A Combinatorial, Primal-Dual Approach to Semidefinite Programs
(Paper Presentation)

Pankaj Prateek
Akshay Kumar

IIT Kanpur

Outline

Outline

Primal-Dual Schema for LP

Primal-Dual Schema for LP

Primal-Dual Schema for LP

Primal

$$\begin{aligned} \min \sum_{j=1}^{n} c_j x_j & \\ \text{s.t.} \sum_{j=1}^{n} a_{ij} x_j \geq b_i, & \quad i = 1, \ldots, m \\ x_j \geq 0 & \quad j = 1, \ldots, n \end{aligned}$$

Primal-Dual Schema for LP

Primal

$$\min \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t.} \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1, \ldots, m$$

$$x_j \geq 0 \quad j = 1, \ldots, n$$

Dual

$$\max \sum_{i=1}^{m} b_i y_i$$

$$\text{s.t.} \sum_{i=1}^{m} a_{ij} y_i \leq c_j, \quad j = 1, \ldots, n$$

$$y_i \geq 0 \quad j = 1, \ldots, m$$

Primal-Dual Schema for LP

Primal

$$\min \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t.} \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1, \ldots, m$$

$$x_j \geq 0 \quad j = 1, \ldots, n$$

Dual

$$\max \sum_{i=1}^{m} b_i y_i$$

$$\text{s.t.} \sum_{i=1}^{m} a_{ij} y_i \leq c_j, \quad j = 1, \ldots, n$$

$$y_i \geq 0 \quad j = 1, \ldots, m$$

Primal Complementary Slackness Conditions

Let $\alpha \geq 1$.

Then for each $1 \leq j \leq n$: either $x_j = 0$ or $c_j/\alpha \leq \Sigma_{i=1}^{m} a_{ij} y_i \leq c_j$.

Primal-Dual Schema for LP

Primal

$$\min \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t.} \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1, \ldots, m$$

$$x_j \geq 0 \quad j = 1, \ldots, n$$

Dual

$$\max \sum_{i=1}^{m} b_i y_i$$

$$\text{s.t.} \sum_{i=1}^{m} a_{ij} y_i \leq c_j, \quad j = 1, \ldots, n$$

$$y_i \geq 0 \quad j = 1, \ldots, m$$

Primal Complementary Slackness Conditions

Let $\alpha \geq 1$.
Then for each $1 \leq j \leq n$: either $x_j = 0$ or $c_j/\alpha \leq \Sigma_{i=1}^{m} a_{ij} y_i \leq c_j$.

Dual Complementary Slackness Conditions

Let $\beta \geq 1$.
Then for each $1 \leq i \leq m$: either $y_i = 0$ or $b_i \leq \Sigma_{j=1}^{n} a_{ij} x_j \leq \beta \cdot b_i$.

Primal-Dual Schema for LP

Theorem

If **x** and **y** are primal and dual feasible solutions satisfying the conditions stated above then

$$\sum_{j=1}^{n} c_j x_j \leq \alpha \cdot \beta \cdot \sum_{i=1}^{m} b_i y_i$$

Primal-Dual Schema for LP

Theorem

If **x** and **y** are primal and dual feasible solutions satisfying the conditions stated above then

$$\sum_{j=1}^{n} c_j x_j \leq \alpha \cdot \beta \cdot \sum_{i=1}^{m} b_i y_i$$

Proof

$$\sum_{j} c_j x_j \leq \alpha \cdot \sum_{i,j} a_{ij} x_j y_i \leq \alpha \cdot \beta \cdot \sum_{i} b_i y_i$$

The first inequality follows from the Primal Complementary Slackness Condition whereas the second follows from the Dual Complementary Slackness Condition.

Primal-Dual Schema for LP

Algorithm

Primal-Dual Schema for LP

Algorithm

- Start with a primal infeasible solution and dual feasible solution, typically $\mathbf{x} = \mathbf{0}$ and $\mathbf{y} = \mathbf{0}$.

Primal-Dual Schema for LP

Algorithm

- Start with a primal infeasible solution and dual feasible solution, typically $\mathbf{x} = \mathbf{0}$ and $\mathbf{y} = \mathbf{0}$.
- Iteratively improve the feasibility of the primal solution and optimality of the dual solution ensuring that a primal feasible solution is obtained in the end and all conditions are satisfied for a suitable choice of α and β.

Primal-Dual Schema for LP

Algorithm

- Start with a primal infeasible solution and dual feasible solution, typically $\mathbf{x} = \mathbf{0}$ and $\mathbf{y} = \mathbf{0}$.
- Iteratively improve the feasibility of the primal solution and optimality of the dual solution ensuring that a primal feasible solution is obtained in the end and all conditions are satisfied for a suitable choice of α and β.
- The primal solution is always extended integrally to ensure final solution is integral. Need not be true for dual solution.

Primal-Dual Schema for LP

Algorithm

- Start with a primal infeasible solution and dual feasible solution, typically $\mathbf{x} = \mathbf{0}$ and $\mathbf{y} = \mathbf{0}$.
- Iteratively improve the feasibility of the primal solution and optimality of the dual solution ensuring that a primal feasible solution is obtained in the end and all conditions are satisfied for a suitable choice of α and β.
- The primal solution is always extended integrally to ensure final solution is integral. Need not be true for dual solution.
- Cost of dual solution used as lower bound on OPT.

Primal-Dual Schema for LP

Algorithm

- Start with a primal infeasible solution and dual feasible solution, typically $\mathbf{x} = \mathbf{0}$ and $\mathbf{y} = \mathbf{0}$.
- Iteratively improve the feasibility of the primal solution and optimality of the dual solution ensuring that a primal feasible solution is obtained in the end and all conditions are satisfied for a suitable choice of α and β.
- The primal solution is always extended integrally to ensure final solution is integral. Need not be true for dual solution.
- Cost of dual solution used as lower bound on OPT.
- Approximation ratio of $\alpha\beta$ by the theorem on previous slide.

Outline

Primal-Dual Schema for SDP

Primal-Dual Schema for SDP

Primal-Dual Schema for SDP

Primal

$$\begin{aligned} \max\ & \mathbf{C} \bullet \mathbf{X} \\ \text{s.t. } & \mathbf{A_j} \bullet \mathbf{X} \leq b_j \qquad \forall j \in [m] \\ & \mathbf{X} \succeq \mathbf{0} \end{aligned}$$

Primal-Dual Schema for SDP

Dual

Primal

$$\begin{aligned} \max\ & \mathbf{C} \bullet \mathbf{X} \\ \text{s.t. } & \mathbf{A_j} \bullet \mathbf{X} \le b_j \quad \forall j \in [m] \\ & \mathbf{X} \succeq \mathbf{0} \end{aligned}$$

$$\begin{aligned} \min\ & \mathbf{b} \cdot \mathbf{y} \\ & \sum_{j=1}^{m} \mathbf{A_j} y_j \succeq \mathbf{C} \\ & \mathbf{y} \ge \mathbf{0} \end{aligned}$$

Primal-Dual Schema for SDP

Primal

$$\begin{aligned} &\max\ \mathbf{C} \bullet \mathbf{X} \\ &\text{s.t. } \mathbf{A_j} \bullet \mathbf{X} \leq b_j \qquad \forall j \in [m] \\ &\quad \mathbf{X} \succeq \mathbf{0} \end{aligned}$$

Dual

$$\begin{aligned} &\min\ \mathbf{b} \cdot \mathbf{y} \\ &\sum_{j=1}^{m} \mathbf{A_j} y_j \succeq \mathbf{C} \\ &\mathbf{y} \geq \mathbf{0} \end{aligned}$$

Primal-Dual Schema for SDP

Primal

$$\begin{aligned} \max\ & \mathbf{C} \bullet \mathbf{X} \\ \text{s.t. } & \mathbf{A_j} \bullet \mathbf{X} \leq b_j \qquad \forall j \in [m] \\ & \mathbf{X} \succeq \mathbf{0} \end{aligned}$$

Dual

$$\begin{aligned} \min\ & \mathbf{b} \cdot \mathbf{y} \\ & \sum_{j=1}^{m} \mathbf{A_j} y_j \succeq \mathbf{C} \\ & \mathbf{y} \geq \mathbf{0} \end{aligned}$$

- $\mathbf{y} = \langle y_1, y_2, \ldots, y_m \rangle$ is the dual variable and $\mathbf{b} = \langle b_1, b_2, \ldots, b_m \rangle$.

Primal-Dual Schema for SDP

Dual

Primal

$$\begin{aligned} \max\ & \mathbf{C} \bullet \mathbf{X} \\ \text{s.t. } & \mathbf{A_j} \bullet \mathbf{X} \leq b_j \quad \forall j \in [m] \\ & \mathbf{X} \succeq \mathbf{0} \end{aligned}$$

$$\begin{aligned} \min\ & \mathbf{b} \cdot \mathbf{y} \\ & \sum_{j=1}^{m} \mathbf{A_j} y_j \succeq \mathbf{C} \\ & \mathbf{y} \geq \mathbf{0} \end{aligned}$$

- $\mathbf{y} = \langle y_1, y_2, \ldots, y_m \rangle$ is the dual variable and $\mathbf{b} = \langle b_1, b_2, \ldots, b_m \rangle$.
- Strong Duality holds under Slater Condition.

Primal-Dual Schema for SDP

Primal

$$\begin{aligned} \max\ & \mathbf{C} \bullet \mathbf{X} \\ \text{s.t. } & \mathbf{A_j} \bullet \mathbf{X} \leq b_j \quad \forall j \in [m] \\ & \mathbf{X} \succeq \mathbf{0} \end{aligned}$$

Dual

$$\begin{aligned} \min\ & \mathbf{b} \cdot \mathbf{y} \\ & \sum_{j=1}^{m} \mathbf{A_j} y_j \succeq \mathbf{C} \\ & \mathbf{y} \geq \mathbf{0} \end{aligned}$$

- $\mathbf{y} = \langle y_1, y_2, \ldots, y_m \rangle$ is the dual variable and $\mathbf{b} = \langle b_1, b_2, \ldots, b_m \rangle$.
- Strong Duality holds under Slater Condition.
- Assume $\mathbf{A_1} = \mathbf{I}$ and $b_1 = R$ to get $\mathbf{Tr(X)} \leq R$ *i.e.* a simple scaling constant. Present in most of SDP relaxation combinatorial optimization problems.

Computing near optimal solution of a SDP using Primal-Dual Approach

- Let current guess for optimum be α.

