CS-E4075 - Special Course in Machine Learning, Data Science and Artificial Intelligence D: Signed graphs: spectral theory and applications

Correlation clustering

Bruno Ordozgoiti

Aalto University 2021

Outline

An informal introduction to computational complexity and approximation algorithms

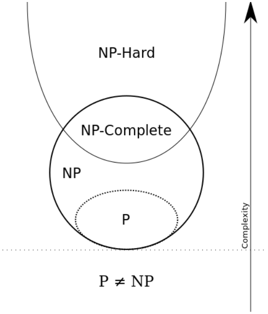


Image: Behnam Esfahbod

- Problems in P: can be solved in polynomial time ($\mathcal{O}(n^c)$ for some constant c).

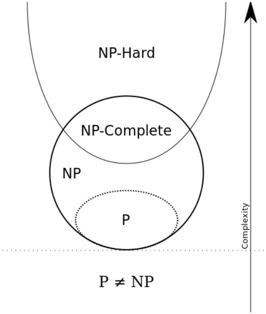
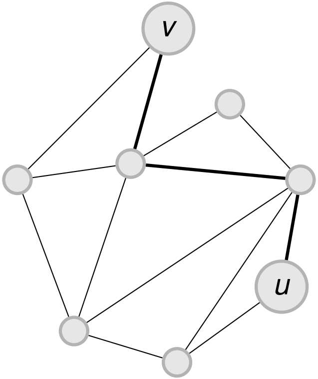


Image: Behnam Esfahbod



Is there a path of length at most k between u and v?
Can be answered computing shortest paths in $\mathcal{O}(n^2)$.

- Problems in P: can be solved in polynomial time ($\mathcal{O}(n^c)$ for some constant c).
- Problems in NP: given a solution, we can verify it in polynomial time.

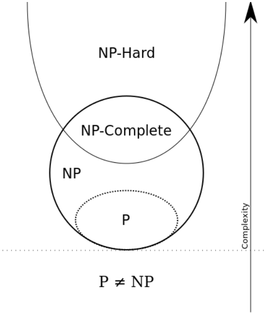
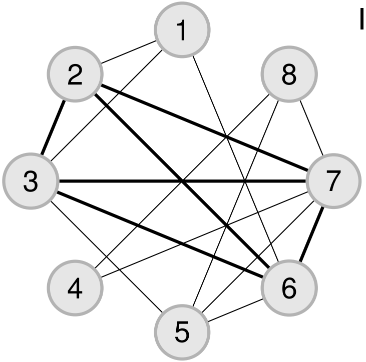


Image: Behnam Esfahbod



Is there a clique of size at least k?

- Problems in P: can be solved in polynomial time ($\mathcal{O}(n^c)$ for some constant c).
- Problems in NP: given a solution, we can verify it in polynomial time.
- Problems in NP-hard: at least as hard as any problem in NP.
  - Working assumption: no polynomial-time algorithm exists.

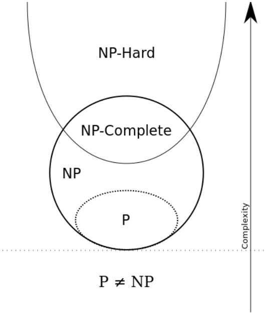
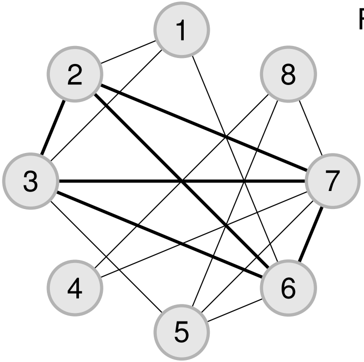


Image: Behnam Esfahbod



Find the largest clique.

Approximation algorithms

For problems in NP-hard, we know we cannot hope to find the opimal solution in polynomial time.

[1] For minimization problems, a c-approximation algorithm satisfies $ALG \leq c \cdot OPT$.

Approximation algorithms

For problems in NP-hard, we know we cannot hope to find the opimal solution in polynomial time.

But can we find a solution close to the optimum?

[1] For minimization problems, a c-approximation algorithm satisfies $ALG \leq c \cdot OPT$.

