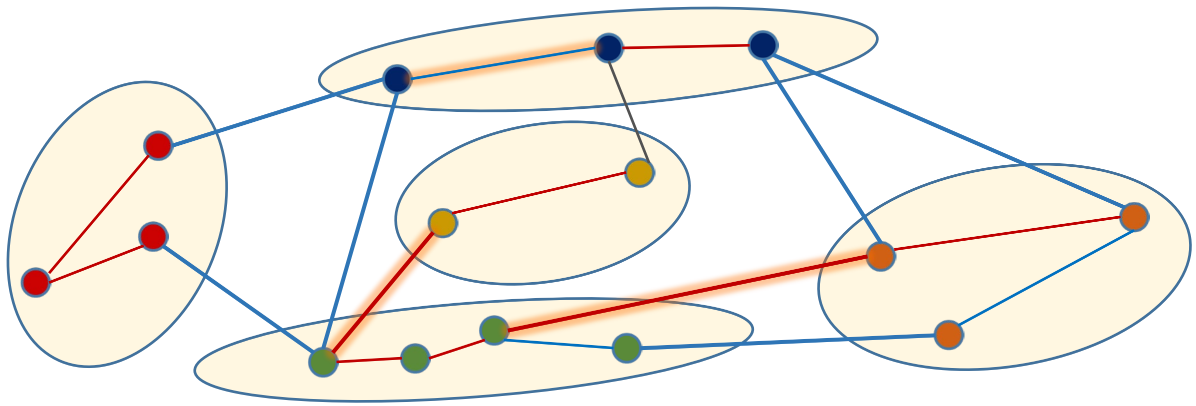


Perfect Information

There is a clustering $C_1, \ldots, C_k$ such that all

- $+$-edges lie within clusters
- $-$-edges connect different clusters

Imperfect Information



Reality: Some edges are inconsistent with clustering

Objective: Find a clustering that minimizes the total cost of edges inconsistent with it

Semirandom Model

Adversarial choices:

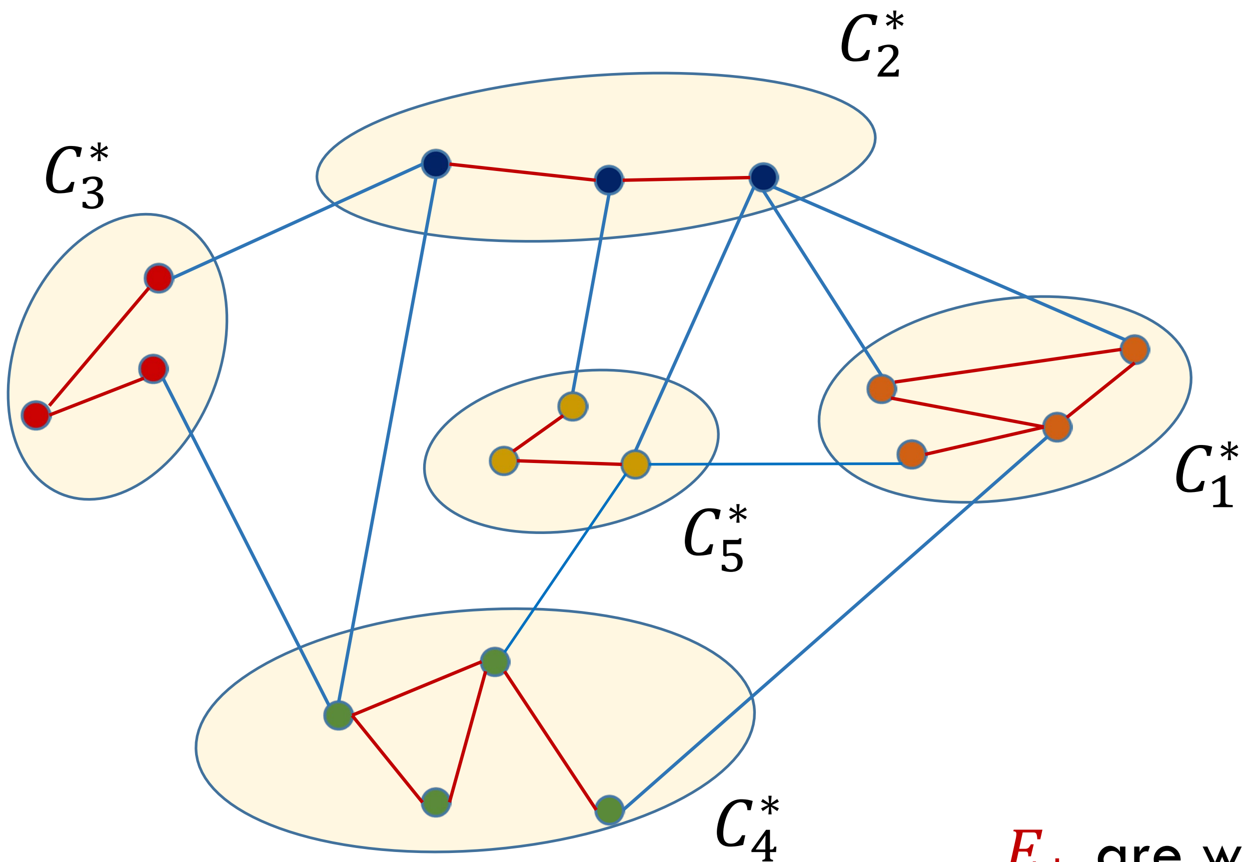
- Choose a planted clustering $C_1^*, \ldots C_k^*$
- Choose a graph $G = (V, E)$ and edge costs c_e
- Label edges within clusters with $+$,
  labels across clusters with $-$.

At this point, we have perfect information.