Computing near optimal solution of a SDP using Primal-Dual Approach

- Let current guess for optimum be α.
- Algorithm either tries to construct a primal feasible PSD matrix $\mathbf{X}$ with objective value $> \alpha$ or a dual feasible solution with value at most $(1+\delta)\alpha$ for arbitrarily small $\delta > 0$.

Computing near optimal solution of a SDP using Primal-Dual Approach

- Let current guess for optimum be α.
- Algorithm either tries to construct a primal feasible PSD matrix $\mathbf{X}$ with objective value $> \alpha$ or a dual feasible solution with value at most $(1+\delta)\alpha$ for arbitrarily small $\delta > 0$.
- Iteratively constructs $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{X}^{(3)}, \ldots$.

Computing near optimal solution of a SDP using Primal-Dual Approach

- Let current guess for optimum be α.
- Algorithm either tries to construct a primal feasible PSD matrix $\mathbf{X}$ with objective value $> \alpha$ or a dual feasible solution with value at most $(1+\delta)\alpha$ for arbitrarily small $\delta > 0$.
- Iteratively constructs $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{X}^{(3)}, \ldots$.
- Starts with $\mathbf{X}^{(1)} = \frac{R}{n}\mathbf{I}$.

Computing near optimal solution of a SDP using Primal-Dual Approach

- Let current guess for optimum be α.
- Algorithm either tries to construct a primal feasible PSD matrix $\mathbf{X}$ with objective value $> \alpha$ or a dual feasible solution with value at most $(1+\delta)\alpha$ for arbitrarily small $\delta > 0$.
- Iteratively constructs $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{X}^{(3)}, \ldots$.
- Starts with $\mathbf{X}^{(1)} = \frac{R}{n}\mathbf{I}$.
- Takes help from an auxillary algorithm ORACLE. ORACLE's task is to:

Computing near optimal solution of a SDP using Primal-Dual Approach

- Let current guess for optimum be α.
- Algorithm either tries to construct a primal feasible PSD matrix $\mathbf{X}$ with objective value $> \alpha$ or a dual feasible solution with value at most $(1+\delta)\alpha$ for arbitrarily small $\delta > 0$.
- Iteratively constructs $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{X}^{(3)}, \ldots$.
- Starts with $\mathbf{X}^{(1)} = \frac{R}{n}\mathbf{I}$.
- Takes help from an auxillary algorithm ORACLE. ORACLE's task is to:
  - Find $\mathbf{X}$ and $\mathbf{y}$ such that $\mathbf{X} \succeq 0$ and $\mathbf{y} \geq 0$

Computing near optimal solution of a SDP using Primal-Dual Approach

- Let current guess for optimum be α.
- Algorithm either tries to construct a primal feasible PSD matrix $\mathbf{X}$ with objective value $> \alpha$ or a dual feasible solution with value at most $(1+\delta)\alpha$ for arbitrarily small $\delta > 0$.
- Iteratively constructs $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{X}^{(3)}, \ldots$.
- Starts with $\mathbf{X}^{(1)} = \frac{R}{n}\mathbf{I}$.
- Takes help from an auxillary algorithm ORACLE. ORACLE's task is to:
  - Find $\mathbf{X}$ and $\mathbf{y}$ such that $\mathbf{X} \succeq 0$ and $\mathbf{y} \geq 0$
  - Try to produce a feasible dual by the end whose value is at most $(1+\delta)\alpha$ for some arbitrarily small $\delta > 0$

MWT

MWT Algorithm

Fix $\epsilon < \frac{1}{2}$ and let $\epsilon' = -\ln(1-\epsilon)$. In every round t, for $t = 1, 2, 3, \ldots$

MWT

MWT Algorithm

Fix $\epsilon < \frac{1}{2}$ and let $\epsilon' = -\ln(1-\epsilon)$. In every round t, for $t = 1, 2, 3, \ldots$

1. Compute

$$\mathbf{W}^{(t)} = (1-\epsilon)^{\sum_{\tau=1}^{t-1} \mathbf{M}^{(t)}} = \exp\left(-\epsilon' \left(\sum_{\tau=1}^{t-1} \mathbf{M}^{(t)}\right)\right)$$

MWT

MWT Algorithm

Fix $\epsilon < \frac{1}{2}$ and let $\epsilon' = -\ln(1-\epsilon)$. In every round t, for $t = 1, 2, 3, \ldots$

1. Compute

$$\mathbf{W}^{(t)} = (1-\epsilon)^{\sum_{\tau=1}^{t-1} \mathbf{M}^{(t)}} = \exp\left(-\epsilon' \left(\sum_{\tau=1}^{t-1} \mathbf{M}^{(t)}\right)\right)$$

2. Use the density matrix $\mathbf{P}^{(t)} = \frac{\mathbf{W}^{(t)}}{\mathbf{Tr}(\mathbf{W}^{(t)})}$ and observe the event $\mathbf{M}^{(t)}$.

MWT

MWT Theorem

The Matrix Multiplicative Weights algorithm generates density matrices $\mathbf{P^{(1)}}, \mathbf{P^{(2)}}, \ldots, \mathbf{P^{(T)}}$ such that:

$$\sum_{t=1}^{T} \mathbf{M^{(t)}} \bullet \mathbf{P^{(t)}} \leq (1+\epsilon)\lambda_n(\sum_{t=1}^{T} \mathbf{M^{(t)}}) + \frac{\ln n}{\epsilon}$$

MWT

MWT Theorem

The Matrix Multiplicative Weights algorithm generates density matrices $\mathbf{P^{(1)}}, \mathbf{P^{(2)}}, \ldots, \mathbf{P^{(T)}}$ such that:

$$\sum_{t=1}^{T} \mathbf{M^{(t)}} \bullet \mathbf{P^{(t)}} \leq (1+\epsilon)\lambda_n\left(\sum_{t=1}^{T} \mathbf{M^{(t)}}\right) + \frac{\ln n}{\epsilon}$$

Proof Idea

Track changes in $\mathbf{Tr}(\mathbf{W}^{(t)})$ over time and use Golden-Thompson inequality.

MWT

Proof

$$
\begin{aligned}
\mathbf{Tr}(\mathbf{W}^{(t+1)}) &= \mathbf{Tr}(\exp(-\epsilon' \sum_{\tau=1}^{t} \mathbf{M}^{(\tau)})) \\
&\leq \mathbf{Tr}(\exp(-\epsilon' \sum_{\tau=1}^{t-1} \mathbf{M}^{(\tau)}) \exp(-\epsilon' \mathbf{M}^{(t)})) \\
&= \mathbf{W}^{(t)} \bullet \exp(-\epsilon' \mathbf{M}^{(t)}) \\
&\leq \mathbf{W}^{(t)} \bullet (\mathbf{I} - \epsilon \mathbf{M}^{(t)}) \\
&= \mathbf{Tr}(\mathbf{W}^{(t)}) \cdot (1 - \epsilon \mathbf{M}^{(t)} \bullet \mathbf{P}^{(t)}) \\
&\leq \mathbf{Tr}(\mathbf{W}^{(t)}) \cdot \exp(-\epsilon \mathbf{M}^{(t)} \bullet \mathbf{P}^{(t)})
\end{aligned}
$$

MWT

Proof

Since $\mathbf{Tr}(\mathbf{W}^1) = \mathbf{Tr}(\mathbf{I}) = n$, by induction,

$$\mathbf{Tr}(\mathbf{W}^{T+1}) \leq n \exp(-\epsilon \sum_{t=1}^{T} \mathbf{M}^{(\mathbf{t})} \bullet \mathbf{P}^{(\mathbf{t})})$$

On the other hand, since $\mathbf{Tr}(e^{\mathbf{A}}) = \sum_{k=1}^{n} e^{\lambda_k(\mathbf{A})} \geq e^{\lambda_n(\mathbf{A})}$,

$$\mathbf{Tr}(\mathbf{W}^{T+1}) = \mathbf{Tr}(\exp(-\epsilon' \sum_{t=1}^{T} \mathbf{M}^{(t)})) \geq \exp(-\epsilon' \lambda_n(\sum_{t=1}^{T} \mathbf{M}^{(t)}))$$

Thus,

$$\exp(-\epsilon' \lambda_n(\sum_{t=1}^{T} \mathbf{M}^{(t)})) \leq n \exp(-\epsilon \sum_{t=1}^{T} \mathbf{M}^{(\mathbf{t})} \bullet \mathbf{P}^{(\mathbf{t})})$$

MWT

MWT

- A feasible dual is the one in which the slack matrix $\sum_{j=1}^{m} \mathbf{A}_j y_j - \mathbf{C} \succeq 0$

MWT

- A feasible dual is the one in which the slack matrix $\sum_{j=1}^{m} \mathbf{A}_j y_j - \mathbf{C} \succeq 0$
- MWT theorem replaces ORACLE's task to be replaced by the following steps:

MWT

- A feasible dual is the one in which the slack matrix $\sum_{j=1}^{m} \mathbf{A}_j y_j - \mathbf{C} \succeq 0$
- MWT theorem replaces ORACLE's task to be replaced by the following steps:
  - Focus on finding a slack matrix which has a non-negative inner product with the current solution matrix $\mathbf{X}^{(t)}$

MWT

- A feasible dual is the one in which the slack matrix $\sum_{j=1}^{m} \mathbf{A}_j y_j - \mathbf{C} \succeq 0$
- MWT theorem replaces ORACLE's task to be replaced by the following steps:
  - Focus on finding a slack matrix which has a non-negative inner product with the current solution matrix $\mathbf{X}^{(t)}$
  - If the ORACLE manages to do this even for a small number of steps, MWT theorem guarantees that the average slack matrix over these steps would be almost psd

MWT

- A feasible dual is the one in which the slack matrix $\sum_{j=1}^{m} \mathbf{A}_j y_j - \mathbf{C} \succeq 0$
- MWT theorem replaces ORACLE's task to be replaced by the following steps:
  - Focus on finding a slack matrix which has a non-negative inner product with the current solution matrix $\mathbf{X}^{(t)}$
  - If the ORACLE manages to do this even for a small number of steps, MWT theorem guarantees that the average slack matrix over these steps would be almost psd
  - Saving:
    Producing a psd matrix $\rightarrow$ linear condition

Computing near optimal solution of a SDP using Primal-Dual Approach

Description of ORACLE

ORACLE searches for a vector $\mathbf{b}$ from the polytope $\mathcal{D}_\alpha = \{\mathbf{y} : \mathbf{y} \geq \mathbf{0},\ \mathbf{b} \cdot \mathbf{y} \leq \alpha\}$ such that

$$\sum_{j=1}^{m} (\mathbf{A}_j \bullet \mathbf{X}^{(t)}) y_j - (\mathbf{C} \bullet \mathbf{X}^{(t)}) \geq 0 \qquad (1)$$

If ORACLE succeeds in finding such a $\mathbf{y}$ then $\mathbf{X}^{(t)}$ is either primal infeasible or has value $\mathbf{C} \bullet \mathbf{X}^{(t)} \leq \alpha$.