Approximation algorithms

For problems in NP-hard, we know we cannot hope to find the opimal solution in polynomial time.

But can we find a solution close to the optimum?

Definition

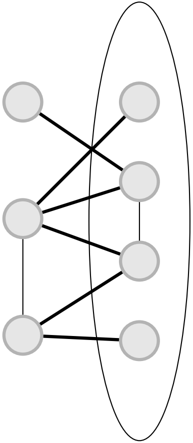
Consider a maximization[1] problem Π with optimal solution of value OPT. We say that an algorithm is a c-approximation algorithm for Π if it outputs a solution of value ALG that satisfies

$$ALG \geq c \cdot OPT.$$

[1] For minimization problems, a c-approximation algorithm satisfies $ALG \leq c \cdot OPT$.

Approximation algorithms

Example: MAXCUT

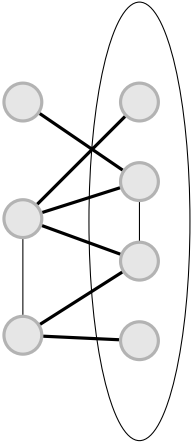


Goal: partition the vertices in two sets to maximize the number of cut edges.

NP-hard problem.

Approximation algorithms

Example: MAXCUT



Goal: partition the vertices in two sets to maximize the number of cut edges.

NP-hard problem.
However, there is a polynomial time algorithm achieving

$$ALG \geq c \cdot OPT,$$

where $c \approx 0.878$.

Consider an NP-hard problem Π. Assume there is an algorithm that runs in $\mathcal{O}(n^2)$ and satisfies $ALG \geq \frac{1}{2}OPT = \left(1 - \frac{1}{2}\right) OPT$.

Consider an NP-hard problem Π. Assume there is an algorithm that runs in $\mathcal{O}(n^2)$ and satisfies $ALG \geq \frac{1}{2}OPT = \left(1 - \frac{1}{2}\right) OPT$.

Perhaps it is possible to improve the quality of the solution by using more time, say run for $\mathcal{O}(n^3)$ and get $ALG \geq \left(1 - \frac{1}{3}\right) OPT$.

Approximation algorithms

PTAS

Consider an NP-hard problem Π. Assume there is an algorithm that runs in $\mathcal{O}(n^2)$ and satisfies $ALG \geq \frac{1}{2}OPT = \left(1 - \frac{1}{2}\right) OPT$.

Perhaps it is possible to improve the quality of the solution by using more time, say run for $\mathcal{O}(n^3)$ and get $ALG \geq \left(1 - \frac{1}{3}\right) OPT$.

Or run for $\mathcal{O}(n^4)$ and get $ALG \geq \left(1 - \frac{1}{4}\right) OPT$; run for $\mathcal{O}(n^5)$ and get $ALG \geq \left(1 - \frac{1}{5}\right) OPT$...

Approximation algorithms

PTAS

Consider an NP-hard problem Π. Assume there is an algorithm that runs in $\mathcal{O}(n^2)$ and satisfies $ALG \geq \frac{1}{2}OPT = \left(1 - \frac{1}{2}\right) OPT$.

Perhaps it is possible to improve the quality of the solution by using more time, say run for $\mathcal{O}(n^3)$ and get $ALG \geq \left(1 - \frac{1}{3}\right) OPT$.

Or run for $\mathcal{O}(n^4)$ and get $ALG \geq \left(1 - \frac{1}{4}\right) OPT$; run for $\mathcal{O}(n^5)$ and get $ALG \geq \left(1 - \frac{1}{5}\right) OPT$...

In general, maybe we can run for $\mathcal{O}(n^{1/\epsilon})$ and get $ALG \geq (1 - \epsilon) OPT$.

Consider an NP-hard problem Π. Assume there is an algorithm that runs in $\mathcal{O}(n^2)$ and satisfies $ALG \geq \frac{1}{2}OPT = \left(1 - \frac{1}{2}\right) OPT$.

Perhaps it is possible to improve the quality of the solution by using more time, say run for $\mathcal{O}(n^3)$ and get $ALG \geq \left(1 - \frac{1}{3}\right) OPT$.

