Perturbation-Stable Maximum Cut

Yuchong Pan

UBC Beyond Worst-Case Analysis Reading Group
(Based on Tim Roughgarden's Notes for Stanford CS264)

June 30, 2020

Maximum Cut

Problem (Maximum Cut)

Input: An undirected graph $G = (V, E)$ with edge weights $w_e > 0$ for each $e \in E$.

Goal: A cut (A, B) that maximizes the weight of the *crossing* edges.

Maximum Cut

Problem (Maximum Cut)

Input: An undirected graph $G = (V, E)$ with edge weights $w_e > 0$ for each $e \in E$.

Goal: A cut (A, B) that maximizes the weight of the *crossing* edges.

- Maximum Cut is a type of 2-clustering problem (e.g. weights measure dissimilarities).

Maximum Cut Is *NP*-Hard

Problem (Maximum Cut, Decision Version)

Input: An undirected graph $G = (V, E)$ with edge weights $w_e > 0$ for each $e \in E$, and a positive integer W.

Output: Yes iff. there is a set $S \subseteq V$ such that the weight of the *crossing* edges is at least W.

MAXIMUM CUT Is *NP*-Hard

Problem (MAXIMUM CUT, Decision Version)

Input: An undirected graph $G = (V, E)$ with edge weights $w_e > 0$ for each $e \in E$, and a positive integer W.

Output: Yes iff. there is a set $S \subseteq V$ such that the weight of the *crossing* edges is at least W.

Problem (PARTITION, Decision Version)

Input: $(c_1, \ldots, c_n) \in \mathbb{Z}^n$.

Output: Yes iff. there is $I \subseteq [n]$ such that $\sum_{i \in I} c_i = \sum_{i \notin I} c_i$.

Maximum Cut Is *NP*-Hard

Problem (Maximum Cut, Decision Version)

Input: An undirected graph $G = (V, E)$ with edge weights $w_e > 0$ for each $e \in E$, and a positive integer W.

Output: Yes iff. there is a set $S \subseteq V$ such that the weight of the *crossing* edges is at least W.

Problem (Partition, Decision Version)

Input: $(c_1, \ldots, c_n) \in \mathbb{Z}^n$.

Output: Yes iff. there is $I \subseteq [n]$ such that $\sum_{i \in I} c_i = \sum_{i \notin I} c_i$.

Proof Sketch (Partition $\leq_P$ Maximum Cut)

- $G = K_n$.
- $w_{ij} = c_i c_j$ for all $i, j \in V, i \neq j$.
- $W = \lceil \frac{1}{4} \sum c_i^2 \rceil$.

Maximum Cut Is *NP*-Hard

- Minimum Cut is *not* *NP*-hard and can be solved by the Maximum-Flow Minimum-Cut Theorem.

MAXIMUM CUT Is *NP*-Hard

- MINIMUM CUT is *not* *NP*-hard and can be solved by the Maximum-Flow Minimum-Cut Theorem.
- **Question:** Can't we negate the edge weights, yielding a MINIMUM CUT instance?

MAXIMUM CUT Is *NP*-Hard

- MINIMUM CUT is *not* *NP*-hard and can be solved by the Maximum-Flow Minimum-Cut Theorem.
- **Question:** Can't we negate the edge weights, yielding a MINIMUM CUT instance?
- No! Polynomial-time algorithms solving MINIMUM CUT require nonnegative edge weights.

Exact Recovery

- **Theme:** To recover the optimal solution in polynomial time in γ-*perturbation-stable* instances, where γ is as small as possible.

Exact Recovery

- **Theme:** To recover the optimal solution in polynomial time in γ-*perturbation-stable* instances, where γ is as small as possible.

Definition (γ-Perturbation-Stability)

For $\gamma \geq 1$, an instance of MAXIMUM CUT is γ-*perturbation-stable* if a cut (A, B) is the *unique* optimal solution to all γ-*perturbations*, where each original edge weight w_e is replaced with an edge weight $w'_e \in [\frac{1}{\gamma} w_e, w_e]$.