Proof: Suppose this is not the case. Then

$$\sum_{j=1}^{m} (\mathbf{A}_j \bullet \mathbf{X}^{(t)}) y_j - (\mathbf{C} \bullet \mathbf{X}^{(t)}) \leq \sum_{j=1}^{m} b_j y_j - (\mathbf{C} \bullet \mathbf{X}^{(t)}) < \alpha - \alpha = 0$$

which contradicts (1)

Computing near optimal solution of a SDP using Primal-Dual Approach

- If ORACLE declares that there is no such $y \in \mathcal{D}_\alpha$, then $\mathbf{X}^{(t)}$ is a primal feasible solution with objective value at least α
- $\mathbf{y}$ need not be dual feasible
- Primal-Dual SDP algorithm depends on width parameter, ρ.

width of ORACLE

Smallest ρ such that for every primal candidate $\mathbf{X}$, the vector $y \in \mathcal{D}_\alpha$ returned by the ORACLE satisfies $\|\mathbf{A}_j y_j - \mathbf{C}\| \leq \rho$

- Higher width equals slow progress

Computing near optimal solution of a SDP using Primal-Dual Approach

Primal-Dual Algorithm for SDP

Set $\mathbf{X}^{(1)} = \frac{R}{n}\mathbf{I}$. Let $\epsilon = \frac{\delta\alpha}{2\rho R}$, and let $\epsilon' = -\ln(1-\epsilon)$. Let $T = \frac{8\rho^2 R^2 \ln(n)}{\delta^2\alpha^2}$. For $t = 1, 2, \ldots, T$:

1. Run the ORACLE with candidate solution $\mathbf{X}^{(t)}$.
2. If the ORACLE fails, stop and output $\mathbf{X}^{(t)}$.
3. Else, let $\mathbf{y}^{(t)}$ be the vector generated by ORACLE.
4. Let $\mathbf{M}^{(t)} = (\Sigma_{j=1}^{m}\mathbf{A}_j y_j^{(t)} - \mathbf{C} + \rho\mathbf{I})/2\rho$.
5. Compute $\mathbf{W}^{(t+1)} = (1-\epsilon)^{\Sigma_{\tau=1}^{t}\mathbf{M}^{(\tau)}} = exp\Big(-\epsilon'(\Sigma_{\tau=1}^{t}\mathbf{M}^{(\tau)})\Big)$.
6. Set $\mathbf{X}^{(t+1)} = \frac{R\mathbf{W}^{(t+1)}}{\mathbf{Tr}(\mathbf{W}^{(t+1)})}$ and continue.

Computing near optimal solution of a SDP using Primal-Dual Approach

Theorem 1

In the Primal-Dual SDP algorithm, assume that the ORACLE never fails for $T = \frac{8\rho^2 R^2 \ln(n)}{\delta^2\alpha^2}$ iterations. Let $\bar{\mathbf{y}} = \frac{\delta\alpha}{R}\mathbf{e}_1 + \frac{1}{T}\Sigma_{t=1}^{T}\mathbf{y}^{(t)}$. Then $\bar{\mathbf{y}}$ is a feasible dual solution with objective value at most $(1+\delta)\alpha$.

Proof.

...mmw used... □

Outline

Approximating MAXCUT SDP for d-regular graphs

MAXCUT SDP in vector and matrix form (ignoring a factor of $\frac{1}{4}$).

$$\max \sum_{\{i,j\}\in E} \|\mathbf{v}_i - \mathbf{v}_j\|^2$$
$$\forall i \in [n] : \|\mathbf{v}_i\|^2 \le 1$$

$$\max \mathbf{C} \bullet \mathbf{X}$$
$$\forall i \in [n] : \mathbf{X}_{ii} \le 1$$
$$\mathbf{X} \succeq \mathbf{0}$$

Dual of SDP:

$$\min \Sigma_{i=1}^{n} x_i$$
$$diag(\mathbf{x}) \succeq \mathbf{C}$$
$$\forall i \in [n] : x_i \ge 0$$

- $\mathbf{C}$ is the combinatorial Laplacian of the graph.
- $diag(\mathbf{x})$ is the diagonal matrix with the vector $\mathbf{x}$ on the diagonal.

Approximating MAXCUT SDP for d-regular graphs

Combinatorial Laplacian of a graph

$$\mathbf{C} = \mathbf{D} - \mathbf{A}$$

where $\mathbf{D}$ is the degree matrix of the graph (diagonal matrix with diagonal entries as the number of edges incident on that vertex) and $\mathbf{A}$ is the adjacency matrix the graph.

- Intuitively, $\mathbf{C}_{ii} = \Sigma_{i \neq j} c_{\{i,j\}}$ and $\mathbf{C}_{ij} = -c_{ij}$
- If d is maximum degree of the graph, then $\mathbf{0} \preceq \mathbf{C} \preceq 2d\mathbf{I}$. (Proof: Using $\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$)
- If $\mathbf{v}_i$ are the vectors obtained from the Cholesky decomposition of $\mathbf{X}$, then $\mathbf{C} \bullet \mathbf{X} = \Sigma_{\{i,j\} \in E} c_{\{i,j\}} \|\mathbf{v}_i - \mathbf{v}_j\|^2$
- When G is d-regular, $\mathbf{C} = \mathbf{I} - \frac{1}{d}\mathbf{A}$

Approximating MAXCUT SDP for d-regular graphs

- $nd \le \alpha \le 3nd$ (Property of d-regular graph).
- Trace of optimal $\mathbf{X}$ is n.
- If width parameter ρ is $O(d)$, then number of iterations is $O(\log n)$.
- Each invocation of ORACLE and matrix exponentiation takes $\widetilde{O}(m)$ time.
  - Approximate Matrix Exponentiation by Johnson-Lindenstrauss Dimension Reduction.
  - Number of non-zero matrix entries in C is $O(m)$.

Approximating MAXCUT SDP for d-regular graphs: Description of ORACLE

- Given a candidate solution $\mathbf{X}$, find a vector $\mathbf{x} \geq 0$ such that $\Sigma_i \mathbf{x}_i \leq \alpha$ and $\Sigma_i x_i \mathbf{X}_{ii} - \mathbf{C} \bullet \mathbf{X} \geq 0$
- Intuitively, to make $\Sigma_i x_i \mathbf{X}_{ii}$ as large as possible, make x_i large whenever $\mathbf{X}_{ii}$ is large.
- However, also ensure that $x_i \leq O(\frac{\alpha}{n}) = O(d)$ to ensure the width bound: $\|diag(\mathbf{x}) - \mathbf{C}\| \leq O(d)$

Approximating MAXCUT SDP for d-regular graphs: Description of ORACLE

1. $\mathbf{C} \bullet \mathbf{X} \leq \alpha$. Set all $x_i = \frac{\alpha}{n}$. Since $\Sigma_i \mathbf{X}_{ii} = \mathbf{Tr}(\mathbf{X}) = n$,

$$\Sigma_i x_i \mathbf{X}_{ii} - \mathbf{C} \bullet \mathbf{X} \geq \frac{\alpha}{n} \Sigma_i \mathbf{X}_{ii} - \alpha = 0$$

2. $\mathbf{C} \bullet \mathbf{X} \geq \alpha$. Let $\mathbf{C} \bullet \mathbf{X} = \lambda\alpha$ for some $\lambda \geq 1$. Since $\mathbf{C} \preceq 2d\mathbf{I}$, $\lambda\alpha = \mathbf{C} \bullet \mathbf{X} \leq 2nd$. Also, $\alpha \geq nd$. Hence, $\lambda \leq 2$.
   Let $S := \{i : \mathbf{X}_{ii} \geq \lambda\}$. Let $k := \Sigma_{i \in S} \mathbf{X}_{ii}$.
   If $k \geq \delta_1 n$ for some constant δ_1, set $x_i = \frac{\lambda\alpha}{k} \forall i \in S$ and $x_i = 0 \forall i \notin S$. Then $\Sigma_i x_i = |S| \frac{\lambda\alpha}{k} \leq \alpha$ since $k \geq \Sigma_i \mathbf{X}_{ii} \geq \lambda|S|$. Then
   $\Sigma_i x_i \mathbf{X}_{ii} - \mathbf{C} \bullet \mathbf{X} = \frac{\lambda\alpha}{k} \Sigma_{i \in S} \mathbf{X}_{ii} - \lambda\alpha \geq 0$.

Approximating MAXCUT SDP for d-regular graphs: Description of ORACLE

3. Otherwise, we construct a feasible primal solution $\mathbf{X}$ of value $\geq (1-\delta)\alpha$. Let $\mathbf{v}_i$ be Cholesky decomposition of $\mathbf{X}$.
   Set $\mathbf{v}'_i := \mathbf{v}_i$ for $i \notin S$ and $\mathbf{v}'_i = \mathbf{v}_0$ for $i \in S$, for some fixed unit vector $\mathbf{v}_0$. Let $\widetilde{\mathbf{X}}$ be the Gram matrix of $\mathbf{v}'$.
   Let E_S be the set of edges with at least one endpoint in S.
   We have $\mathbf{C} \bullet (\widetilde{\mathbf{X}} - \mathbf{X}) \geq -\Sigma_{\{i,j\} \in E_S} \|\mathbf{v}_i - \mathbf{v}_j\|^2$. Also,

$$\sum_{\{i,j\} \in E_S} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \leq \sum_{\{i,j\} \in E_S} 2[\|\mathbf{v}_i\|^2 + \|\mathbf{v}_j\|^2]$$
$$\leq 2d \sum_{i \in S} \|\mathbf{v}_i\|^2 + 2d\lambda|S| \leq 4dk \leq 4\delta_1 nd$$

   Hence $\mathbf{C} \bullet \widetilde{\mathbf{X}} \geq \lambda\alpha - 4\delta_1 nd$. For error parameter δ, choose $\delta_1 \leq \frac{\delta\lambda}{4}$ to lower bound RHS by $(1-\delta)\lambda\alpha$. So $\mathbf{X}^* = \frac{1}{\lambda}\widetilde{\mathbf{X}}$ is feasible with value $\geq (1-\delta)\alpha$.