Or run for $\mathcal{O}(n^4)$ and get $ALG \geq \left(1 - \frac{1}{4}\right) OPT$; run for $\mathcal{O}(n^5)$ and get $ALG \geq \left(1 - \frac{1}{5}\right) OPT$...

In general, maybe we can run for $\mathcal{O}(n^{1/\epsilon})$ and get $ALG \geq (1 - \epsilon) OPT$.

PTAS

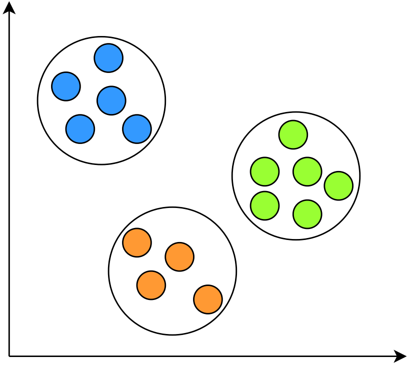
This is called a Polynomial-Time Approximation Scheme (PTAS).
Requisite: for fixed ϵ, the algorithm gives a $(1 - \epsilon)$-approximation and runs in time polynomial in n. Not always possible!

Introduction to correlation clustering

Correlation clustering

k-means clustering. Input: $X = \{x_i : i = 1, \ldots, n\}$.

Objective: Find k-partition of X to minimize $\sum_i d(x_i, c(x_i))$.



Correlation clustering

Correlation clustering.

Input: are x and y similar or dissimilar?

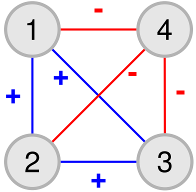
$$\begin{pmatrix} \cdot & + & + & - \\ + & \cdot & + & - \\ + & + & \cdot & - \\ - & - & - & \cdot \end{pmatrix}$$

Correlation clustering

Correlation clustering.

Input: are x and y similar or dissimilar?

$$\begin{pmatrix} \cdot & + & + & - \\ + & \cdot & + & - \\ + & + & \cdot & - \\ - & - & - & \cdot \end{pmatrix}$$

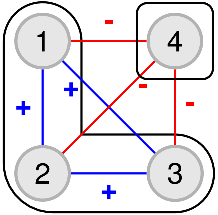


Correlation clustering

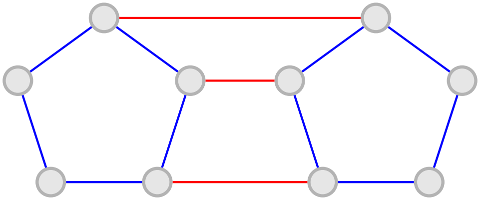
Correlation clustering.

Input: are x and y similar or dissimilar?

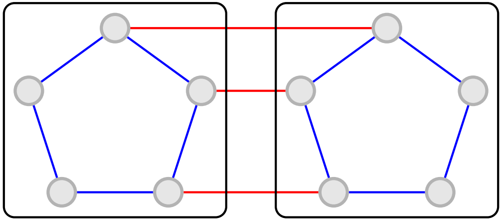
$$\begin{pmatrix} \cdot & + & + & - \\ + & \cdot & + & - \\ + & + & \cdot & - \\ - & - & - & \cdot \end{pmatrix}$$



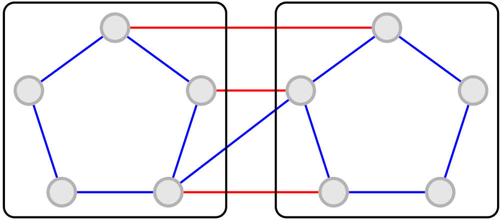
Correlation clustering



Correlation clustering

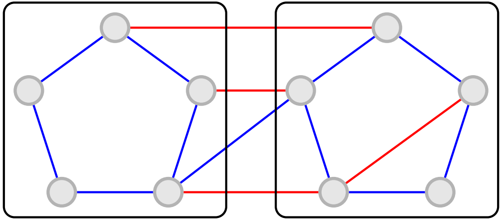


Correlation clustering



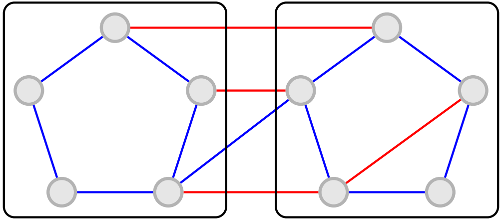
13 correct, 1 mistake.