LP Relaxation, Take 1

- **Question:** Can we use an LP relaxation similar to the one for MINIMUM CUT, i.e.

$$
\begin{aligned}
\max \quad & \sum_{e \in E} w_e x_e \\
\text{s.t.} \quad & x_e \geq |d_u - d_v|, && \forall e = uv \in E. \\
& x_e \in [0,1], && \forall e \in E. \\
& d_v \in [0,1], && \forall v \in V.
\end{aligned}
$$

LP Relaxation, Take 1

- **Question:** Can we use an LP relaxation similar to the one for MINIMUM CUT, i.e.

$$\begin{aligned} \max \quad & \sum_{e \in E} w_e x_e & \\ \text{s.t.} \quad & x_e \geq |d_u - d_v|, & \forall e = uv \in E. \\ & x_e \in [0,1], & \forall e \in E. \\ & d_v \in [0,1], & \forall v \in V. \end{aligned}$$

- No! $x_e = 1$ for each $e \in E$ is a feasible solution and maximizes the objective value.

LP Relaxation, Take 1

- **Question:** Can we use an LP relaxation similar to the one for MINIMUM CUT, i.e.

$$
\begin{aligned}
\text{max} \quad & \sum_{e \in E} w_e x_e \\
\text{s.t.} \quad & x_e \geq |d_u - d_v|, && \forall e = uv \in E. \\
& x_e \in [0, 1], && \forall e \in E. \\
& d_v \in [0, 1], && \forall v \in V.
\end{aligned}
$$

- No! $x_e = 1$ for each $e \in E$ is a feasible solution and maximizes the objective value.
- **Question:** What about $x_e \leq d_u - d_v$ and $x_e \leq d_v - d_u$?

LP Relaxation, Take 1

- **Question:** Can we use an LP relaxation similar to the one for Minimum Cut, i.e.

$$\begin{aligned} \max \quad & \sum_{e \in E} w_e x_e \\ \text{s.t.} \quad & x_e \geq |d_u - d_v|, && \forall e = uv \in E. \\ & x_e \in [0,1], && \forall e \in E. \\ & d_v \in [0,1], && \forall v \in V. \end{aligned}$$

- No! $x_e = 1$ for each $e \in E$ is a feasible solution and maximizes the objective value.
- **Question:** What about $x_e \leq d_u - d_v$ and $x_e \leq d_v - d_u$?
- This forces $x_e = 0$, instead of $x_e \leq |d_u - d_v|$.

LP Relaxation, Take 2

- Let $x_{ij} \in \{0, 1\}$ denote whether or not i, j are on different sides of the cut, for all distinct $i, j \in V$. We denote by x_{ij} and x_{ji} the same variable.

LP Relaxation, Take 2

- Let $x_{ij} \in \{0, 1\}$ denote whether or not i, j are on different sides of the cut, for all distinct $i, j \in V$. We denote by x_{ij} and x_{ji} the same variable.
- **Intuition:** If i, j are on different sides, and i, k are also on different sides, then j, k must be on the same sides.

LP Relaxation, Take 2

- Let $x_{ij} \in \{0, 1\}$ denote whether or not i, j are on different sides of the cut, for all distinct $i, j \in V$. We denote by x_{ij} and x_{ji} the same variable.
- **Intuition:** If i, j are on different sides, and i, k are also on different sides, then j, k must be on the same sides.
- For any distinct $i, j, k \in V$, at most two of x_{ij}, x_{ik}, x_{jk} are 1.