Primal-Dual approach: Extension to minimization problems

- ORACLE finds a vector $\mathbf{y}$ from the polytope $\mathcal{D}_\alpha = \{\mathbf{y} : \mathbf{y} \geq \mathbf{0}, \mathbf{b} \cdot \mathbf{y} \geq \alpha\}$ such that $\Sigma_{j=1}^{m}(\mathbf{A}_j \bullet \mathbf{X})y_j - (\mathbf{C} \bullet \mathbf{X}) < 0$.
- Matrix exponentiation is computed with base $(1+\epsilon)$ rather than $(1-\epsilon)$.
- Allow ORACLE to find a matrix $\mathbf{F}^{(t)}$ such that for all primal feasible $\mathbf{X}$, $\mathbf{F}^{(t)} \bullet \mathbf{X} \leq \mathbf{C} \bullet \mathbf{X}$ and a vector $\mathbf{y}^{(t)} \in \mathcal{D}_\alpha$ such that

$$\Sigma_{j=1}^{m}(\mathbf{A}_j \bullet \mathbf{X}^{(t)})y_j^{(t)} - (\mathbf{F}^{(t)} \bullet \mathbf{X}) \leq 0$$

  - We can replace $\mathbf{C}$ by $\mathbf{F}^{(t)}$ (which can be decided by us). If $\mathbf{F}^{(t)} \preceq \mathbf{C}$, then since any primal feasible $\mathbf{X}$ is PSD, we have $\mathbf{F}^{(t)} \bullet \mathbf{X} \leq \mathbf{C} \bullet \mathbf{X}$. So if suffices to find $\mathbf{F}^{(t)} \preceq \mathbf{C}$.
  - This is done to reduce the width parameter, ρ.
- $M := (\Sigma_{j=1}^{m}\mathbf{A}_j y_j^{(t)} - \mathbf{F}^{(t)} + \rho\mathbf{I})/2\rho$

Primal-Dual approach: Extension to minimization problems

Theorem

In the modified Primal-Dual Algorithm for a minimization SDP as described in the previous slide, if the ORACLE *never fails for* $T = \frac{8\rho^2 R^2 \ln(n)}{\delta^2\alpha^2}$ *iterations, then* $\bar{\mathbf{y}} = \frac{\delta\alpha}{R}\mathbf{e}_1 + \frac{1}{T}\Sigma_{t=1}^{T}\mathbf{y}^{(t)}$ *is a feasible dual solution with dual objective value at least* $(1-\delta)\alpha$*.*

Proof.

Similar to maximization theorem's proof. □

Matrix exponentiation

$$e^{\mathbf{M}} = \sum_{i=0}^{\infty} \frac{\mathbf{M}^i}{i!} = \mathbf{I} + \frac{\mathbf{M}}{1} + \frac{\mathbf{M}^2}{2!} + \cdots$$

- $e^{\mathbf{0}} = \mathbf{I}$
- $e^{a\mathbf{X}} e^{b\mathbf{X}} = e^{(a+b)\mathbf{X}}$ where a and b are reals
- $\exp(\mathbf{A} + \mathbf{B}) \neq \exp(\mathbf{A}) + \exp(\mathbf{B})$ in general.
- $\exp(\mathbf{A}^T) = (\exp\ \mathbf{A})^T$
- $\exp(\mathbf{A})$ is PSD for all symmetric A since $\exp(\mathbf{A}) = \exp(\frac{1}{2}\mathbf{A})^T \exp(\frac{1}{2}\mathbf{A})$.
- Cholesky decomposition of $\exp(\mathbf{A})$ is $\exp(\frac{1}{2}\mathbf{A})$.
- Golden-Thompson Inequality: $\mathbf{Tr}\ \exp(\mathbf{A} + \mathbf{B}) \leq \mathbf{Tr}\ (\exp(\mathbf{A}) \exp(\mathbf{B}))$

Outline

Outline

Undirected BALANCED SEPARATOR

Undirected Balanced Separator Problem

Given a graph $G(V, E)$ with $|V| = n, |E| = m$, and capacity c_e on edge $e \in E$, find the c-balanced cut with minimum capacity. A cut $(S, \overline{S})$ is called c-balanced if $|S| \geq cn$ and $|\overline{S}| \geq cn$.

Undirected BALANCED SEPARATOR

Undirected Balanced Separator Problem

Given a graph $G(V,E)$ with $|V| = n, |E| = m$, and capacity c_e on edge $e \in E$, find the c-balanced cut with minimum capacity. A cut $(S, \overline{S})$ is called c-balanced if $|S| \geq cn$ and $|\overline{S}| \geq cn$.

t pseudo-approximation

A t pseudo-approximation for minimum c-BALANCED SEPARATOR problem is a c'-balanced cut for some constant c' whose expansion is within a factor of t of that of minimum c-BALANCED SEPARATOR ($c' \leq ct$).

Undirected BALANCED SEPARATOR

Undirected Balanced Separator Problem

Given a graph $G(V, E)$ with $|V| = n, |E| = m$, and capacity c_e on edge $e \in E$, find the c-balanced cut with minimum capacity. A cut $(S, \overline{S})$ is called c-balanced if $|S| \geq cn$ and $|\overline{S}| \geq cn$.

Theorem

An $O(\log n)$ pseudo-approximation to the minimum c-BALANCED SEPARATOR can be computed in $\tilde{O}(m + n^{1.5})$ time using $O(\log^2(n))$ single commodity flow computations.

Undirected BALANCED SEPARATOR

SDP

$$\min \sum_{e=\{i,j\}\in E} c_e \|\mathbf{v}_i - \mathbf{v}_j\|^2$$

$$\forall i : \|\mathbf{v}_i\|^2 = 1$$

$$\forall p : \sum_{j=1}^{k-1} \|\mathbf{v}_{i_j} - \mathbf{v}_{i_{j+1}}\|^2 \geq \|\mathbf{v}_{i_1} - \mathbf{v}_{i_k}\|^2$$

$$\forall S : \sum_{i,j\in S} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq an^2$$

$$\min \mathbf{C} \bullet \mathbf{X}$$

$$\forall i : \mathbf{X}_{ii} = 1$$

$$\forall p : \mathbf{T}_p \bullet \mathbf{X} \geq 0$$

$$\forall S : \mathbf{K}_S \bullet \mathbf{X} \geq an^2$$

$$X \succeq 0$$

Undirected BALANCED SEPARATOR

Notations

- Assign vectors $\mathbf{v}_i$ to nodes in G. Let $\mathbf{X}$ be the Gram matrix of these vectors.
- $\mathbf{C}$ is the Combinatorial Laplacian of the graph.
- For any subset S of the nodes, $\mathbf{K}_S$ is defined to be the Laplacian of the graph where all nodes in S are connected by edges, all other edges are absent.
- $|S| \geq (1-\epsilon)n$
- For a generic path $p = (i_1, i_2, \ldots, i_k)$ of nodes in the complete graph, $\mathbf{T}_p$ is the difference of the Laplacian of p and that of a single edge connecting its endpoints i_1 and i_k.
- $a = 4[c(1-c) - \epsilon]$

Undirected BALANCED SEPARATOR

Dual Program

$$\max \sum_i x_i + an^2 \sum_S z_S$$
$$\mathbf{C} \succeq \mathrm{diag}(\mathbf{x}) + \sum_p f_p \mathbf{T}_p + \sum_S z_S \mathbf{K}_S$$
$$\forall p, S : f_p, z_S \geq 0$$

Notations

- Variable x_i for every node i, f_p for every path p and z_S for every set S of size at least $(1-\epsilon)n$

Undirected BALANCED SEPARATOR : Oracle

Undirected BALANCED SEPARATOR : Oracle

- Let α be the current guess of the solution.

Undirected BALANCED SEPARATOR : Oracle

- Let α be the current guess of the solution.
- Let $\mathbf{X}$ be the current solution generated by the Primal-Dual algorithm.

Undirected BALANCED SEPARATOR : Oracle

- Let α be the current guess of the solution.
- Let $\mathbf{X}$ be the current solution generated by the Primal-Dual algorithm.
- $\mathbf{Tr}(\mathbf{X}) = n$

Undirected BALANCED SEPARATOR : Oracle

- Let α be the current guess of the solution.
- Let $\mathbf{X}$ be the current solution generated by the Primal-Dual algorithm.
- $\mathbf{Tr}(\mathbf{X}) = n$
- Using *MWT* Theorem for minimization problems, ORACLE needs to find variables $x_i, f_p \geq 0, z_S \geq 0$ and a matrix $\mathbf{F} \preceq \mathbf{C}$ such that $\sum_i x_i + an^2 \sum_S z_S \geq \alpha$ and

$$\text{diag}(\mathbf{x}) \bullet \mathbf{X} + \sum_p f_p(\mathbf{T}_p \bullet \mathbf{X}) + \sum_S z_S(\mathbf{K} \bullet \mathbf{X}) - (\mathbf{F} \bullet \mathbf{X}) \leq 0$$

Undirected BALANCED SEPARATOR : Oracle

- Let α be the current guess of the solution.
- Let $\mathbf{X}$ be the current solution generated by the Primal-Dual algorithm.
- $\mathbf{Tr}(\mathbf{X}) = n$
- Using *MWT* Theorem for minimization problems, ORACLE needs to find variables $x_i, f_p \geq 0, z_S \geq 0$ and a matrix $\mathbf{F} \preceq \mathbf{C}$ such that $\sum_i x_i + an^2 \sum_S z_S \geq \alpha$ and

  $$\text{diag}(\mathbf{x}) \bullet \mathbf{X} + \sum_p f_p(\mathbf{T}_p \bullet \mathbf{X}) + \sum_S z_S(\mathbf{K} \bullet \mathbf{X}) - (\mathbf{F} \bullet \mathbf{X}) \leq 0$$
- If the ORACLE succeeds, then the matrix returned as feedback is $\mathbf{M} = \text{diag}(\mathbf{x}) + \sum_p f_p \mathbf{T}_p + \sum_S z_S \mathbf{K}_S - \mathbf{F}$

Undirected BALANCED SEPARATOR : Oracle

- Let α be the current guess of the solution.
- Let $\mathbf{X}$ be the current solution generated by the Primal-Dual algorithm.
- $\mathbf{Tr}(\mathbf{X}) = n$
- Using *MWT* Theorem for minimization problems, ORACLE needs to find variables $x_i, f_p \geq 0, z_S \geq 0$ and a matrix $\mathbf{F} \preceq \mathbf{C}$ such that $\sum_i x_i + an^2 \sum_S z_S \geq \alpha$ and
$$\text{diag}(\mathbf{x}) \bullet \mathbf{X} + \sum_p f_p(\mathbf{T}_p \bullet \mathbf{X}) + \sum_S z_S(\mathbf{K} \bullet \mathbf{X}) - (\mathbf{F} \bullet \mathbf{X}) \leq 0$$
- If the ORACLE succeeds, then the matrix returned as feedback is $\mathbf{M} = \text{diag}(\mathbf{x}) + \sum_p f_p \mathbf{T}_p + \sum_S z_S \mathbf{K}_S - \mathbf{F}$
- ORACLE needs to ensure a width of $\tilde{O}(\frac{\alpha}{n})$

Undirected BALANCED SEPARATOR : Oracle

Implementation: Basic Idea

Undirected BALANCED SEPARATOR : Oracle

Implementation: Basic Idea

- Given a candidate solution $\mathbf{X}$, check if all $\mathbf{X}_{ii}$ are $O(1)$.