Random corruption:

- Flip the label $+ \leftrightarrow -$ of every edge w.p. $\varepsilon < {}^1/_2$.

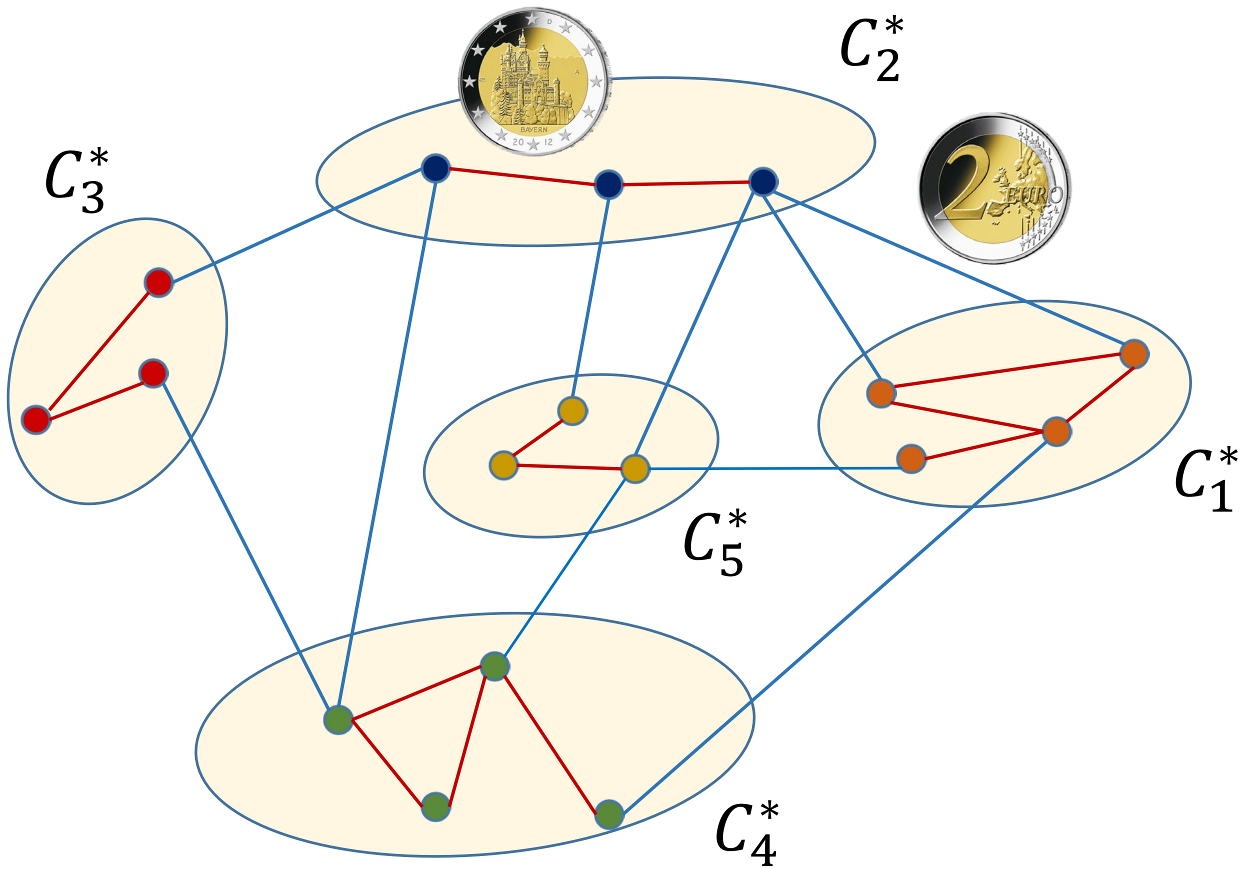
Semirandom: Planted Solution



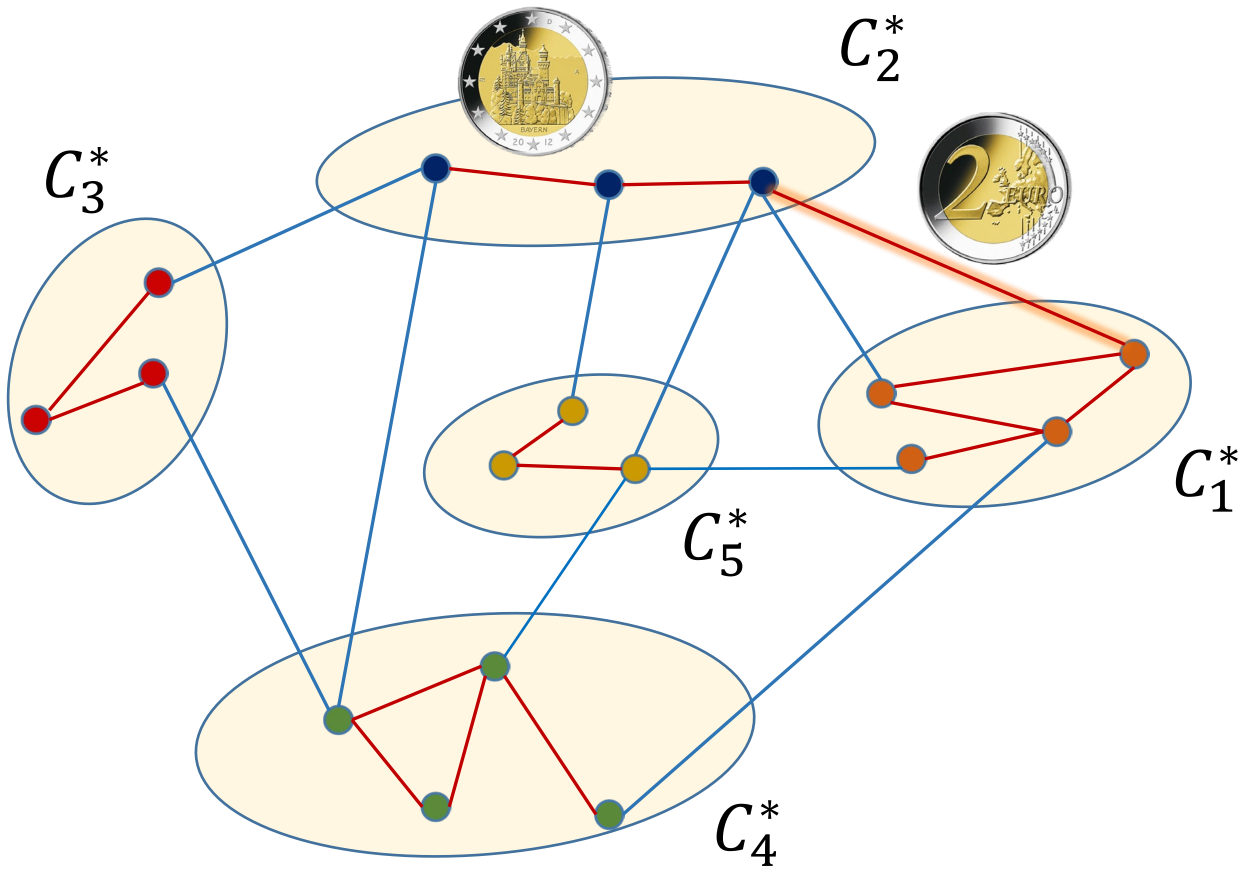
E_+ are within clusters

E_- are across clusters

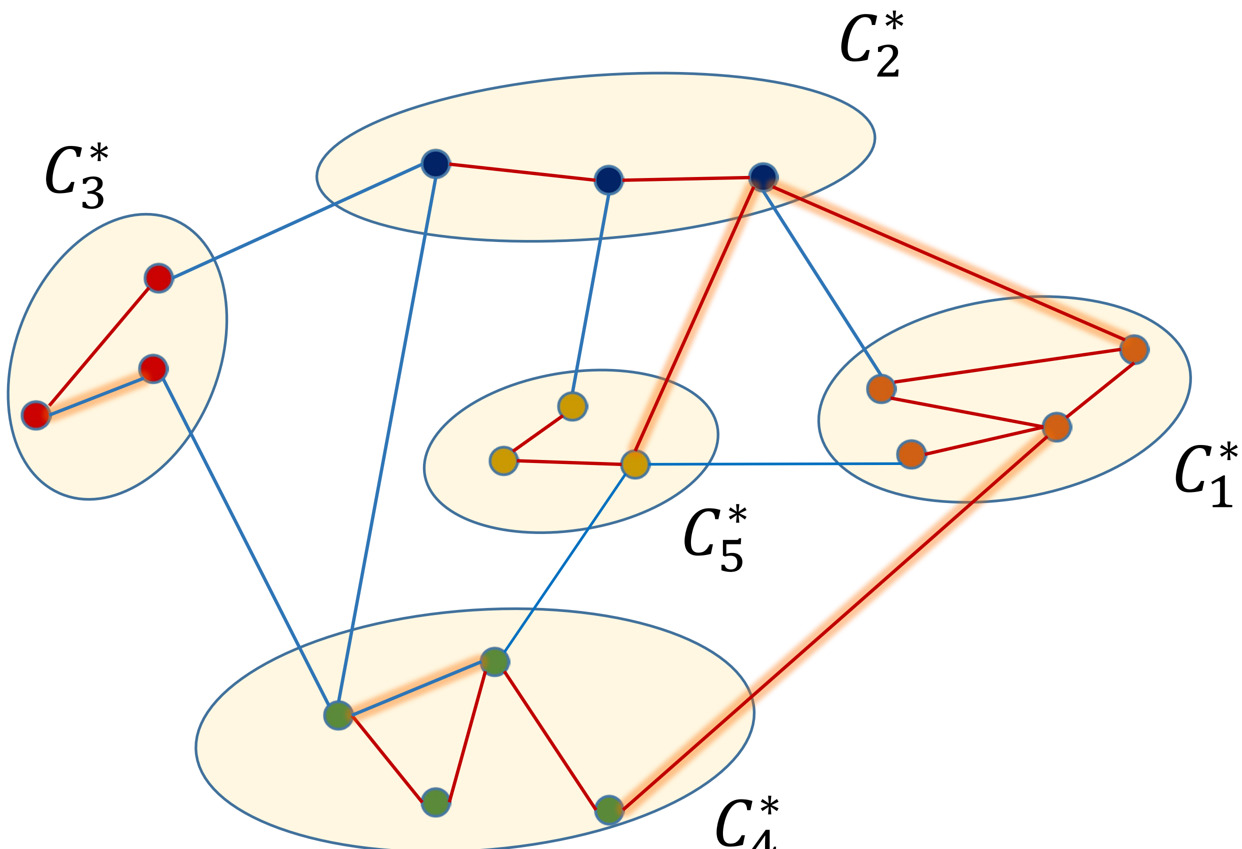
Semirandom: Random Corruption



Semirandom: Random Corruption



Semirandom: Random Corruption



Results

Results: Worst Case

Arbitrary graph with arbitrary costs c_e

[Charikar, Guruswami, and Wirth '05]

[Demaine, Emanuel, Fiat, and Immorlica '06]

$O(\log n)$ approximation

Completete graph with unit costs $c_e = 1$

[Cohen-Addad, Lee, and Newman '23]

$1.994\ldots$ approximation

Completete graph with costs $c_e \in [\alpha, 1]$

[Jafarov, Kalhan, Makarychev, and M '20]

$3 + 2\log_e 1/\alpha$ approximation

Results: Random & Semi-random Models

[Ben-Dor, Shamir, and Yakhini '99] [Bansal, Blum, and Chawla '04] [Mathieu and Schudy '10] [Chen, Jalali, Sanghavi, and Xu '14]

Algorithms for complete and $G(n, p)$ graphs

[Makarychev, M, Vijayaraghavan '14] An algorithm for arbitrary graphs which finds a solution of cost

$$(1 + \delta)OPT + O(n \operatorname{polylog} n)$$

This is a PTAS when $OPT \gg \varepsilon^{-1} n \operatorname{polylog} n$. The algorithm recovers the planted solution under mild expansion assumptions on G.