LP Relaxation, Take 2

- Let $x_{ij} \in \{0,1\}$ denote whether or not i, j are on different sides of the cut, for all distinct $i, j \in V$. We denote by x_{ij} and x_{ji} the same variable.
- **Intuition:** If i, j are on different sides, and i, k are also on different sides, then j, k must be on the same sides.
- For any distinct $i, j, k \in V$, at most two of x_{ij}, x_{ik}, x_{jk} are 1.

$$x_{ij} + x_{ik} + x_{jk} \leq 2, \qquad \forall i, j, k \in V \text{ distinct.}$$

LP Relaxation, Take 2

- Let $x_{ij} \in \{0, 1\}$ denote whether or not i, j are on different sides of the cut, for all distinct $i, j \in V$. We denote by x_{ij} and x_{ji} the same variable.
- **Intuition:** If i, j are on different sides, and i, k are also on different sides, then j, k must be on the same sides.
- For any distinct $i, j, k \in V$, at most two of x_{ij}, x_{ik}, x_{jk} are 1.

$$x_{ij} + x_{ik} + x_{jk} \leq 2, \qquad \forall i, j, k \in V \text{ distinct.}$$

- **Intuition:** If i, j are on the same side, and i, k are on the same side, then j, k are on the same side.

LP Relaxation, Take 2

- Let $x_{ij} \in \{0, 1\}$ denote whether or not i, j are on different sides of the cut, for all distinct $i, j \in V$. We denote by x_{ij} and x_{ji} the same variable.
- **Intuition:** If i, j are on different sides, and i, k are also on different sides, then j, k must be on the same sides.
- For any distinct $i, j, k \in V$, at most two of x_{ij}, x_{ik}, x_{jk} are 1.

$$x_{ij} + x_{ik} + x_{jk} \leq 2, \qquad \forall i, j, k \in V \text{ distinct.}$$

- **Intuition:** If i, j are on the same side, and i, k are on the same side, then j, k are on the same side.
- For any distinct $i, j, k \in V$, $x_{ij} = x_{ik} = 0$ implies $x_{jk} = 0$.

LP Relaxation, Take 2

- Let $x_{ij} \in \{0,1\}$ denote whether or not i,j are on different sides of the cut, for all distinct $i,j \in V$. We denote by x_{ij} and x_{ji} the same variable.
- **Intuition:** If i,j are on different sides, and i,k are also on different sides, then j,k must be on the same sides.
- For any distinct $i,j,k \in V$, at most two of x_{ij}, x_{ik}, x_{jk} are 1.

$$x_{ij} + x_{ik} + x_{jk} \leq 2, \qquad \forall i,j,k \in V \text{ distinct.}$$

- **Intuition:** If i,j are on the same side, and i,k are on the same side, then j,k are on the same side.
- For any distinct $i,j,k \in V$, $x_{ij} = x_{ik} = 0$ implies $x_{jk} = 0$.

$$x_{jk} \leq x_{ij} + x_{ik}, \qquad \forall i,j,k \in V \text{ distinct.}$$

LP Relaxation, Take 2

- Hence we obtain the LP relaxation $(\text{LP-}\textsc{MaxCut})$:

$$\begin{aligned}
\max \quad & \sum_{(i,j)\in E} w_{ij}x_{ij} & \\
\text{s.t.} \quad & x_e \geq |d_u - d_v|, & \forall e = uv \in E. \\
& x_{ij} + x_{ik} + x_{jk} \leq 2, & \forall i,j,k \in V \text{ distinct.} \\
& x_{jk} \leq x_{ij} + x_{ik}, & \forall i,j,k \in V \text{ distinct.} \\
& x_{ij} \in [0,1], & \forall i,j \in V \text{ distinct.}
\end{aligned}$$

Main Theorem

Theorem

There is a constant $c > 0$ such that in every $(c \log n)$-perturbation-stable instance of $\textsc{Maximum Cut}$ *with n vertices,* $(\text{LP-}\textsc{MaxCut})$ *solves to integers.*

Main Theorem

Theorem

There is a constant $c > 0$ such that in every $(c \log n)$-perturbation-stable instance of Maximum Cut *with n vertices,* (LP-MaxCut) *solves to integers.*

- Recall the proofs of exact recovery by LP in 1-perturbation-stable Minimum s-t Cut instances and in 4-perturbation-stable Minimum Multiway Cut instances.