Correlation clustering



13 correct, 2 mistakes.

Correlation clustering



13 correct, 2 mistakes.

The goal of correlation clustering is to partition a signed graph so as to

- minimize the number of mistakes (MINDISAGREE),
- or maximize the number of correct edges (MAXAGREE).

Correlation clustering

Correlation clustering variants:

- Is the input graph complete?
- Is the graph weighted?
- Maximize agreements or minimize disagreements?
- Is the number of clusters fixed?

All these variants are different in terms of hardness of approximation.

Correlation clustering

Correlation clustering does not require the number of clusters as input.

The optimal value could be any number between 1 and n.

Consider MINDISAGREE (complete graph, minimize mistakes).

When is it 1?

When is it n?

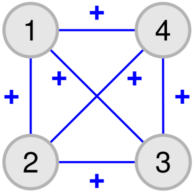
Correlation clustering

Correlation clustering does not require the number of clusters as input.

The optimal value could be any number between 1 and n.

Consider MINDISAGREE (complete graph, minimize mistakes).

When is it 1?



When is it n?

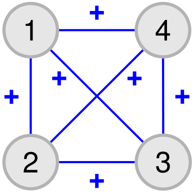
Correlation clustering

Correlation clustering does not require the number of clusters as input.

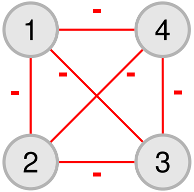
The optimal value could be any number between 1 and n.

Consider MINDISAGREE (complete graph, minimize mistakes).

When is it 1?



When is it n?



Correlation clustering

We know when a graph has a perfect 2-correlation-clustering (partition into 2 sets with no mistakes).

Correlation clustering

We know when a graph has a perfect 2-correlation-clustering (partition into 2 sets with no mistakes).

When does a graph have a perfect k-correlation-clustering, for any k?

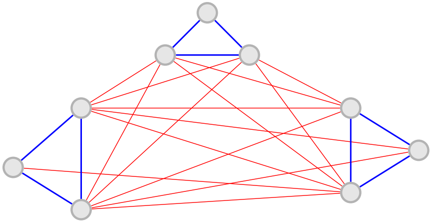
Correlation clustering

We know when a graph has a perfect 2-correlation-clustering (partition into 2 sets with no mistakes).

When does a graph have a perfect k-correlation-clustering, for any k?

Theorem

A signed graph G has a k-correlation-clustering with no mistakes if and only if G contains no cycle with exactly 1 negative edge.



Correlation clustering: algorithm analysis

We are going to analyze a few algorithms for correlation clustering:

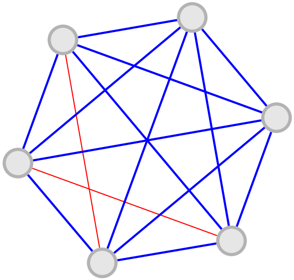
- A 2-approximation for MaxAgree.
- A 3-approximation for 2-MinDisagree.
- A PTAS for MaxAgree (incomplete analysis).

Correlation clustering

A 2-approximation algorithm for MaxAgree

Given a complete signed graph G, we seek a clustering maximizing agreements.

Algorithm:

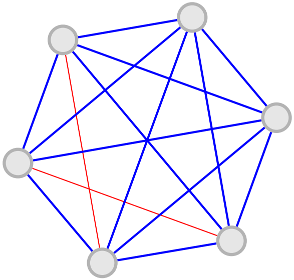


Correlation clustering

A 2-approximation algorithm for MAXAGREE

Given a complete signed graph G, we seek a clustering maximizing agreements.