Undirected BALANCED SEPARATOR : Oracle

Implementation: Basic Idea

- Given a candidate solution $\mathbf{X}$, check if all $\mathbf{X}_{ii}$ are $O(1)$.
  - If a significant fraction of them aren't, punish $\mathbf{X}$ by setting x_i in a similar way as in MAX-CUT

Undirected BALANCED SEPARATOR : Oracle

Implementation: Basic Idea

- Given a candidate solution $\mathbf{X}$, check if all $\mathbf{X}_{ii}$ are $O(1)$.
  - If a significant fraction of them aren't, punish $\mathbf{X}$ by setting x_i in a similar way as in MAX-CUT
- Check if $\mathbf{K}_V \bullet \mathbf{X} \geq \Omega(n^2)$

Undirected BALANCED SEPARATOR : Oracle

Implementation: Basic Idea

- Given a candidate solution $\mathbf{X}$, check if all $\mathbf{X}_{ii}$ are $O(1)$.
  - If a significant fraction of them aren't, punish $\mathbf{X}$ by setting x_i in a similar way as in MAX-CUT
- Check if $\mathbf{K}_V \bullet \mathbf{X} \geq \Omega(n^2)$
  - If not, set z_S appropriately to punish $\mathbf{X}$.

Undirected BALANCED SEPARATOR : Oracle

Implementation: Basic Idea

- Given a candidate solution $\mathbf{X}$, check if all $\mathbf{X}_{ii}$ are $O(1)$.
  - If a significant fraction of them aren't, punish $\mathbf{X}$ by setting x_i in a similar way as in MAX-CUT
- Check if $\mathbf{K}_V \bullet \mathbf{X} \geq \Omega(n^2)$
  - If not, set z_S appropriately to punish $\mathbf{X}$.
- If both the above conditions are satisfied, do a flow computation and interpret f_p variables as multicommodity flow in the graph.

Undirected BALANCED SEPARATOR : Oracle

Lemma 1

For at least a $\frac{a}{32}$ fraction of directions $\mathbf{u}$, there are efficiently computable sets S and T, each of size at least $\frac{a}{128n}$, such that for any $i \in S$ and $j \in T$, $(\mathbf{v}_j - \mathbf{v}_i) \cdot \mathbf{u} \geq \frac{a}{48\sqrt{n}}$

Undirected BALANCED SEPARATOR : Oracle

Lemma 1

For at least a $\frac{a}{32}$ fraction of directions $\mathbf{u}$, there are efficiently computable sets S and T, each of size at least $\frac{a}{128n}$, such that for any $i \in S$ and $j \in T$, $(\mathbf{v}_j - \mathbf{v}_i) \cdot \mathbf{u} \geq \frac{a}{48\sqrt{n}}$

Proof Idea

Consider the Gaussian beahviour of projections on a random vector $\mathbf{u}$ the median value of $\mathbf{v}_i \cdot \mathbf{u} = m$

$S = \{i : \mathbf{v}_i \cdot \mathbf{u} \leq m - \delta\}$

$T = \{i : \mathbf{v}_i \cdot \mathbf{u} \geq m\}$

$\delta = \frac{a}{48\sqrt{n}}$

Undirected BALANCED SEPARATOR : Oracle

Lemma 2

Let $S \subseteq V$ be a set of nodes of size $\Omega(n)$. Suppose for all $i \in S$, vectors $\mathbf{v}_i$ of length $O(1)$ are given such that $\sum_{i,j \in S} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \Omega(n^2)$, and a quantity α. Then there is an algorithm, which, using a single max-flow computation, either outputs a valid $O(\frac{\log(n)\alpha}{n})$-regular flow f_p such that $\sum_{ij} f_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, or a c'-balanced cut of expansion $O(\log(n)\frac{\alpha}{n})$.

Undirected BALANCED SEPARATOR : Oracle

Lemma 2

Let $S \subseteq V$ be a set of nodes of size $\Omega(n)$. Suppose for all $i \in S$, vectors $\mathbf{v}_i$ of length $O(1)$ are given such that $\sum_{i,j \in S} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \Omega(n^2)$, and a quantity α. Then there is an algorithm, which, using a single max-flow computation, either outputs a valid $O(\frac{\log(n)\alpha}{n})$-regular flow f_p such that $\sum_{ij} f_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, or a c'-balanced cut of expansion $O(\log(n)\frac{\alpha}{n})$.

Proof Idea

Using Lemma 1

Undirected BALANCED SEPARATOR : Oracle

ORACLE Description

Given a candidate solution $\mathbf{X}$, the ORACLE runs the following steps (set all unspecified variables, including $\mathbf{F}$ to 0)

1. Assume, WLOG, $\mathbf{X}_{11} \leq \mathbf{X}_{22} \leq \cdots \leq \mathbf{X}_{nn}$. Define $h = (1-\epsilon)n + 1$. If $\mathbf{X}_{hh} \geq 2$, set $x_i = -\frac{\alpha}{\epsilon n}$ for $i \geq k$ and $x_i = \frac{2\alpha}{(1-\epsilon)n}$ for $i < k$. Then,

$$\begin{aligned}
\mathrm{diag}(\mathbf{x}) \bullet \mathbf{X} &= \sum_{i \geq k} -\frac{\alpha}{\epsilon n}\mathbf{X}_{ii} + \sum_{i<k} \frac{2\alpha}{(1-\epsilon)n}\mathbf{X}_{ii} \\
&\leq -\frac{\alpha}{\epsilon n} \cdot 2 \cdot \epsilon n + \frac{2\alpha}{(1-\epsilon)n} \cdot (n - 2\epsilon n) \leq 0
\end{aligned}$$

Since all $x_i = O(\frac{\alpha}{n})$, $\|\mathrm{diag}(\mathbf{x})\| \leq O(\frac{\alpha}{n})$

Undirected BALANCED SEPARATOR : Oracle

ORACLE Description

Given a candidate solution $\mathbf{X}$, the ORACLE runs the following steps (set all unspecified variables, including $\mathbf{F}$ to 0)

2. Assume that for all but ϵn exceptional nodes i, $\mathbf{X}_{ii} \leq 2$. Let $W := \{i : \mathbf{X}_{ii} \leq 2\}$ and $S := V \backslash W$. Since $|S| \geq (1-\epsilon)n$ so we have $\mathbf{K}_S \bullet \mathbf{X} \geq an^2$ in the SDP. If $\mathbf{K}_S \bullet \mathbf{X} \leq \frac{an^2}{2}$, choose $z_S = \frac{2\alpha}{an^2}$ and all $x_i = -\frac{\alpha}{n}$. Then,

$$\left(-\frac{\alpha}{n}\mathbf{I} + \frac{2\alpha}{an^2}\mathbf{K}_S\right) \bullet \mathbf{X} \leq \alpha - \alpha = 0$$

   Since, $\mathbf{0} \preceq \mathbf{K}_S \preceq n\mathbf{I}$, $\| -\frac{\alpha}{n}\mathbf{I} + \frac{2\alpha}{an^2}\mathbf{K}_S\| \leq O(\frac{\alpha}{n})$

Undirected BALANCED SEPARATOR : Oracle

ORACLE Description

Given a candidate solution $\mathbf{X}$, the ORACLE runs the following steps (set all unspecified variables, including $\mathbf{F}$ to 0)

3. Assume $\mathbf{K}_S \bullet \mathbf{X} \geq \frac{an^2}{2}$, and $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ be the vectors obtained from the Cholesky decomposition of $\mathbf{X}$. For all nodes $i \in S$, $\|\mathbf{v}_i\|^2 \leq 2$. Also, $\mathbf{K}_S \bullet \mathbf{X} \geq \frac{an^2}{2}$ implies $\sum_{i,j \in S} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \frac{an^2}{2}$.
   Try satisfying path inequalities by using multicommodity flow and Lemma 2 (either we can find a nice flow which gives substantial feedback or a cut with desired expansion, i.e, a near-optimal integral solution).

Undirected BALANCED SEPARATOR : Oracle

ORACLE Description : Notations

- f_p is the flow on a path p.
- f_e is the flow on edge e; $f_e := \sum_{p \ni e} f_p$.
- f_i is the total flow through a node; $f_i = \sum_{p \in \mathcal{P}_i}$ where $\mathcal{P}_i$ is the set of paths starting from i.
- f_{ij} is total flow between nodes i, j; $f_{ij} = \sum_{p \in \mathcal{P}_{ij}} f_p$ where $\mathcal{P}_{ij}$ is the set of paths from i to j.
- A valid d-regular flow satisfies the following constraints:
  - $\forall e : f_e \leq c_e$
  - $\forall i : f_i \leq d$

Undirected BALANCED SEPARATOR : Oracle

ORACLE Description

Undirected BALANCED SEPARATOR : Oracle

ORACLE Description

- Apply Lemma 2 to set S.

Undirected BALANCED SEPARATOR : Oracle

ORACLE Description

- Apply Lemma 2 to set S.
- If a cut of desired expansion is found, stop.

Undirected BALANCED SEPARATOR : Oracle

ORACLE Description

- Apply Lemma 2 to set S.
- If a cut of desired expansion is found, stop.
- If a valid d-regular flow is obtained which satisfies $\sum_{ij} f_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, where $d = O(\frac{\log(n)\alpha}{n})$.