Main Theorem

Theorem

There is a constant $c > 0$ such that in every $(c \log n)$-perturbation-stable instance of MAXIMUM CUT *with n vertices,* (LP-MAXCUT) *solves to integers.*

- Recall the proofs of exact recovery by LP in 1-perturbation-stable MINIMUM s-t CUT instances and in 4-perturbation-stable MINIMUM MULTIWAY CUT instances.
- In each of the two proofs we design a **randomized rounding algorithm** that outputs a (random) cut such that the probablity of an edge being cut is approximately the same as the value of the corresponding decision variable.

Main Theorem

Theorem

There is a constant $c > 0$ such that in every $(c \log n)$-perturbation-stable instance of $\textsc{Maximum Cut}$ *with n vertices,* $(\text{LP-}\textsc{MaxCut})$ *solves to integers.*

- Recall the proofs of exact recovery by LP in 1-perturbation-stable $\textsc{Minimum}$ s-t $\textsc{Cut}$ instances and in 4-perturbation-stable $\textsc{Minimum Multiway Cut}$ instances.
- In each of the two proofs we design a **randomized rounding algorithm** that outputs a (random) cut such that the probablity of an edge being cut is approximately the same as the value of the corresponding decision variable.
- $\textsc{Minimum}$ s-t $\textsc{Cut}$: $A = \{v \in V : \hat{d}_v \le r\}$ and $B = V \setminus A$, where $r \sim \text{Uniform}(0, 1)$.

Main Theorem

Theorem

There is a constant $c > 0$ such that in every $(c \log n)$-perturbation-stable instance of Maximum Cut *with n vertices,* (LP-MaxCut) *solves to integers.*

- Recall the proofs of exact recovery by LP in 1-perturbation-stable Minimum s-t Cut instances and in 4-perturbation-stable Minimum Multiway Cut instances.
- In each of the two proofs we design a **randomized rounding algorithm** that outputs a (random) cut such that the probablity of an edge being cut is approximately the same as the value of the corresponding decision variable.
- Minimum s-t Cut: $A = \{v \in V : \hat{d}_v \leq r\}$ and $B = V \setminus A$, where $r \sim \text{Uniform}(0, 1)$.
- Minimum Multiway Cut: For each iteration, a group and a threshold are chosen uniformly randomly.

Main Theorem

Fact

LP algorithms (e.g. the ellipsoid method) always return an extreme point of the feasible region.

Main Theorem

Fact

LP algorithms (e.g. the ellipsoid method) always return an extreme point of the feasible region.

- Proof omitted here. For the ellipsoid method see e.g. CPSC 536S Submodular Optimization.

Main Theorem

Fact

LP algorithms (e.g. the ellipsoid method) always return an extreme point of the feasible region.

- Proof omitted here. For the ellipsoid method see e.g. CPSC 536S Submodular Optimization.

Exercise 2. *Show how to find (in polytime) a bfs with objective value within the range. You may use the LP oracle.*

Main Theorem

Fact

LP algorithms (e.g. the ellipsoid method) always return an extreme point of the feasible region.

- Proof omitted here. For the ellipsoid method see e.g. CPSC 536S Submodular Optimization.

  Exercise 2. *Show how to find (in polytime) a bfs with objective value within the range. You may use the LP oracle.*

- Since all of the extreme points of the feasible region are integral and correspond to a cut, then LP algorithms always solve $(\text{LP-}\textsc{MaxCut})$ to an integral optimal solution.