Algorithm:



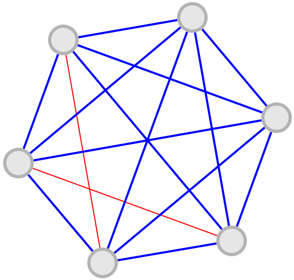
Upper-bounding OPT:

Correlation clustering

A 2-approximation algorithm for MAXAGREE

Given a complete signed graph G, we seek a clustering maximizing agreements.

Algorithm:

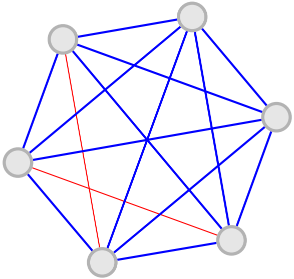


Upper-bounding OPT: $\binom{n}{2} \geq OPT$.

Given a complete signed graph G, we seek a clustering maximizing agreements.

Algorithm:

- If G has more **+** edges than **-** edges, put all vertices in the same cluster.
- Otherwise, put each vertex in a singleton cluster.



Upper-bounding OPT: $\binom{n}{2} \geq OPT$.

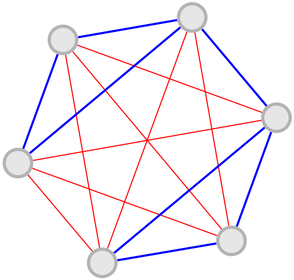
Correlation clustering

A 2-approximation algorithm for MaxAgree

Given a complete signed graph G, we seek a clustering maximizing agreements.

Algorithm:

- If G has more **+** edges than **-** edges, put all vertices in the same cluster.
- Otherwise, put each vertex in a singleton cluster.



Upper-bounding OPT: $\binom{n}{2} \geq OPT$.

We achieve a $1/2$-approximation.

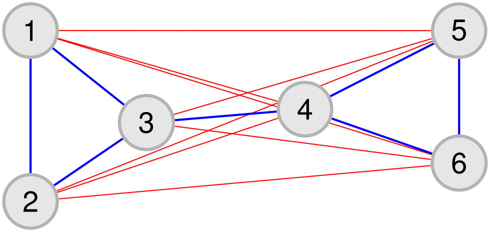
Correlation clustering

Algorithms for 2-MINDISAGREE

Given a complete signed graph $G = (V, E^+, E^-)$, we seek two clusters, C_1, C_2.

Algorithm: consider all clusterings $C_1 = N^+(v), C_2 = N^-(v)$ for all $v \in V$, where

- $N^+(v) = \{v\} \cup \{u \in V : (v, u) \in E^+\}$,
- $N^-(v) = \{u \in V : (v, u) \in E^-\}$.

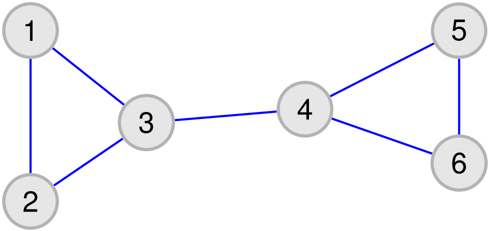


Note: in complete graphs the unsigned problem is equivalent, with missing edges playing the part of negative edges.

Given a complete signed graph $G = (V, E^+, E^-)$, we seek two clusters, C_1, C_2.

Algorithm: consider all clusterings $C_1 = N^+(v), C_2 = N^-(v)$ for all $v \in V$, where

- $N^+(v) = \{v\} \cup \{u \in V : (v, u) \in E^+\}$,
- $N^-(v) = \{u \in V : (v, u) \in E^-\}$.



Note: in complete graphs the unsigned problem is equivalent, with missing edges playing the part of negative edges.

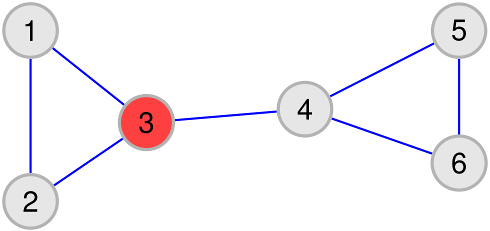
Correlation clustering

Algorithms for 2-MINDISAGREE

Given a complete signed graph $G = (V, E^+, E^-)$, we seek two clusters, C_1, C_2.