Undirected BALANCED SEPARATOR : Oracle

ORACLE Description

- Apply Lemma 2 to set S.
- If a cut of desired expansion is found, stop.
- If a valid d-regular flow is obtained which satisfies $\sum_{ij} f_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, where $d = O(\frac{\log(n)\alpha}{n})$.
  - $F :=$ Laplacian of the weighed graph with edge weights f_e.

Undirected BALANCED SEPARATOR : Oracle

ORACLE Description

- Apply Lemma 2 to set S.
- If a cut of desired expansion is found, stop.
- If a valid d-regular flow is obtained which satisfies $\sum_{ij} f_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, where $d = O(\frac{\log(n)\alpha}{n})$.
  - $F :=$ Laplacian of the weighed graph with edge weights f_e.
  - Capacity constraints $f_e \leq c_e$ imply that $\mathbf{F} \preceq \mathbf{C}$

Undirected BALANCED SEPARATOR : Oracle

ORACLE Description

- Apply Lemma 2 to set S.
- If a cut of desired expansion is found, stop.
- If a valid d-regular flow is obtained which satisfies $\sum_{ij} f_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, where $d = O(\frac{\log(n)\alpha}{n})$.
  - $F :=$ Laplacian of the weighed graph with edge weights f_e.
  - Capacity constraints $f_e \leq c_e$ imply that $\mathbf{F} \preceq \mathbf{C}$
  - $D :=$ Laplacian of the complete graph where only edges $\{i, j\}$ with $i \in S$ and $j \in T$ have weight f_{ij} and rest have 0 weight.

Undirected BALANCED SEPARATOR : Oracle

ORACLE Description

- Apply Lemma 2 to set S.
- If a cut of desired expansion is found, stop.
- If a valid d-regular flow is obtained which satisfies $\sum_{ij} f_{ij}\|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, where $d = O(\frac{\log(n)\alpha}{n})$.
  - $F :=$ Laplacian of the weighed graph with edge weights f_e.
  - Capacity constraints $f_e \leq c_e$ imply that $\mathbf{F} \preceq \mathbf{C}$
  - $D :=$ Laplacian of the complete graph where only edges $\{i, j\}$ with $i \in S$ and $j \in T$ have weight f_{ij} and rest have 0 weight.
  - $\mathbf{D} \bullet \mathbf{X} = \sum_{ij} f_{ij}\|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq 2\alpha$ (Using Lemma 2).

Undirected BALANCED SEPARATOR : Oracle

ORACLE Description

Undirected BALANCED SEPARATOR : Oracle

ORACLE Description

- Set all $x_i = \frac{\alpha}{n}$, and all $z_S = 0$.

Undirected BALANCED SEPARATOR : Oracle

ORACLE Description

- Set all $x_i = \frac{\alpha}{n}$, and all $z_S = 0$.
- $\sum_p f_p \mathbf{T}_p = \mathbf{F} - \mathbf{D}$

Undirected BALANCED SEPARATOR : Oracle

ORACLE Description

- Set all $x_i = \frac{\alpha}{n}$, and all $z_S = 0$.
- $\sum_p f_p \mathbf{T}_p = \mathbf{F} - \mathbf{D}$
- Thus the feedback matrix becomes

$$\mathrm{diag}(\mathbf{x}) + \mathbf{F} - \mathbf{D} - \mathbf{F} = \mathrm{diag}(\mathbf{x}) - \mathbf{D}$$

Then $(\frac{\alpha}{n}\mathbf{I} - \mathbf{D}) \bullet \mathbf{X} \leq \alpha - \alpha = 0$.
Also, since the flow is d-regular, $\mathbf{0} \preceq \mathbf{D} \preceq 2d\mathbf{I}$. Hence,
$-2d\mathbf{I} \preceq \frac{\alpha}{n}\mathbf{I} - \mathbf{D} \preceq \frac{\alpha}{n}\mathbf{I}$

Undirected BALANCED SEPARATOR : Time Complexity Analysis

- Assume that graph is preprocessed using algorithm of Benczúr and Karger
- $\rho = O(\frac{\log(n)\alpha}{n})$ and $R = n$. Thus the number of iterations from Theorem 1 is $O(\log^3(n))$.
- Each iteration involves at most one max-flow computation which can be done by Goldberg and Rao's algorithm in $\tilde{O}(n^{1.5})$ time since there are $O(n)$ edges.
- We also compute, in each iteration, an approximation of Cholesky decomposition of the matrix exponential by projecting on a random $O(\log n)$ dimensional subspace. Since there are only $O(\log^3(n))$ iterations and each iteration adds at most $\tilde{O}(n^{1.5})$ demand pairs in the max-flow computation, the matrix exponential has only $\tilde{O}(n^{1.5})$ non-zero entries and can be computed in $\tilde{O}(n^{1.5})$ time.
- Thus running time is $\tilde{O}(m + n^{1.5})$

Outline

Undirected SPARSEST CUT

Undirected Sparsest Cut Problem

Given a graph $G(V, E)$ with $|V| = n, |E| = m$, and capacity c_e on edge $e \in E$, find the cut $(S, \overline{S})$ with minimum expansion, $\frac{E(S,\overline{S})}{min\{|S|,|\overline{S}|\}}$

Undirected SPARSEST CUT

Undirected Sparsest Cut Problem

Given a graph $G(V,E)$ with $|V| = n, |E| = m$, and capacity c_e on edge $e \in E$, find the cut $(S, \overline{S})$ with minimum expansion, $\frac{E(S,\overline{S})}{min\{|S|,|\overline{S}|\}}$

Theorem

An $O(\log n)$ pseudo-approximation to the SPARSEST CUT can be computed in $\tilde{O}(m + n^{1.5})$ time using $O(\log^2(n))$ single commodity flow computations.

Undirected SPARSEST CUT

SDP

$$\min \sum_{e=\{i,j\} \in E} c_e \|\mathbf{v}_i - \mathbf{v}_j\|^2$$

$$\forall p : \sum_{j=1}^{k-1} \|\mathbf{v}_{i_j} - \mathbf{v}_{i_{j+1}}\|^2 \geq \|\mathbf{v}_{i_1} - \mathbf{v}_{i_k}\|^2$$

$$\|\sum_i \mathbf{v}_i\|^2 = 0$$

$$\sum_i \|\mathbf{v}_i\|^2 = n$$

$$\min \mathbf{C} \bullet \mathbf{X}$$

$$\forall p : \mathbf{T}_p \bullet \mathbf{X} \geq 0$$

$$\mathbf{J} \bullet \mathbf{X} = 0$$

$$\mathbf{Tr}(\mathbf{X}) = n$$

$$X \succeq 0$$

J is the all ones matrix.

Undirected SPARSEST CUT

SDP

$$\min \mathbf{C} \bullet \mathbf{X}$$
$$\forall p : \mathbf{T}_p \bullet \mathbf{X} \geq 0$$
$$\mathbf{J} \bullet \mathbf{X} = 0$$
$$\mathbf{Tr}(\mathbf{X}) = n$$
$$X \succeq 0$$

J is the all ones matrix.

Dual Program

$$\max nx$$
$$x\mathbf{I} + \sum_p f_p \mathbf{T}_p + z\mathbf{J} \preceq C$$
$$\forall p : f_p \geq 0$$

Undirected SPARSEST CUT : Oracle

Lemma 3

Given for all $i \in V$, vectors $\mathbf{v}_i$, such that for some constant δ_1, $n^2 \geq \sum_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq (1-\delta_1)n^2$, and a quantity α. Then there is an algorithm, which, using a single max-flow computation, outputs either,

Undirected SPARSEST CUT : Oracle

Lemma 3

Given for all $i \in V$, vectors $\mathbf{v}_i$, such that for some constant δ_1, $n^2 \geq \sum_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq (1-\delta_1)n^2$, and a quantity α. Then there is an algorithm, which, using a single max-flow computation, outputs either,

1. a valid $O(\frac{\alpha}{n})$-regular flow f_p, such that $\sum_{ij} f_{ij}\|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, or,

Undirected SPARSEST CUT : Oracle

Lemma 3

Given for all $i \in V$, vectors $\mathbf{v}_i$, such that for some constant δ_1, $n^2 \geq \sum_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq (1-\delta_1)n^2$, and a quantity α. Then there is an algorithm, which, using a single max-flow computation, outputs either,

1. a valid $O(\frac{\alpha}{n})$-regular flow f_p, such that $\sum_{ij} f_{ij}\|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, or,
2. a cut of expansion $O(\frac{\alpha}{n})$, or,

Undirected SPARSEST CUT : Oracle

Lemma 3

Given for all $i \in V$, vectors $\mathbf{v}_i$, such that for some constant δ_1, $n^2 \geq \sum_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq (1-\delta_1)n^2$, and a quantity α. Then there is an algorithm, which, using a single max-flow computation, outputs either,

1. a valid $O(\frac{\alpha}{n})$-regular flow f_p, such that $\sum_{ij} f_{ij}\|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, or,
2. a cut of expansion $O(\frac{\alpha}{n})$, or,
3. a set of nodes $S \subseteq V$ of size $\Omega(n)$, such that for all $i \in S$, $\|\mathbf{v}_i\|^2 = O(1)$, $\sum_{i,j \in S} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \Omega(n^2)$

Undirected SPARSEST CUT : Oracle

ORACLE Description

Given a candidate solution $\mathbf{X}$, the oracle always sets $x = \frac{\alpha}{n}$. Since $x\mathbf{I} \bullet \mathbf{X} = \alpha$, it now needs to find f_p, z and $\mathbf{F} \preceq \mathbf{C}$ such that

$$\alpha + \sum_p f_p(\mathbf{T}_p \bullet \mathbf{X}) + z(\mathbf{J} \bullet \mathbf{X}) - (\mathbf{F} \bullet \mathbf{X}) \leq 0$$

It runs the following steps:

Undirected SPARSEST CUT : Oracle

ORACLE Description

Given a candidate solution $\mathbf{X}$, the oracle always sets $x = \frac{\alpha}{n}$. Since $x\mathbf{I} \bullet \mathbf{X} = \alpha$, it now needs to find f_p, z and $\mathbf{F} \preceq \mathbf{C}$ such that

$$\alpha + \sum_p f_p(\mathbf{T}_p \bullet \mathbf{X}) + z(\mathbf{J} \bullet \mathbf{X}) - (\mathbf{F} \bullet \mathbf{X}) \leq 0$$