SDP relaxation

Introduce a variable x_{uv} for every pair of vertices.

The intended solution is

$$x_{uv} = \begin{cases} 1, \text{if } u \text{ and } v \text{ are in the same cluster} \\ 0, \text{if } u \text{ and } v \text{ are in different clusters} \end{cases}$$

$$X = (x_{uv}) = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

block matrix

SDP relaxation

minimize $\sum_{e \in E_+} (1 - x_e) + \sum_{e \in E_-} x_e$

s.t.

$X = (x_{uv}) \succcurlyeq 0$ (positive semidefinite)

$0 \leq x_{uv} \leq 1$

Assume $c_e = 1$ to simplify the exposition.

SDP relaxation

minimize $\sum_{e \in E_+} (1 - x_e) + \sum_{e \in E_-} x_e$

s.t.

$$X = (x_{uv}) \succcurlyeq 0 \qquad \text{(positive semidefinite)}$$

$$0 \leq x_{uv} \leq 1$$

Let $f_e(x_e) = 1 - x_e$ if $e \in E_+$ and

$f_e(x_e) = x_e$ if $e \in E_-$

SDP relaxation

minimize $\sum_{e \in E} f_e(x_e)$

s.t.

$X = (x_{uv}) \succcurlyeq 0$ (positive semidefinite)

$0 \le x_{uv} \le 1$

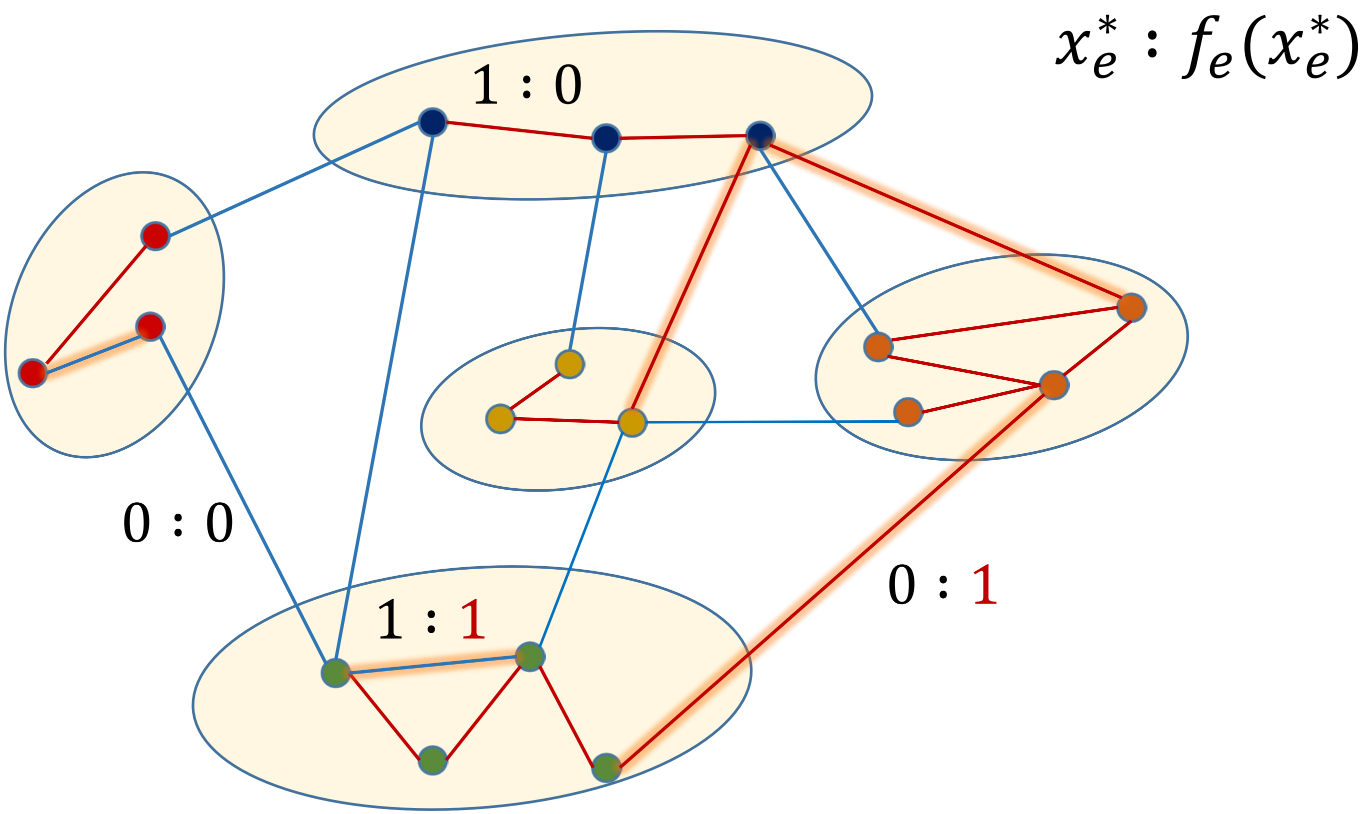
Let $f_e(x_e) = 1 - x_e$ if $e \in E_+$ and

$f_e(x_e) = x_e$ if $e \in E_-$

What is $f_e(x_e^*)$?