Main Theorem

- A randomized rounding algorithm implies the exact recovery theorem since:

Main Theorem

- A randomized rounding algorithm implies the exact recovery theorem since:
  1. The optimal fractional solution $\hat{x}$ can only be better than the optimal integral solution C^*;

Main Theorem

- A randomized rounding algorithm implies the exact recovery theorem since:
  1. The optimal fractional solution $\hat{x}$ can only be better than the optimal integral solution C^*;
  2. The randomized rounding algorithm gives a distribution over s-t cuts that is as good, on average, as C^*;

Main Theorem

- A randomized rounding algorithm implies the exact recovery theorem since:
  1. The optimal fractional solution $\hat{x}$ can only be better than the optimal integral solution C^*;
  2. The randomized rounding algorithm gives a distribution over s-t cuts that is as good, on average, as C^*;
  3. Hence the distribution must be a point mass on C^*.

Main Theorem

- A randomized rounding algorithm implies the exact recovery theorem since:
    1. The optimal fractional solution $\hat{x}$ can only be better than the optimal integral solution C^*;
    2. The randomized rounding algorithm gives a distribution over s-t cuts that is as good, on average, as C^*;
    3. Hence the distribution must be a point mass on C^*.
- Formally, we define $\Delta(C)$ to be the total cost of C that exceeds that of C^* and $\Delta(\hat{x})$ to be total cost of C^* that exceeds the objective function value of $\hat{x}$.

Main Theorem

- A randomized rounding algorithm implies the exact recovery theorem since:
  1. The optimal fractional solution $\hat{x}$ can only be better than the optimal integral solution C^*;
  2. The randomized rounding algorithm gives a distribution over s-t cuts that is as good, on average, as C^*;
  3. Hence the distribution must be a point mass on C^*.
- Formally, we define $\Delta(C)$ to be the total cost of C that exceeds that of C^* and $\Delta(\hat{x})$ to be total cost of C^* that exceeds the objective function value of $\hat{x}$.
- We show that $\mathbb{E}[\Delta(C)] \leq 0$ by the probablity properties of the cut generated by the randomized rounding algorithm.

Main Theorem

- A randomized rounding algorithm implies the exact recovery theorem since:
  1. The optimal fractional solution $\hat{x}$ can only be better than the optimal integral solution C^*;
  2. The randomized rounding algorithm gives a distribution over s-t cuts that is as good, on average, as C^*;
  3. Hence the distribution must be a point mass on C^*.
- Formally, we define $\Delta(C)$ to be the total cost of C that exceeds that of C^* and $\Delta(\hat{x})$ to be total cost of C^* that exceeds the objective function value of $\hat{x}$.
- We show that $\mathbb{E}[\Delta(C)] \leq 0$ by the probablity properties of the cut generated by the randomized rounding algorithm.
- Since $\Delta(C) \geq 0$ and since the equality holds iff. C is an optimal cut, it follows that the randomized rounding algorithm outputs an optimal cut w.p.1.

Randomized Rounding Algorithm

Lemma

Fix an instance of the $\textsc{Maximum Cut}$ *problem, with* F^* *the edges in the optimal cut, and* $\hat{x}$ *the optimal solution to* $(\text{LP-}\textsc{MaxCut})$*. Then there exists a randomized algorithm that generates a random cut* (A, B) *and a scaling parameter* $\sigma > 0$ *such that:*

1. *For every edge* $e = ij \notin F^*$,
$$\mathbb{P}[e \text{ cut by } (A, B)] \geq \sigma \cdot \frac{\hat{x}_{ij}}{\alpha},$$
where $\alpha = \Theta(\log n)$*;*
2. *For every edge* $e = ij \in F^*$,
$$\mathbb{P}[e \text{ not cut by } (A, B)] \leq \sigma \cdot (1 - \hat{x}_{ij});$$
3. *The rounding algorithm is deterministic iff.* $\hat{x}$ *is integral.*

Randomized Rounding Algorithm, Roadmap

- **Exercise:** Show that this lemma implies the main theorem (outlined above, Homework #4).

Randomized Rounding Algorithm, Roadmap

- **Exercise:** Show that this lemma implies the main theorem (outlined above, Homework #4).

