Today

Semidefinite Programming
Semidefinite Programming
...for Approximating MaxCut.
Positive Semidefinite Matrices

$n \times n$ matrix $A$ positive semidefinite $\rightarrow$ for all $x \in \mathbb{R}^n$, $x^T Ax \geq 0$. 

- Spectral decomposition: $A = \sum_i \lambda_i \nu_i \nu_i^T$.
  - $\lambda_i \geq 0$ – eigenvalue.
  - $\nu_i$ – associated eigenvector.

$A = B^T B$.

- Representation $\rightarrow$ positive semidefinite too: $(x^T B^T B) (Bx) \geq 0$

- Possibly many such representations.

- Efficiently computable: Cholesky factorization.
Positive Semidefinite Matrices

An $n \times n$ matrix $A$ is positive semidefinite if for all $x \in \mathbb{R}^n$, $x^T A x \geq 0$.

Hmm...
Positive Semidefinite Matrices

An $n \times n$ matrix $A$ is positive semidefinite if for all $x \in \mathbb{R}^n$, $x^T A x \geq 0$.

Hmm...

Ax is same direction as $x$?
Positive Semidefinite Matrices

$n \times n$ matrix $A$ positive semidefinite $\rightarrow$ for all $x \in \mathbb{R}^n$, $x^T Ax \geq 0$.

Hmm...

$Ax$ is same direction as $x$?

$Ax$, $x$
Positive Semidefinite Matrices

\( n \times n \) matrix \( A \) positive semidefinite \( \rightarrow \) for all \( x \in \mathbb{R}^n \), \( x^T A x \geq 0 \).

Hmm...

Ax is same direction as \( x \)?

Spectral decomposition:
\[
A = \sum_i \lambda_i v_i v_i^T.
\]
Positive Semidefinite Matrices

An $n \times n$ matrix $A$ is positive semidefinite if for all $x \in \mathbb{R}^n$, $x^T A x \geq 0$.

Hmm...

- $Ax$ is same direction as $x$?

Spectral decomposition:

$$A = \sum_i \lambda_i v_i v_i^T.$$ 

- $\lambda_i \geq 0$ – eigenvalue.
- $v_i$ – associated eigenvector.
Positive Semidefinite Matrices

An $n \times n$ matrix $A$ is positive semidefinite if for all $x \in \mathbb{R}^n$, $x^T A x \geq 0$.

Hmm...

Ax is same direction as x?

Spectral decomposition:

$A = \sum_i \lambda_i v_i v_i^T$.

$\lambda_i \geq 0$ – eigenvalue.

$v_i$ – associated eigenvector.

$A = B^T B$
Positive Semidefinite Matrices

\[ n \times n \text{ matrix } A \text{ positive semidefinite } \rightarrow \text{ for all } x \in R^n, \ x^T A x \geq 0. \]

Hmm...

Ax is same direction as x?

![Diagram of vector Ax in the same direction as x](image)

Spectral decomposition:

\[ A = \sum_i \lambda_i v_i v_i^T. \]

\( \lambda_i \geq 0 \) – eigenvalue.

\( v_i \) – associated eigenvector.

\[ A = B^T B \quad B = \sqrt{\lambda_i} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \end{bmatrix} \]
Positive Semidefinite Matrices

An $n \times n$ matrix $A$ is positive semidefinite if for all $x \in \mathbb{R}^n$, $x^T Ax \geq 0$.

Hmm...

Ax is same direction as x?

Spectral decomposition:
\[ A = \sum_i \lambda_i v_i v_i^T. \]

$\lambda_i \geq 0$ – eigenvalue.

$v_i$ – associated eigenvector.

\[ A = B^T B \quad B = \sqrt{\lambda_i} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \end{bmatrix} \]

Representation $\rightarrow$ positive semidefinite too: $(x^T B^T)(Bx) \geq 0$
Positive Semidefinite Matrices

An $n \times n$ matrix $A$ is positive semidefinite $\rightarrow$ for all $x \in \mathbb{R}^n$, $x^T A x \geq 0$.

Hmm...

$A x$ is same direction as $x$?

Spectral decomposition:

$A = \sum \lambda_i v_i v_i^T$.

$\lambda_i \geq 0$ – eigenvalue.
$v_i$ – associated eigenvector.

$A = B^T B \quad B = \sqrt{\lambda_i} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \end{bmatrix}$

Representation $\rightarrow$ positive semidefinite too: $(x^T B^T)(Bx) \geq 0$
Possibly many such representations.
Positive Semidefinite Matrices

$n \times n$ matrix $A$ positive semidefinite $\rightarrow$ for all $x \in \mathbb{R}^n$, $x^T A x \geq 0$.

Hmm...

$Ax$ is same direction as $x$?

Spectral decomposition:

$A = \sum_i \lambda_i v_i v_i^T$.

$\lambda_i \geq 0$ – eigenvalue.

$v_i$ – associated eigenvector.

$A = B^T B$ \hspace{0.5cm} $B = \sqrt{\lambda_i} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \end{bmatrix}$

Representation $\rightarrow$ positive semidefinite too: $(x^T B^T)(Bx) \geq 0$

Possibly many such representations.