Algorithm: consider all clusterings $C_1 = N^+(v), C_2 = N^-(v)$ for all $v \in V$, where

- $N^+(v) = \{v\} \cup \{u \in V : (v, u) \in E^+\}$,
- $N^-(v) = \{u \in V : (v, u) \in E^-\}$.



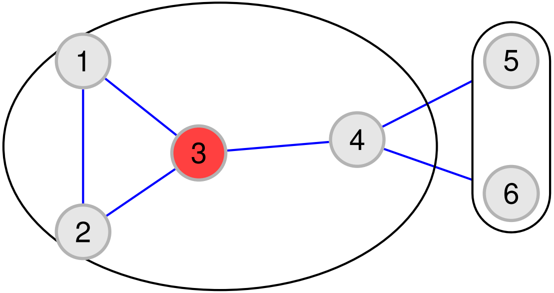
Correlation clustering

Algorithms for 2-MinDisagree

Given a complete signed graph $G = (V, E^+, E^-)$, we seek two clusters, C_1, C_2.

Algorithm: consider all clusterings $C_1 = N^+(v), C_2 = N^-(v)$ for all $v \in V$, where

- $N^+(v) = \{v\} \cup \{u \in V : (v, u) \in E^+\}$,
- $N^-(v) = \{u \in V : (v, u) \in E^-\}$.



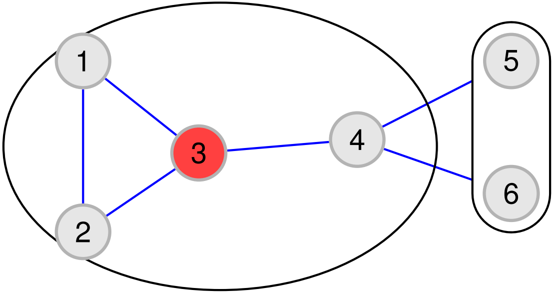
Correlation clustering

Algorithms for 2-MinDisagree

Given a complete signed graph $G = (V, E^+, E^-)$, we seek two clusters, C_1, C_2.

Algorithm: consider all clusterings $C_1 = N^+(v), C_2 = N^-(v)$ for all $v \in V$, where

- $N^+(v) = \{v\} \cup \{u \in V : (v, u) \in E^+\}$,
- $N^-(v) = \{u \in V : (v, u) \in E^-\}$.



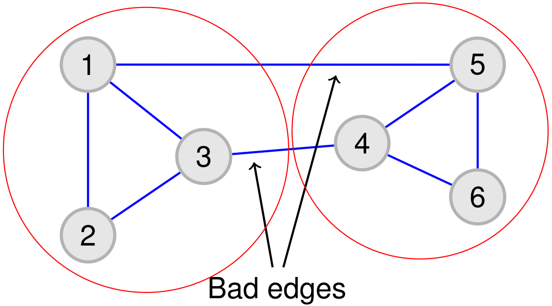
Claim: this algorithm makes at most $3OPT$ mistakes.

Correlation clustering

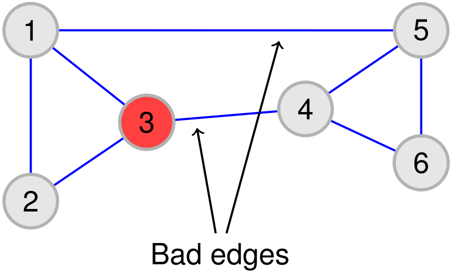
Algorithms for 2-MINDISAGREE

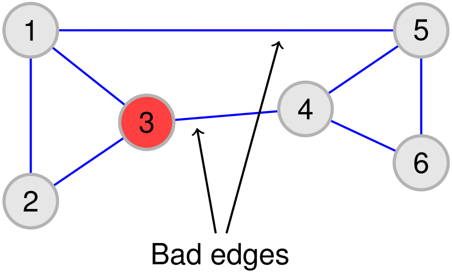
$ALG \leq 3OPT$. Analysis:

Optimal solution.