It runs the following steps:

1. If $\mathbf{J} \bullet \mathbf{X} \geq \delta_1 n^2$, for some small constant δ_1, then set $z = -\frac{\alpha}{\delta_1 n^2}$, so that $z(\mathbf{J} \bullet \mathbf{X}) \leq -\alpha$. Also, $\|\frac{\alpha}{n}\mathbf{I} - z\mathbf{J}\| \leq O(\frac{\alpha}{n})$

Undirected SPARSEST CUT : Oracle

ORACLE Description

Given a candidate solution $\mathbf{X}$, the oracle always sets $x = \frac{\alpha}{n}$. Since $x\mathbf{I} \bullet \mathbf{X} = \alpha$, it now needs to find f_p, z and $\mathbf{F} \preceq \mathbf{C}$ such that

$$\alpha + \sum_p f_p(\mathbf{T}_p \bullet \mathbf{X}) + z(\mathbf{J} \bullet \mathbf{X}) - (\mathbf{F} \bullet \mathbf{X}) \leq 0$$

It runs the following steps:

1. If $\mathbf{J} \bullet \mathbf{X} \geq \delta_1 n^2$, for some small constant δ_1, then set $z = -\frac{\alpha}{\delta_1 n^2}$, so that $z(\mathbf{J} \bullet \mathbf{X}) \leq -\alpha$. Also, $\|\frac{\alpha}{n}\mathbf{I} - z\mathbf{J}\| \leq O(\frac{\alpha}{n})$
2. Assume $\mathbf{J} \bullet \mathbf{X} \leq \delta_1 n^2$ and $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ be the vectors obtained from the Cholesky decomposition of $\mathbf{X}$.
   $\mathbf{J} \bullet \mathbf{X} \implies n^2 \geq \sum_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq (1 - \delta_1)n^2$. Apply lemma 3.

Undirected SPARSEST CUT : Oracle

ORACLE Description

Undirected SPARSEST CUT : Oracle

ORACLE Description

- If from the previous step, a cut of expansion $O(\frac{\alpha}{n})$ is obtained, output it.

Undirected SPARSEST CUT : Oracle

ORACLE Description

- If from the previous step, a cut of expansion $O(\frac{\alpha}{n})$ is obtained, output it.
- If we get a flow f_p such that $\sum_{ij} f_{ij}\|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, define $\mathbf{F}$ and $\mathbf{D}$ to be the flow and demand graph Laplacians respectively and proceed as in step 3 of undirected BALANCED SEPARATOR.

Undirected SPARSEST CUT : Oracle

ORACLE Description

- If from the previous step, a cut of expansion $O(\frac{\alpha}{n})$ is obtained, output it.
- If we get a flow f_p such that $\sum_{ij} f_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, define $\mathbf{F}$ and $\mathbf{D}$ to be the flow and demand graph Laplacians respectively and proceed as in step 3 of undirected BALANCED SEPARATOR.
- Finally if a set of nodes $S \subseteq V$ of size $\Omega(n)$ is obtained, such that for all $i \in S$, $\|\mathbf{v}_i\|^2 = O(1)$ and $\sum_{i,j \in S} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \Omega(n^2)$, apply lemma 1 to S.

Undirected SPARSEST CUT : Oracle

ORACLE Description

- If from the previous step, a cut of expansion $O(\frac{\alpha}{n})$ is obtained, output it.
- If we get a flow f_p such that $\sum_{ij} f_{ij}\|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, define $\mathbf{F}$ and $\mathbf{D}$ to be the flow and demand graph Laplacians respectively and proceed as in step 3 of undirected BALANCED SEPARATOR.
- Finally if a set of nodes $S \subseteq V$ of size $\Omega(n)$ is obtained, such that for all $i \in S$, $\|\mathbf{v}_i\|^2 = O(1)$ and $\sum_{i,j\in S} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \Omega(n^2)$, apply lemma 1 to S.
  - If a cut of small expansion is obtained, stop.

Undirected SPARSEST CUT : Oracle

ORACLE Description

- If from the previous step, a cut of expansion $O(\frac{\alpha}{n})$ is obtained, output it.
- If we get a flow f_p such that $\sum_{ij} f_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, define $\mathbf{F}$ and $\mathbf{D}$ to be the flow and demand graph Laplacians respectively and proceed as in step 3 of undirected BALANCED SEPARATOR.
- Finally if a set of nodes $S \subseteq V$ of size $\Omega(n)$ is obtained, such that for all $i \in S$, $\|\mathbf{v}_i\|^2 = O(1)$ and $\sum_{i,j \in S} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \Omega(n^2)$, apply lemma 1 to S.
  - If a cut of small expansion is obtained, stop.
  - Else if a d-regular flow such that $\sum_{ij} f_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$ is obtained, proceed as before.

Outline

References

1. Arora, Sanjeev, and Satyen Kale. "A combinatorial, primal-dual approach to semidefinite programs." *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*. ACM, 2007.
2. Vazirani, Vijay V. *Approximation algorithms*. springer, 2001.
3. Arora, Sanjeev, Elad Hazan, and Satyen Kale. "The Multiplicative Weights Update Method: a Meta-Algorithm and Applications." *Theory of Computing* 8.1 (2012): 121-164.
4. Arora, Sanjeev, Elad Hazan, and Satyen Kale. "Fast algorithms for approximate semidefinite programming using the multiplicative weights update method." *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*. IEEE, 2005.

Questions???

Outline

Undirected BALANCED SEPARATOR : Oracle

Lemma 1

For at least a $\frac{a}{32}$ fraction of directions $\mathbf{u}$, there are efficiently computable sets S and T, each of size at least $\frac{a}{128n}$, such that for any $i \in S$ and $j \in T$, $((v)_j - \mathbf{v}_i) \cdot \mathbf{u} \geq \frac{a}{48\sqrt{n}}$

Undirected BALANCED SEPARATOR : Oracle

Lemma 1: Proof

- Since $\sum_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq an^2$ and $\|\mathbf{v}_i - \mathbf{v}_j\| \leq 2$, $\sum_{ij} \|\mathbf{v}_i - \mathbf{v}_j\| \geq \frac{a}{2} n^2$.
- Thus for any node i,

$$\frac{a}{2} n^2 \leq \sum_{jk} \|\mathbf{v}_j - \mathbf{v}_k\| \leq \sum_{jk} \|\mathbf{v}_j - \mathbf{v}_i\| + \|\mathbf{v}_i - \mathbf{v}_k\| \leq n \sum_{j} \|\mathbf{v}_i - \mathbf{v}_j\|$$

So,

$$\sum_{j} \|\mathbf{v}_i - \mathbf{v}_j\| \geq \frac{a}{2} n$$

Lemma 1: Proof

- Since the maximum value of $\|\mathbf{v}_i - \mathbf{v}_j\|$ is 2, there must be at least $\frac{a}{8}n$ nodes i such that $\|\mathbf{v}_i - \mathbf{v}_j\| \geq \frac{a}{4}$.
- Define a *stretched pair* as a pair of nodes i, j, such that $|\mathbf{v}_i \cdot \mathbf{u} - \mathbf{v}_j \cdot \mathbf{u}| \geq \frac{a}{24\sqrt{n}}$. The Gaussian nature of projections guarantees that this occurs with probability $\frac{1}{2}$. Thus,

$$E[\#\text{stretched pairs}] \geq \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{a}{8}n \cdot n = \frac{a}{32}n^2$$

- Since there are atmost $\frac{1}{2}n^2$ pairs, for at least $\frac{a}{32}$ fraction of directions $\mathbf{u}$, we have $\frac{a}{64}n^2$ stretched pairs.

Undirected BALANCED SEPARATOR : Oracle

Lemma 1: Proof

- Let $\mathbf{u}$ be such a direction. Define $\delta = \frac{a}{48\sqrt{n}}$ and m to be the median value of $\mathbf{v}_i \cdot \mathbf{u}$
- Define sets $L = \{i : \mathbf{v}_i \cdot \mathbf{u} \leq m - \delta\}$,
  $M^- = \{i : \mathbf{v}_i \cdot \mathbf{u} \in [m - \delta, m]\}$,
  $M^+ = \{i : \mathbf{v}_i \cdot \mathbf{u} \in [m, m + \delta]\}$, $R = \{i : \mathbf{v}_i \cdot \mathbf{u} \geq m + \delta\}$.
  Thus, any stretched pair has at least one node in $L \cup R$.
- At least one of L or R has size at least $\frac{a}{128}n$, as otherwise the number of stretched pairs is less than $2 \cdot \frac{a}{128}n \cdot n = \frac{a}{64}n$ (contradiction).
- If $|L| \geq \frac{a}{128}n$, set $S = L$, $T = M^+ \cup R$.
- $|T| \geq \frac{n}{2}$ as T is the set of all points with projection higher than median.

Undirected BALANCED SEPARATOR : Oracle

Lemma 2

Let $S \subseteq V$ be a set of nodes of size $\Omega(n)$. Suppose for all $i \in S$, vectors $\mathbf{v}_i$ of length $O(1)$ are given such that $\sum_{i,j \in S} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \Omega(n^2)$, and a quantity α. Then there is an algorithm, which, using a single max-flow computation, either outputs a valid $O(\frac{\log(n)\alpha}{n})$-regular flow f_p such that $\sum_{ij} f_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, or a c'-balanced cut of expansion $O(\log(n)\frac{\alpha}{n})$.