Proposition

Fix an instance of $\textsc{Maximum Cut}$, a cut C, and a feasible solution $\hat{x}$ to $(\text{LP-}\textsc{MaxCut})$. For distinct $i, j \in V$, define

$$\hat{y}_{ij} = \begin{cases} \hat{x}_{ij}, & \text{if } i, j \text{ are on the same side of } C, \\ 1 - \hat{x}_{ij}, & \text{if } i, j \text{ are on different sides of } C. \end{cases}$$

Then $\hat{y}$ satisfies the triangle inequality:

$$\hat{y}_{jk} \leq \hat{y}_{ij} + \hat{y}_{ik}$$

for every $i, j, k \in V$.

Randomized Rounding Algorithm, Roadmap

- **Exercise:** Show that this lemma implies the main theorem (outlined above, Homework #4).

Proposition

Fix an instance of MAXIMUM CUT, a cut C, and a feasible solution $\hat{x}$ to (LP-MAXCUT). For distinct $i, j \in V$, define

$$\hat{y}_{ij} = \begin{cases} \hat{x}_{ij}, & \text{if } i, j \text{ are on the same side of } C, \\ 1 - \hat{x}_{ij}, & \text{if } i, j \text{ are on different sides of } C. \end{cases}$$

Then $\hat{y}$ satisfies the triangle inequality:

$$\hat{y}_{jk} \leq \hat{y}_{ij} + \hat{y}_{ik}$$

for every $i, j, k \in V$.

- That is, $\hat{x}, \hat{y}$ are both *semi-metrics* (metrics except that distinct points may have zero distances).

Randomized Rounding Algorithm, Roadmap

Theorem (Bourgain's Theorem)

For every n-point semi-metric space (X, d), there exists a randomized algorithm that generates a random partition (A, B) of X and a scaling parameter $\sigma > 0$ such that, for all distinct $i, j \in X$,

$$\mathbb{P}[i, j \textit{ on different sides of } (A, B)] \in \sigma \cdot \left[\frac{d(i, j)}{\alpha}, d(i, j) \right],$$

where $\alpha = \Theta(\log n)$.

Randomized Rounding Algorithm, Roadmap

Theorem (Bourgain's Theorem)

For every n-point semi-metric space (X, d), there exists a randomized algorithm that generates a random partition (A, B) of X and a scaling parameter $\sigma > 0$ such that, for all distinct $i, j \in X$,

$$\mathbb{P}[i, j \textit{ on different sides of } (A, B)] \in \sigma \cdot \left[\frac{d(i, j)}{\alpha}, d(i, j)\right],$$

where $\alpha = \Theta(\log n)$.

- That is, every n-point metric space admits a randomized partitioning algorithm so that the sepration probabilities between pairs of points are *proportional* to the distances, up to a $\Theta(\log n)$ factor.

Randomized Rounding Algorithm, Roadmap

Theorem (Bourgain's Theorem)

For every n-point semi-metric space (X, d), there exists a randomized algorithm that generates a random partition (A, B) of X and a scaling parameter $\sigma > 0$ such that, for all distinct $i, j \in X$,

$$\mathbb{P}[i, j \textit{ on different sides of } (A, B)] \in \sigma \cdot \left[\frac{d(i,j)}{\alpha}, d(i,j)\right],$$

where $\alpha = \Theta(\log n)$.

- That is, every n-point metric space admits a randomized partitioning algorithm so that the sepration probabilities between pairs of points are *proportional* to the distances, up to a $\Theta(\log n)$ factor.
- The $\Theta(\log n)$ approximation factor is the best possible for *arbitrary* semi-metric spaces.

Randomized Rounding Algorithm, Roadmap

Proof (Proposition & Bourgain's Theorem $\implies$ Lemma).

- Fix an instance of MAXIMUM CUT. Let C^* denote an optimal cut, cutting the edges F^*.

Randomized Rounding Algorithm, Roadmap

Proof (Proposition & Bourgain's Theorem $\implies$ Lemma).