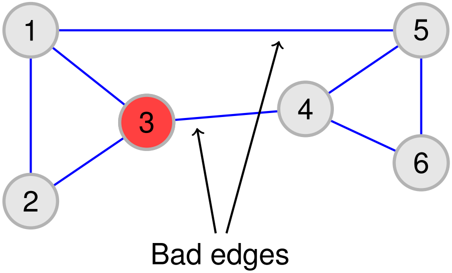
$ALG \leq 3OPT$. Analysis:





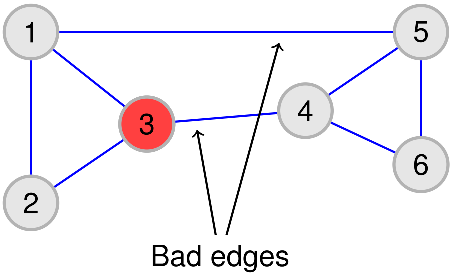
$ALG \leq 3OPT$. Analysis:

- We make some of the mistakes of OPT (pessimistically, *all* of them).



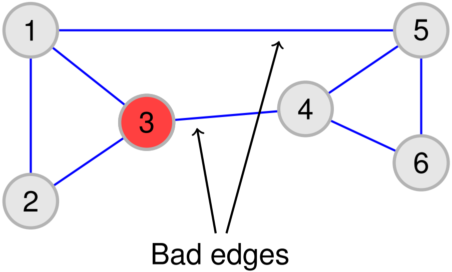
$ALG \leq 3OPT$. Analysis:

- We make some of the mistakes of OPT (pessimistically, *all* of them).
- Let d be the "bad" degree of v.
  - $v = 3, d = 1$.



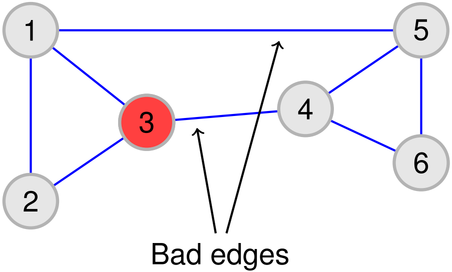
$ALG \leq 3OPT$. Analysis:

- We make some of the mistakes of *OPT* (pessimistically, *all* of them).
- Let d be the "bad" degree of v.
  - $v = 3, d = 1$.
- Each of the d "bad" neighbors induces less than n mistakes: nd mistakes at most (pessimistic).



$ALG \leq 3OPT$. Analysis:

- We make some of the mistakes of *OPT* (pessimistically, *all* of them).
- Let d be the “bad” degree of v.
  - $v = 3, d = 1$.
- Each of the d “bad” neighbors induces less than n mistakes: nd mistakes at most (pessimistic).
- Suppose d is minimal over all v. Then $OPT \geq nd/2$.

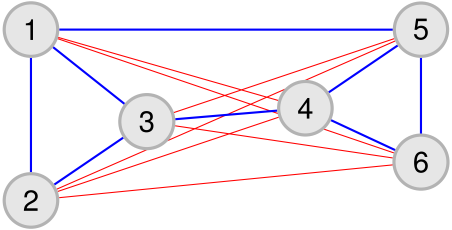


$ALG \leq 3OPT$. Analysis:

- We make some of the mistakes of OPT (pessimistically, *all* of them).
- Let d be the "bad" degree of v.
  - $v = 3, d = 1$.
- Each of the d "bad" neighbors induces less than n mistakes: nd mistakes at most (pessimistic).
- Suppose d is minimal over all v. Then $OPT \geq nd/2$.
- So we make at most $OPT + nd \leq OPT + 2OPT \leq 3OPT$ mistakes!

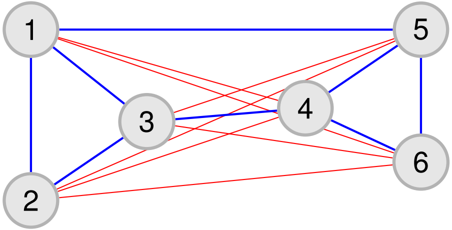
Correlation clustering

PTAS for MaxAgree



Correlation clustering

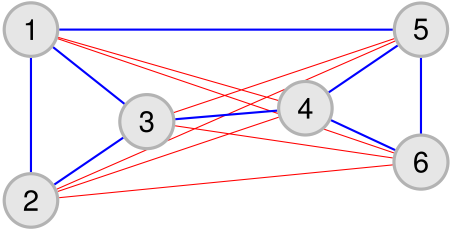
PTAS for MAXAGREE



- Remember: $OPT \geq \frac{1}{2}\binom{n}{2}$.
  - More **+** or **-** edges?