$f_e(x_e) = 1 - x_e$ if $e \in E_+$
$f_e(x_e) = x_e$ if $e \in E_-$

Q: Let x_e^* be the planted solution. What is $f_e(x_e^*)$?



SDP relaxation

minimize $\sum_{e \in E} f_e(x_e)$

s.t.

$X = (x_{uv}) \succcurlyeq 0$ (positive semidefinite)

$0 \leq x_{uv} \leq 1$

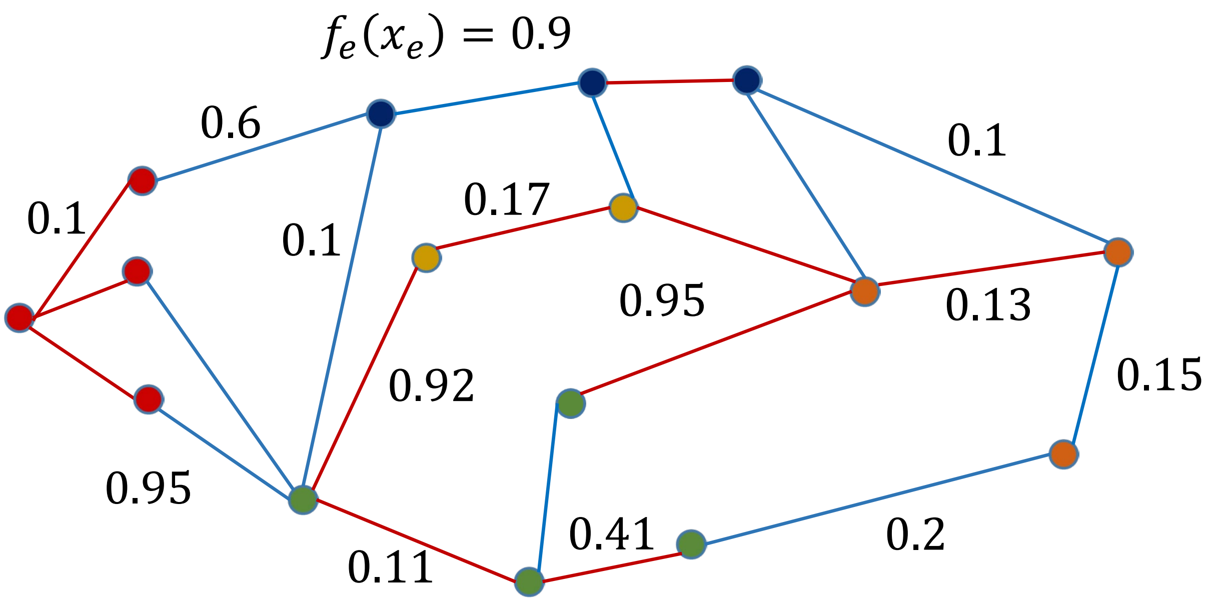
Let $f_e(x_e) = 1 - x_e$ if $e \in E_+$ and

$f_e(x_e) = x_e$ if $e \in E_-$

Denote the cost of the planted solution by OPT.

Algorithm

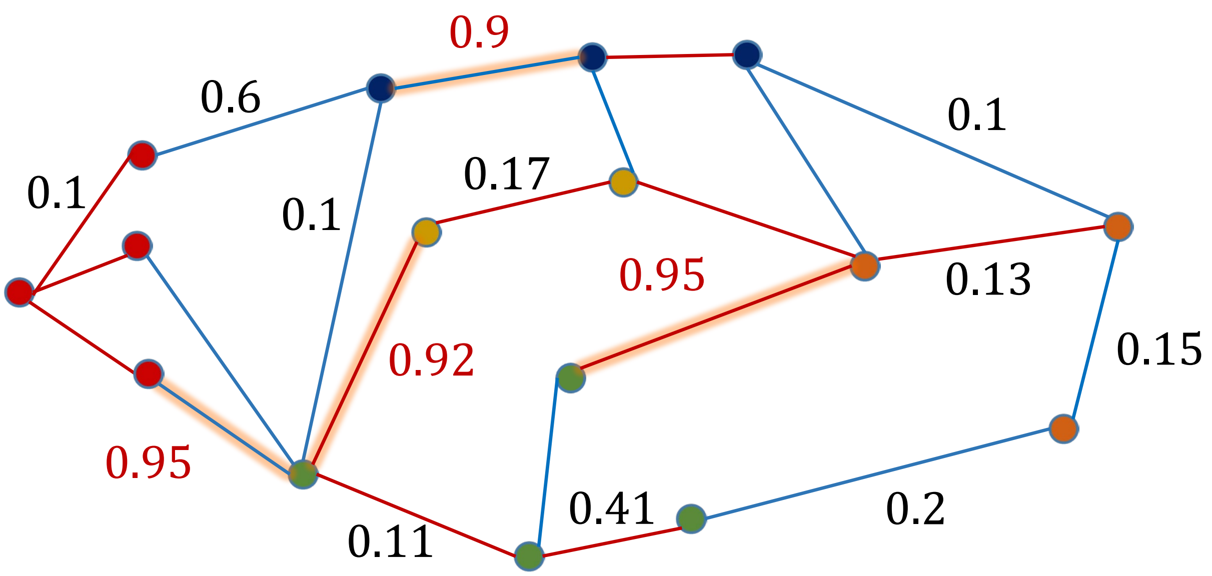
Step 0: solve the SDP, obtain $X = (x_{uv})$ and $f_e(x_e)$



Algorithm

Step 0: solve the SDP, obtain $X = (x_{uv})$ and $f_e(x_e)$

Step 1: remove all edges e with $f_e(x_e) \geq 1 - \delta$



Algorithm

Step 0: solve the SDP, obtain $X = (x_{uv})$ and $f_e(x_e)$

Step 1: remove all edges e with $f_e(x_e) \geq 1 - \delta$

Q: What is the total cost $Cost_1$ of all removed edges?