Efficiently computable: cholesky factorization.
Semidefinite Programming.

\[
\begin{align*}
\text{max} & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*}
\] (1)
Semidefinite Programming.

\[
\begin{align*}
\text{max} & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*}
\]

(1)

\(A.X\) is matrix inner product:
Semidefinite Programming.

\[ \begin{align*}
\text{max} & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*} \] (1)

\(A.X\) is matrix inner product: \(\sum_{ij} a_{ij}x_{ij}\).
Semidefinite Programming.

$$\begin{align*}
\text{max} \quad & A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*} \tag{1}$$

$A.X$ is matrix inner product: $\sum_{ij} a_{ij} x_{ij}$.

view $A$ and $X$ as $n^2$ dimensional vector.
Semidefinite Programming.

\[ \begin{align*}
\text{max} & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*} \] (1)

A.X is matrix inner product: \( \sum_{ij} a_{ij}x_{ij} \).

view A and X as \( n^2 \) dimensional vector.

Linear Programming?
Semidefinite Programming.

\[
\begin{align*}
\text{max} & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0 \quad (1)
\end{align*}
\]

\(A.X\) is matrix inner product: \(\sum_{ij} a_{ij}x_{ij}\).
view \(A\) and \(X\) as \(n^2\) dimensional vector.

Linear Programming? \(A\) must be diagonal.
Semidefinite Programming.

\[
\begin{align*}
\max & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*}
\]  \hspace{1cm} (1)

\(A.X\) is matrix inner product: \(\sum_{ij} a_{ij}x_{ij}\).

view \(A\) and \(X\) as \(n^2\) dimensional vector.

Linear Programming? \(A\) must be diagonal.

Constraint for each \(i \neq j\),
Semidefinite Programming.

\[
\begin{align*}
\text{max} & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*}
\]

(1)

\(A.X\) is matrix inner product: \(\sum_{ij} a_{ij}x_{ij}\).

view \(A\) and \(X\) as \(n^2\) dimensional vector.

Linear Programming? \(A\) must be diagonal.

Constraint for each \(i \neq j\),

\(X_{jk}\) is 1 at entry \(jk\), 0 elsewhere.
Semidefinite Programming.

\[
\begin{align*}
\text{max} & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0 
\end{align*}
\]  \hfill (1)

\(A.X\) is matrix inner product: \(\sum_{ij} a_{ij}x_{ij}\).

view \(A\) and \(X\) as \(n^2\) dimensional vector.

Linear Programming? \(A\) must be diagonal.

Constraint for each \(i \neq j\),
\(X_{jk}\) is 1 at entry \(jk\), 0 elsewhere. \(b_{jk}\) is 0.
Semidefinite Programming.

\[
\begin{align*}
\text{max} & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*}
\]

(1)

$A.X$ is matrix inner product: $\sum_{ij} a_{ij} x_{ij}$.

- view $A$ and $X$ as $n^2$ dimensional vector.

Linear Programming? $A$ must be diagonal.

- Constraint for each $i \neq j$,
  $X_{jk}$ is 1 at entry $jk$, 0 elsewhere. $b_{jk}$ is 0.

Solvable?
Semidefinite Programming.

\[
\begin{align*}
\text{max} & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*}
\] (1)

$A.X$ is matrix inner product: $\sum_{ij} a_{ij}x_{ij}$.
view $A$ and $X$ as $n^2$ dimensional vector.

Linear Programming? $A$ must be diagonal.
Constraint for each $i \neq j$,
$X_{jk}$ is 1 at entry $jk$, 0 elsewhere. $b_{jk}$ is 0.

Solvable?
Convex: Solution $A$ and $A'$.
Semidefinite Programming.

\[
\begin{align*}
\text{max} & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*}
\]

(1)

\(A.X\) is matrix inner product: \(\sum_{ij} a_{ij}x_{ij}\).
view \(A\) and \(X\) as \(n^2\) dimensional vector.

Linear Programming? \(A\) must be diagonal.
Constraint for each \(i \neq j\),
\(X_{jk}\) is 1 at entry \(jk\), 0 elsewhere. \(b_{jk}\) is 0.

Solvable?
Convex: Solution \(A\) and \(A'\).
\(\mu A + (1 - \mu)A'\) is solution.
Semidefinite Programming.

\[
\begin{align*}
\text{max} \quad & A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*}
\]

(1)

\(A.X\) is matrix inner product: \(\sum_{ij} a_{ij} x_{ij}\).

view \(A\) and \(X\) as \(n^2\) dimensional vector.

Linear Programming? \(A\) must be diagonal.

Constraint for each \(i \neq j\),

\(X_{jk}\) is 1 at entry \(jk\), 0 elsewhere. \(b_{jk}\) is 0.

Solvable?

Convex: Solution \(A\) and \(A'\).

\(\mu A + (1 - \mu)A'\) is solution.

Linear constraints, objective.
Semidefinite Programming.

\[
\begin{align*}
\text{max} & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*}
\]  

(1)

\(A.X\) is matrix inner product: \(\sum_{ij} a_{ij}x_{ij}\).

view \(A\) and \(X\) as \(n^2\) dimensional vector.