- Fix an instance of $\textsc{Maximum Cut}$. Let C^* denote an optimal cut, cutting the edges F^*.
- Let $\hat{x}$ be an optimal solution to $(\text{LP-}\textsc{MaxCut})$. Define $\hat{y}$ as in Proposition (with C^* being the cut).

Randomized Rounding Algorithm, Roadmap

Proof (Proposition & Bourgain's Theorem $\implies$ Lemma).

- Fix an instance of $\textsc{Maximum Cut}$. Let C^* denote an optimal cut, cutting the edges F^*.
- Let $\hat{\mathrm{x}}$ be an optimal solution to $(\text{LP-}\textsc{MaxCut})$. Define $\hat{\mathrm{y}}$ as in Proposition (with C^* being the cut).
- By Proposition, $\hat{\mathrm{y}}$ is a semi-metric.

Randomized Rounding Algorithm, Roadmap

Proof (Proposition & Bourgain's Theorem $\implies$ Lemma).

- Fix an instance of Maximum Cut. Let C^* denote an optimal cut, cutting the edges F^*.
- Let $\hat{x}$ be an optimal solution to (LP-MaxCut). Define $\hat{y}$ as in Proposition (with C^* being the cut).
- By Proposition, $\hat{y}$ is a semi-metric.
- By Bourgain's Theorem, there is a randomized algorithm that outputs a partition (A, B) and $\sigma > 0$ such that

$$\mathbb{P}[i, j \text{ on different sides of } (A, B)] = \sigma \cdot \left[\frac{\hat{y}_{ij}}{\alpha}, \hat{y}_{ij}\right],$$

where $\alpha = \Theta(\log n)$.

Randomized Rounding Algorithm, Roadmap

Proof (Proposition & Bourgain's Theorem $\implies$ Lemma).

- By the definition of $\hat{y}$,
  1. If i, j are on the same side of C^*, then
$$\mathbb{P}[i, j \text{ on different sides of } (A, B)] \in \sigma \cdot \left[\frac{\hat{x}_{ij}}{\alpha}, \hat{x}_{ij}\right].$$
  2. If i, j are on different sides of C^*, then
$$\mathbb{P}[i, j \text{ on different sides of } (A, B)] \in \sigma \cdot \left[\frac{1 - \hat{x}_{ij}}{\alpha}, 1 - \hat{x}_{ij}\right].$$

Randomized Rounding Algorithm, Roadmap

Proof (Proposition & Bourgain's Theorem $\implies$ Lemma).

- By the definition of $\hat{y}$,
  1. If i, j are on the same side of C^*, then
  $$\mathbb{P}[i, j \text{ on different sides of } (A, B)] \in \sigma \cdot \left[\frac{\hat{x}_{ij}}{\alpha}, \hat{x}_{ij}\right].$$
  2. If i, j are on different sides of C^*, then
  $$\mathbb{P}[i, j \text{ on different sides of } (A, B)] \in \sigma \cdot \left[\frac{1 - \hat{x}_{ij}}{\alpha}, 1 - \hat{x}_{ij}\right].$$
- Lemma follows.

$\square$

Randomized Rounding Algorithm, Roadmap

Proof (Proposition & Bourgain's Theorem $\implies$ Lemma).

- By the definition of $\hat{y}$,
  1. If i, j are on the same side of C^*, then
  $$\mathbb{P}[i, j \text{ on different sides of } (A, B)] \in \sigma \cdot \left[\frac{\hat{x}_{ij}}{\alpha}, \hat{x}_{ij}\right].$$
  2. If i, j are on different sides of C^*, then
  $$\mathbb{P}[i, j \text{ on different sides of } (A, B)] \in \sigma \cdot \left[\frac{1 - \hat{x}_{ij}}{\alpha}, 1 - \hat{x}_{ij}\right].$$
- Lemma follows.

$\square$

- **Exercise:** Prove Proposition and Bourgain's Theorem (Homework #4). For Bourgain's Theorem see e.g. CPSC 531F Tools for Modern Algorithm Analysis.

Metric Embedding