Undirected BALANCED SEPARATOR : Oracle

Lemma 2 : Proof

- What we seek: A d-regular flow f_p for $d := \frac{\beta \log(n) \cdot \alpha}{n}$ where β is a sufficiently large constant.
- Choose a direction represented by a unit vector $\mathbf{u}$ at random.
- Since $\mathbf{K}_S \bullet \mathbf{X} \geq \Omega n^2$, thus $\sum_{i,j \in S} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \Omega n^2$
- Using Lemma 1, we can find sets S and T of size cn each, for some constant $c > 0$, such that for all $i \in S$ and $j \in T$, we have $(\mathbf{v}_j - \mathbf{v}_i) \cdot \mathbf{u} \geq \frac{\sigma}{\sqrt{n}}$ for some constant $\sigma > 0$
- Using Gaussian nature of projections, with very high probability, for any pair of nodes i, j we have $|(\mathbf{v}_j - \mathbf{v}_i) \cdot \mathbf{u}| \leq O(\sqrt{\log(n)}) \cdot \frac{\|\mathbf{v}_i - \mathbf{v}_j\|}{\sqrt{n}}$

Undirected BALANCED SEPARATOR : Oracle

Lemma 2 : Proof

- Thus with constant probability, we get sets S and T such that for all nodes $i \in S$ and $j \in T$, we have $\|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \frac{\gamma}{\log(n)}$ for some constant $\gamma > 0$
- If this is the case, connect all nodes in S to a single source and all nodes in T to a single sink with edges of capacity d each. Let f_p be the max flow in this network. Suppose the total flow obtained is at least $\frac{c\beta}{2}\log(n)\cdot\alpha$. Assume that all flow originates from a node $i \in S$ and ends at some node $j \in T$. Then,

$$\sum_{i \in S, j \in T} f_{ij}\|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \frac{c\beta}{2}\log(n)\cdot\alpha \times \frac{\gamma}{\log(n)} = 2\alpha$$

if $\beta = \frac{4}{c\gamma}$

Undirected BALANCED SEPARATOR : Oracle

Lemma 2 : Proof

- If the total flow obtained is less than $\frac{c\beta}{2}\log(n)\cdot\alpha$, by max-flow-min-cut theorem, the cut obtained is also at most this size. This is $c/2$-balanced, since atmost $\frac{c\beta}{2}\log(n)\cdot\alpha/d = cn/2$ source (and sink) edges can be cut. Thus cut expansion is $O(\log(n)\cdot\frac{\alpha}{n})$

Undirected SPARSEST CUT : Oracle

Lemma 3

Given for all $i \in V$, vectors $\mathbf{v}_i$, such that for some constant δ_1, $n^2 \geq \sum_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq (1-\delta_1)n^2$, and a quantity α. Then there is an algorithm, which, using a single max-flow computation, outputs either,

Undirected SPARSEST CUT : Oracle

Lemma 3

Given for all $i \in V$, vectors $\mathbf{v}_i$, such that for some constant δ_1, $n^2 \geq \sum_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq (1 - \delta_1)n^2$, and a quantity α. Then there is an algorithm, which, using a single max-flow computation, outputs either,

1. a valid $O(\frac{\alpha}{n})$-regular flow f_p, such that $\sum_{ij} f_{ij}\|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, or,

Undirected SPARSEST CUT : Oracle

Lemma 3

Given for all $i \in V$, vectors $\mathbf{v}_i$, such that for some constant δ_1, $n^2 \geq \sum_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq (1-\delta_1)n^2$, and a quantity α. Then there is an algorithm, which, using a single max-flow computation, outputs either,

1. a valid $O(\frac{\alpha}{n})$-regular flow f_p, such that $\sum_{ij} f_{ij}\|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, or,
2. a cut of expansion $O(\frac{\alpha}{n})$, or,

Undirected SPARSEST CUT : Oracle

Lemma 3

Given for all $i \in V$, vectors $\mathbf{v}_i$, such that for some constant δ_1, $n^2 \geq \sum_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq (1 - \delta_1)n^2$, and a quantity α. Then there is an algorithm, which, using a single max-flow computation, outputs either,

1. a valid $O(\frac{\alpha}{n})$-regular flow f_p, such that $\sum_{ij} f_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \alpha$, or,
2. a cut of expansion $O(\frac{\alpha}{n})$, or,
3. a set of nodes $S \subseteq V$ of size $\Omega(n)$, such that for all $i \in S$, $\|\mathbf{v}_i\|^2 = O(1)$, $\sum_{i,j \in S} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \Omega(n^2)$

Undirected SPARSEST CUT : Oracle

Lemma 3 : Proof

Given vectors $\mathbf{v}_i$ such that $n^2 \geq \sum_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq (1-\delta_1)n^2$, run the following steps,

1. For a node i and radius r, let $B(i,r) = \{j : \|\mathbf{v}_i - \mathbf{v}_j\| \leq r\}$. If there is a node i such that for some constant δ_2, $|B(i,\delta_2)| \geq n/4$, then any $i_0 \in B(i,\delta_2)$ satisfies $|B(i_0, 2\delta_2)| \geq n/4$. Find such i_0 by random sampling. Define $L = B(i_0, 2\delta_2)$, and $R = V \backslash L$. For $j \in R$, define $d(j,L) = \min_{i \in L} \|\mathbf{v}_i - \mathbf{v}_j\|^2$. Then, since $\sum_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq (1-\delta_1)n^2$, $\sum_{j \in R} d(j,L) \geq \frac{n}{10}$ for suitable choice of δ_1, δ_2. Define $k := \frac{|R|}{|L|}$. $k = O(1)$

Undirected SPARSEST CUT : Oracle

Lemma 3 : Proof

Given vectors $\mathbf{v}_i$ such that $n^2 \geq \sum_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq (1 - \delta_1)n^2$, run the following steps,

1. ...

   Connect all nodes in L to a single source with edges of capacity $\frac{10k\alpha}{n}$ and all nodes in R to a single sink with edges of capacity $\frac{10\alpha}{n}$ and compute the max-flow. If the flow saturates all source and sink nodes, then

$$\sum_{i \in L, j \in R} f_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \sum_{j \in R} \frac{10\alpha}{n} \cdot d(j, L) \geq \alpha$$

Undirected SPARSEST CUT : Oracle

Lemma 3 : Proof

Given vectors $\mathbf{v}_i$ such that $n^2 \geq \sum_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq (1-\delta_1)n^2$, run the following steps,

2. If the flow doesn't saturate all source and sink edges, then let the number of nodes in L in the resulting cut connected to source be n_s and the number of nodes in R connected to the sink be n_t. Then the capacity of the graph edges cut is at most $\frac{10\alpha}{n}(|R| - kn_s - n_t)$, and the smaller side of the cut has at least $\min\{|L| - n_s, |R| - n_t\}$ nodes. Thus, expansion of cut is at most $\frac{10k\alpha}{n} = O(\frac{\alpha}{n})$.

Undirected SPARSEST CUT : Oracle

Lemma 3 : Proof

Given vectors $\mathbf{v}_i$ such that $n^2 \geq \sum_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq (1 - \delta_1)n^2$, run the following steps,

3. For all nodes i, let $|B(i, \delta_2)| < n/4$. Then it can be easily checked that there is a node i such that $|B(i, \sqrt{2})| \geq \frac{(1-\delta_1)}{2} n$. Find i_0, by random sampling, such that $|B(i, 2\sqrt{2})| \geq \frac{(1-\delta_1)}{2} n$. Let $S = B(i, 2\sqrt{2})$. Since for every $i \in S$, $|B(i, \delta_2)| < n/4$, $\sum_{i,j \in S} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \Omega(n^2)$. Output S.

Matrix Exponentiation

Complexity of computing exponential of a matrix

- No fast algorithm known. Still an area of active research.
- Special cases:
  1. Exponential of a diagonal matrix: A diagonal matrix whose diaogonal elements are exponential of diagonal elements of original matrix.
  2. Projection Matrix ($\mathbf{P}^2 = \mathbf{P}$): $exp(\mathbf{P}) = \mathbf{I} + \mathbf{P}(1 + \frac{1}{2!} + \cdots) = \mathbf{I} + (e-1)\mathbf{P}$
  3. Nilpotent Matrix ($\mathbf{P}^q = \mathbf{0}$): $exp(\mathbf{P}) = \mathbf{I} + \mathbf{P} + \frac{\mathbf{P^2}}{2!} + \cdots + \frac{\mathbf{P^{q-1}}}{(q-1)!}$
- Other techniques include using Laurent Series, Sylvester's Formula, etc.
- If $\mathbf{Y}$ is invertible, then $e^{\mathbf{YXY}^{-1}} = \mathbf{Y}e^{\mathbf{X}}\mathbf{Y}^{-1}$. This gives a $O(n^3)$ algorithm for matrix exponentiation.

Matrix Exponentiation

- Idea: only approximate computation suffices.
- ORACLE finds $\mathbf{y}$ such that $\sum_{j=1}^{m}(\mathbf{A}_j \bullet \mathbf{X}^{(t)})y_j - (\mathbf{C} \bullet \mathbf{X}^{(t)}) \geq 0$.
- Let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ be vectors obtained from Cholesky decomposition of $\mathbf{X}^{(t)}$ such that $\mathbf{X}_{ij}^{(t)} = \mathbf{v}_i \cdot \mathbf{v}_j = \frac{1}{2}[\|\mathbf{v}_i\|^2 + \|\mathbf{v}_j\|^2 - \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq 0$.
- ORACLE needs to find appropiate variables s_i and t_{ij} such that $\Sigma_i s_i \|\mathbf{v}_i\|^2 + \Sigma_{ij} t_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq 0$.
- Vectors $\mathbf{v}_i$ obtained from Cholesky decomposition of $\mathbf{X}^{(t)} = exp(\mathbf{M})$ are simply the row vectors of $exp(\frac{1}{2}\mathbf{M})$.
- Since we are only interested in norms, we can try Johnson-Lindenstrauss dimension reduction.

Matrix Exponentiation

Johnson-Lindenstrauss Lemma

Given $0 < \epsilon < 1$, a set X of m points in $\mathbb{R}^n$ and a number $n > 8 \ln n/\epsilon^2$, there is a linear map $\mathbb{R}^N \to \mathbb{R}^n$ such that

$$(1-\epsilon)\|\mathbf{u}-\mathbf{v}\|^2 \leq \|f(\mathbf{u}) - f(\mathbf{v})\|^2 \leq (1+\epsilon)\|\mathbf{u}-\mathbf{v}\|^2$$

for all $\mathbf{u}, \mathbf{v} \in \mathbf{X}$.

- $\mathbf{v}_i$ are the vectors obtained from Cholesky Decomposition of $\mathbf{X}^{(t)}$.
- Project the vectors $\mathbf{v}_i$ on a random $d = O(\frac{\log n}{\delta^2})$ dimensional subspace, and scale the projections by $\sqrt{\frac{n}{d}}$ to get vectors $\mathbf{v}'$.
- With high probability, $\|\mathbf{v}'_i\|^2$ and $\|\mathbf{v}'_i - \mathbf{v}'_j\|^2$ are within $(1 \pm \delta)$ of $\|\mathbf{v}_i\|^2$ and $\|\mathbf{v}_i - \mathbf{v}_j\|^2$.
- Run ORACLE for $\mathbf{X}'$ in a way that its feedback is also valid for $\mathbf{X}^{(t)}$.