A: A contribution of a removed edge e

- to $Cost_1$ is 1
- to SDP is $\geq 1 - \delta$

$\Rightarrow Cost_1 \leq \frac{SDP}{1-\delta} \leq (1 + 2\delta)SDP$

Algorithm

It turns out that we removed most corrupted edges!

Main Structural Theorem: W.h.p. the cost $Cost_2$ of the remaining corrupted edges is at most

$$Cost_2 \leq \frac{\delta \, OPT}{D} + O_\delta\left(n \log^3 n\right)$$

where $D = O(\log n)$.

Step 2: Apply a standard $D = O(\log n)$ approximation algorithm to the remaining instance

[Charikar, Guruswami, Wirth '05; Demaine, Emanuel, Fiat, Immorlica '06]

Assume the Structural Theorem

We obtain a clustering whose cost is at most

$$\left(\frac{\delta\, OPT}{D} + O\left(n \log^3 n\right)\right) \times D = \delta\, OPT + O_\delta(n \log^4 n)$$

Taking into account $Cost_1$, we upper bound the cost of all the edges:

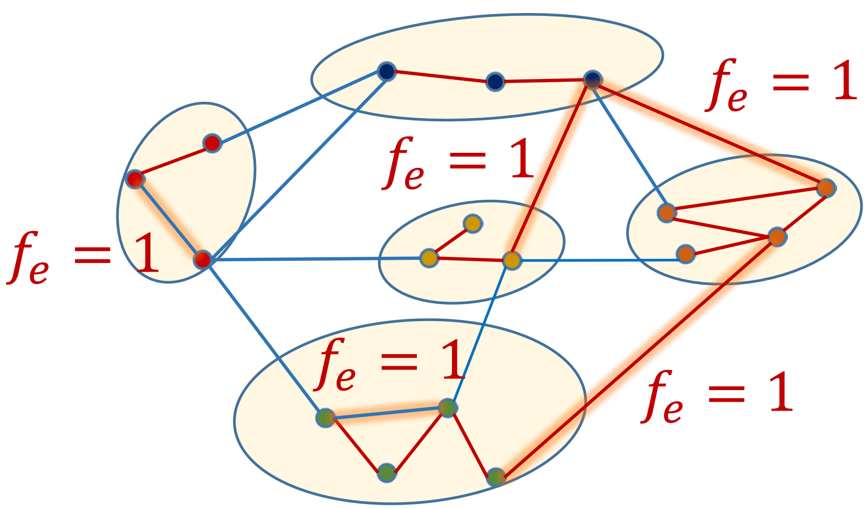
$$(1 + 3\delta)OPT + O_\delta(n \log^4 n)$$

☺

Structural Theorem

Main Structural Theorem: W.h.p. the cost $Cost_2$ of the remaining corrupted edges is at most $\frac{\delta\, OPT}{D} +$ $O_\delta\left(n \log^3 n\right)$ where $D = O(\log n)$.

Q: What is $Cost_2$ for the planted solution x_e^*?



Idea: There are few integrality gap examples

The SDP solution maybe

☺ Close to the planted solution – Good news!
Step 1 removes most corrupted edges.

☹ Far from X^* – Too bad!
Step 1 might not accomplish much.

If a feasible SDP solution is far from X^*,

- its cost before corruption is much larger than that of X^*
- the expected cost after corruption is also much larger than that of X^*
- Bernstein's concentration inequality ⇒ unlikely that

$$SDP \leq OPT$$

Idea: There are few integrality gap examples

If a feasible SDP solution is far from X^*,

- its cost before corruption is much larger than that of X^*
- the expected cost after corruption is also much larger than that of X^*
- Bernstein's concentration inequality ⇒ unlikely that
$$SDP \leq OPT$$
- W.h.p. there is no feasible SDP solution that is far from X^* and whose value $SDP \leq OPT$

  ⇒ the optimal SDP solution must be close to X^*

Structural Theorem

Choose $G = (V, E, c_e)$ and clustering $C_1^*, \ldots, C_k^*$. The clustering defines edge labels.

Random Step: Flip the label of every e w.p. $\varepsilon < {}^1\!/_2$

SDP Step: Find an optimal SDP solution

Let E_R be the set of randomly corrupted edges.

Need to show that

$$Cost_2 = |\{e \in E_R : f_e(x_e) < 1 - \delta\}|$$

is small.

Structural Theorem

Choose $G = (V, E, c_e)$ and clustering $C_1^*, \dots, C_k^*$. The clustering defines edge labels.

Random Step: Flip the label of every e w.p. $\varepsilon < {}^1/_2$

SDP Step: Find an SDP solution with $SDP \le OPT$

Let E_R be the set of randomly corrupted edges.

Need to show that

$$Cost_2 = |\{e \in E_R : f_e(x_e) < 1 - \delta\}|$$

is small.

A game between SDP & Random

Think that the SDP solution is chosen by an adversary who wants to disprove our theorem.

Random Player: Flip the label of every e w.p. $\varepsilon < {}^1\!/_2$

SDP Player: Choose a feasible SDP solution

SDP Player wins ☹ if

$$SDP \leq OPT \text{ and } Cost_2 \text{ is large.}$$

We will show that SDP wins with exponentially small probability.