Correlation clustering

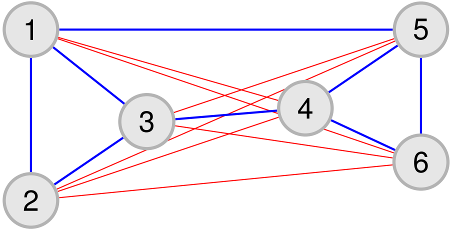
PTAS for MAXAGREE



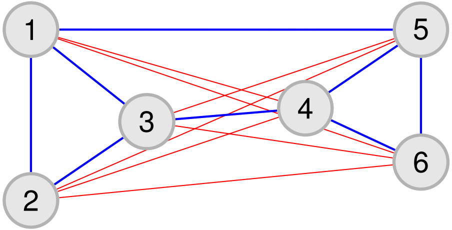
- Remember: $OPT \geq \frac{1}{2}\binom{n}{2}$.
  - More **+** or **-** edges?
- $\frac{1}{2}\binom{n}{2} = n(n-1)/4 = \frac{n^2}{4} - \frac{n}{4} = \Omega(n^2)$.

Correlation clustering

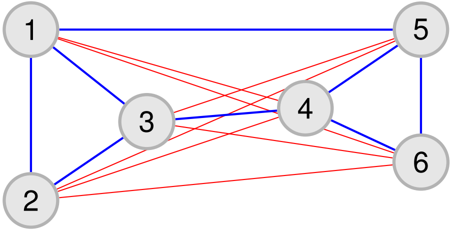
PTAS for MAXAGREE



- Remember: $OPT \geq \frac{1}{2}\binom{n}{2}$.
  - More **+** or **-** edges?
- $\frac{1}{2}\binom{n}{2} = n(n-1)/4 = \frac{n^2}{4} - \frac{n}{4} = \Omega(n^2)$.
- So it is enough to find a clustering $OPT - \epsilon n^2$ correct edges.



- Remember: $OPT \geq \frac{1}{2}\binom{n}{2}$.
  - More **+** or **-** edges?
- $\frac{1}{2}\binom{n}{2} = n(n-1)/4 = \frac{n^2}{4} - \frac{n}{4} = \Omega(n^2)$.
- So it is enough to find a clustering $OPT - \epsilon n^2$ correct edges.
- Rest of the analysis: reduction to General Partitioning and use as black box.



- Remember: $OPT \geq \frac{1}{2}\binom{n}{2}$.
  - More **+** or **-** edges?
- $\frac{1}{2}\binom{n}{2} = n(n-1)/4 = \frac{n^2}{4} - \frac{n}{4} = \Omega(n^2)$.
- So it is enough to find a clustering $OPT - \epsilon n^2$ correct edges.
- Rest of the analysis: reduction to General Partitioning and use as black box.
- Total running time: $e^{\mathcal{O}\left(\left(1/\epsilon\right)^{1/\epsilon}\right)} poly(n)$.

Consider a correlation clustering instance $G = (V, E^-, E^+)$, and a clustering $V = C_1 \cup C_2$.

Let A be the adjacency matrix of G.

Let x be the partition indicator vector, i.e.

$$x_i = \begin{cases} 1 & \text{if } v_i \in C_1 \\ -1 & \text{if } v_i \in C_2. \end{cases}$$

Then $x^T Ax = \text{agreements} - \text{disagreements}$.

Take-aways from this lecture:

- Basics of computational complexity.
- Basics of approximation algorithms.
- Correlation clustering.
  - Differences with respect to conventional clustering (e.g. k-means).
  - Perfect k-way partitioning.
- Analyses of some approximation algorithms.