A game between SDP & Random

Think that the SDP solution is chosen by an adversary who wants to disprove our theorem.

SDP Player: Choose a feasible SDP solution

Random Player: Flip the label of every e w.p. $\varepsilon < {}^{1}/_{2}$

SDP Player wins ☹ if

$$SDP \leq OPT \text{ and } Cost_2 \text{ is large.}$$

We will show that SDP wins with exponentially small probability.

Game: SDP

SDP Player:

Choose a feasible SDP solution: $X = (x_{uv}) \succcurlyeq 0$

When SDP chooses X, E_R and f_e are not yet defined.

Let $f_e^*(x_e) = 1 - x_e$ if e is in some C_i^*

$f_e^*(x_e) = x_e$ otherwise

Think of $bet_e \equiv f_e^*(x_e) \in [0,1]$ as a bet that SDP places on edge e.

Game: SDP

SDP Player:

Choose a feasible SDP solution: $X = (x_{uv}) \succcurlyeq 0$

Define $f_e^*(x_e) = 1 - x_e$ if e is in some C_i^*

$f_e^*(x_e) = x_e$ otherwise

Think of $bet_e \equiv f_e^*(x_e) \in [0,1]$ as a bet that SDP places on edge e.

Q: What bet $f_e^*(x_e^*)$ does the planted solution x_e^* place on every edge?

Game: SDP

SDP Player:

Choose a feasible SDP solution: $X = (x_{uv}) \succcurlyeq 0$

Define $f_e^*(x_e) = 1 - x_e$ if e is in some C_i^*

$f_e^*(x_e) = x_e$ otherwise

Think of $bet_e \equiv f_e^*(x_e) \in [0,1]$ as a bet that SDP places on edge e.

Q: What bet $f_e^*(x_e^*)$ does the planted solution x_e^* place on every edge?

A: $f_e^*(x_e^*) = 0$. Further, $f_e^*(x_e) = |x_e - x_e^*|$ shows by how much x_e deviates from x_e^*.

Game: Random

Random Player:

Flips the label of each e w.p. $\varepsilon < {}^{1}\!/_{2}$.

Let $Z_e = 1$ if $e \in E_R$ (that is, was flipped by Random)

and $Z_e = -1$ otherwise.

$$\mathbb{E}[Z_e] = \varepsilon \cdot 1 + (1 - \varepsilon) \cdot (-1) = 2\varepsilon - 1 < 0$$

A game between SDP & Random

SDP Player: Places a bet $bet_e \equiv f_e^*(x_e)$ on each e

Random Player: flips a biased ± 1 "coin" Z_e with

$$\mathbb{E}[Z_e] = \varepsilon \cdot 1 + (1-\varepsilon) \cdot (-1) = 2\varepsilon - 1 < 0$$

SDP Player wins ☹ only if

$OPT - SDP \geq 0$

$Cost_2$ is large

$f_e(x_e) + f_e^*(x_e) = 1$ if $e \in E_R$

Formula for $OPT - SDP$

$$OPT - SDP = \sum_e f_e(x_e^*) - f(x_e)$$

If $e \notin E_R$

$$f_e(x_e^*) - f_e(x_e) = -f_e(x_e) = -f_e^*(x_e) = Z_e f^*(x_e)$$

If $e \in E_R$

$$f_e(x_e^*) - f_e(x_e) = 1 - f_e(x_e) = 1 - \big(1 - f_e^*(x_e)\big)$$
$$= f_e^*(x_e) = Z_e f^*(x_e)$$

$f_e(x_e) + f_e^*(x_e) = 1$ if $e \in E_R$

Formula for $OPT - SDP$

$$OPT - SDP = \sum_e Z_e f^*(x_e)$$

If $e \notin E_R$

$$f_e(x_e^*) - f_e(x_e) = -f_e(x_e) = -f_e^*(x_e) = Z_e f^*(x_e)$$

If $e \in E_R$

$$f_e(x_e^*) - f_e(x_e) = 1 - f_e(x_e) = 1 - \left(1 - f_e^*(x_e)\right)$$
$$= f_e^*(x_e) = Z_e f^*(x_e)$$

$f_e(x_e) + f_e^*(x_e) = 1$ if $e \in E_R$

Upper Bound for $Cost_2$

$$Cost_2 = |\{e \in E_R : f_e(x_e) < 1 - \delta\}|$$

$$= |\{e \in E_R : f_e^*(x_e) > \delta\}| \leq \sum_e \frac{f^*(x_e)}{\delta}$$

A game between SDP & Random

SDP Player: Places a bet $bet_e \equiv f_e^*(x_e)$ on each e

Random Player: flips a biased ± 1 "coin" Z_e with

$$\mathbb{E}[Z_e] = \varepsilon \cdot 1 + (1-\varepsilon)\cdot(-1) = 2\varepsilon - 1 < 0$$

SDP Player wins ☹ only if

$$OPT - SDP = \sum_e f^*(x_e) \cdot Z_e \geq 0$$

$$Cost_2 \leq \frac{1}{\delta} \sum_e f^*(x_e) \text{ is large}$$

A game between SDP & Random

SDP Player wins only if

$$OPT - SDP = \sum_e f^*(x_e) \cdot Z_e \geq 0$$

$Cost_2 \leq \frac{1}{\delta} \sum_e bet_e$ is large

But…

$$\mathbb{E}\left[\sum f^*(x_e) \cdot Z_e\right] = \sum_e (2\varepsilon - 1) f^*(x_e) = (2\varepsilon - 1) \sum_e f^*(x_e) < 0$$

Bernstein's Inequality:

$$\Pr\left(\sum f^*(x_e) \cdot Z_e \geq 0\right) = \exp\left(-\Omega(1 - 2\varepsilon) \sum_e f^*(x_e)\right)$$

is exponentially small when $\sum_e f^*(x_e)$ is large!

A game between SDP & Random

The probability that a given SDP solution wins is exponentially small ☺.

In reality, we solve the SDP after – not before – edge labels are randomly perturbed. What shall we do about that?

Random moves first & SDP second

Changing the order of moves may appear problematic: in a casino, if we were allowed to place a bet after we see where the ball lands, we could easily win!

If X could be any matrix with $x_{uv} \in [0,1]$ then the SDP player could win by defining x_e so that

$$f_e^*(x_e) = \begin{cases} 0, & \text{if } Z_e = -1 \\ 1, & \text{if } Z_e = 1 \end{cases}$$

Then,

$$\sum_e f^*(x_e) \cdot Z_e = \sum_{e:Z_e=1} c_e \approx \varepsilon\, c(E) > 0$$

$\sum_e f^*(x_e) \approx \varepsilon\, c(E)$ is large

Random moves first & SDP second

We showed that every fixed "strategy" $X = (x_{uv})$ wins with exponentially small probability

$$p = \exp(-P)$$

Union bound: the SDP player still looses w.h.p. if he chooses $X = (x_{uv})$ from an exponentially large family of solutions $|\mathcal{F}| = \exp(F)$ as long as

$$P \gg F$$

To conclude, we show that there exists a representative family of SDP solutions of size $\exp\Big(O\big(n\log^3 n\big)\Big)$.

Grothendieck Inequality

Given:

- vectors $u_1, \dots, u_n$ and $v_1, \dots, v_n$ with $\|u_i\|, \|v_j\| \leq 1$
- a matrix $M = (m_{ij})$

There exist $a_1, \dots, a_n \in \{\pm 1\}$ and $b_1, \dots, b_n \in \{\pm 1\}$ s.t.

$$\sum_{ij} m_{ij} \langle u_i, v_j \rangle \leq K_G \sum_{ij} m_{ij}\, a_i b_j$$

where $K_G < 1.7823$ is an absolute constant.

Grothendieck Inequality: Dual Form

Let $S = \left\{ ab^T = \begin{pmatrix} a_1 b_1 & \cdots & a_n b_1 \\ \vdots & \ddots & \vdots \\ a_1 b_n & \cdots & a_n b_n \end{pmatrix} : a_i, b_j \in \{\pm 1\} \right\}$

For vectors $u_1, \dots, u_n$ and $v_1, \dots, v_n$ with $\|u_i\|, \|v_j\| \leq 1$, we have for their Gram matrix:

$$G = \left(\langle u_i, v_j \rangle \right)_{ij} \in K_G \cdot \mathrm{conv}(S)$$

Grothendieck Inequality: Dual Form

Let $S = \left\{ ab^T = \begin{pmatrix} a_1 b_1 & \cdots & a_n b_1 \\ \vdots & \ddots & \vdots \\ a_1 b_n & \cdots & a_n b_n \end{pmatrix} : a_i, b_j \in \{\pm 1\} \right\}$

If $X \succcurlyeq 0$ and diagonal entries $x_{ii} \leq 1$, then

$$X \in K_G \cdot \mathrm{conv}(S)$$

Grothendieck Inequality: Dual Form

Let $S = \left\{ ab^T = \begin{pmatrix} a_1 b_1 & \cdots & a_n b_1 \\ \vdots & \ddots & \vdots \\ a_1 b_n & \cdots & a_n b_n \end{pmatrix} : a_i, b_j \in \{\pm 1\} \right\}$

If $X \succcurlyeq 0$ and diagonal entries $x_{ii} \leq 1$, then

$$X \in K_G \cdot \mathrm{conv}(S)$$

S has size $|S| = 2^{2n}$.

Grothendieck Inequality: Dual Form

If $X \succcurlyeq 0$ and diagonal entries $x_{ii} \leq 1$, then

$$X \in K_G \cdot \mathrm{conv}(S)$$

S has size $|S| = 2^{2n}$.

Approximate Carathéodory's Theorem [Maurey]:

Every X is approximated by an average of $k = O\left(\frac{\log n}{\gamma^2}\right)$ matrices* from S with ℓ_∞-error $\leq \gamma$.

* with repetitions

Grothendieck Inequality: Dual Form

If $X \succcurlyeq 0$ and diagonal entries $x_{ii} \leq 1$, then

$$X \in K_G \cdot \mathrm{conv}(S)$$

S has size $|S| = 2^{2n}$.

Approximate Carathéodory's Theorem [Maurey]:

Every X is approximated by an average of $k = O\left(\frac{\log n}{\gamma^2}\right)$ matrices* from S with ℓ_∞-error $\leq \gamma$.

Let $\mathrm{F} = \{\frac{M_1 + \cdots + M_k}{k} : M_i \in F\}$.

* with repetitions

Grothendieck Inequality: Dual Form

Let $\mathrm{F} = \left\{\frac{M_1+\cdots+M_k}{k} : M_i \in S\right\}$ where $k = O\left(\frac{\log n}{\gamma^2}\right)$.

Every feasible SDP solution X is approximated by a matrix $M \in \mathrm{F}$: $\|X - M\|_\infty \le \gamma$. We need $\gamma = \frac{1}{\log n}$.

There are $|\mathrm{F}| = |S|^k = 2^{O\left(\frac{n \log n}{\gamma^2}\right)} = \exp\left(O\left(n \log^3 n\right)\right)$ such matrices.

We are done!

Only need to take care of the error term γ. This is a bit technical but not difficult step.