Linear Programming? \(A\) must be diagonal.
Constraint for each \(i \neq j\),
\(X_{jk}\) is 1 at entry \(jk\), 0 elsewhere. \(b_{jk}\) is 0.

Solvable?
Convex: Solution \(A\) and \(A'\).
\(\mu A + (1 - \mu)A'\) is solution.
Linear constraints, objective.
\(x^T A x, x^T A' x \geq 0 \implies x^T (\mu A + (1 - \mu)A') x \geq 0.\)
Semidefinite Programming.

\[
\begin{align*}
\text{max} & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*}
\] (1)

\(A.X\) is matrix inner product: \(\sum_{ij} a_{ij}x_{ij}\).

view \(A\) and \(X\) as \(n^2\) dimensional vector.

Linear Programming? \(A\) must be diagonal.

Constraint for each \(i \neq j\),

\(X_{jk}\) is 1 at entry \(jk\), 0 elsewhere. \(b_{jk}\) is 0.

Solvable?

Convex: Solution \(A\) and \(A'\).

\(\mu A + (1 - \mu)A'\) is solution.

Linear constraints, objective.

\(x^T Ax, x^T A' x \geq 0 \implies x^T (\mu A + (1 - \mu)A') x \geq 0\).

Actully: psd is “cone” constraint.
Semidefinite Programming: polynomial time.

Ellipsoid algorithm.
Semidefinite Programming: polynomial time.

Ellipsoid algorithm.

Center point not feasible. New Ellipsoid.

$\leq \left(1 - \frac{1}{\text{poly}(n)}\right)$ volume.

Center point feasible? Linear Programming: find violated constraint. Semidefinite Programming: find $x$ where $x^T A x \leq 0$. Compute smallest eigenvalue. Only get close! $O(\log \frac{1}{\epsilon})$ dependence on closeness.
Semidefinite Programming: polynomial time.

Ellipsoid algorithm.

Center point not feasible.

Enclosing Ellipse.

Linear Programming: find violated constraint.

Semidefinite Programming: find $x$ where $x^T Ax \leq 0$.

Compute smallest eigenvalue.

Only get close!

$O(\log \frac{1}{\epsilon})$ dependence on closeness.
Semidefinite Programming: polynomial time.

Ellipsoid algorithm.

Enclosing Ellipse. Center point not feasible. New Ellipsoid.
Semidefinite Programming: polynomial time.

Ellipsoid algorithm.

Enclosing Ellipse.
Center point not feasible.
New Ellipsoid.
\[ \leq (1 - 1/\text{poly}(n)) \text{ volume}. \]
Semidefinite Programming: polynomial time.

Ellipsoid algorithm.

Enclosing Ellipse.
Center point not feasible.
New Ellipsoid.
\[ \leq (1 - 1/\text{poly}(n)) \text{ volume.} \]
Center point feasible?
Semidefinite Programming: polynomial time.

Ellipsoid algorithm.

Enclosing Ellipse.
Center point not feasible.
New Ellipsoid.
\[ \leq (1 - 1/\text{poly}(n)) \text{ volume}. \]
Center point feasible?
Linear Programming:
find violated constraint.
Semidefinite Programming: polynomial time.

Ellipsoid algorithm.

Enclosing Ellipse. Center point not feasible. New Ellipsoid.

\[ \leq (1 - 1/\text{poly}(n)) \text{ volume.} \]

Center point feasible? Linear Programming:
find violated constraint.

Semidefinite Programming:
find \( x \) where \( x^T Ax \leq 0 \).
Semidefinite Programming: polynomial time.

Ellipsoid algorithm.

Enclosing Ellipse.
Center point not feasible.
New Ellipsoid.
\[ \leq (1 - 1/\text{poly}(n)) \text{ volume.} \]
Center point feasible?
Linear Programming:
find violated constraint.
Semidefinite Programming:
find \( x \) where \( x^T A x \leq 0 \).
Compute smallest eigenvalue.
Semidefinite Programming: polynomial time.

Ellipsoid algorithm.

Enclosing Ellipse.
Center point not feasible.
New Ellipsoid.
\[ \leq \left(1 - \frac{1}{\text{poly}(n)}\right) \text{volume}.\]
Center point feasible?
Linear Programming:
find violated constraint.
Semidefinite Programming:
find \(x\) where \(x^T Ax \leq 0\).
Compute smallest eigenvalue.

Only get close!
\[O(\log \frac{1}{\varepsilon})\text{ dependence on closeness.}\]
Semidefinite Programming: another view.

$$\begin{align*}
\text{max} & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*}$$

(2)
Semidefinite Programming: another view.

\[
\begin{align*}
\text{max} & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*}
\]  
(2)

Recall \( A = B^T B \).
Semidefinite Programming: another view.

\[
\begin{align*}
\max & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*}
\] (2)

Recall \( A = B^TB \).

Programming over vectors: \( v_1, v_2, \ldots, v_n \).
Semidefinite Programming: another view.

\[
\begin{align*}
\text{max} \quad & A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*}
\] (2)

Recall \( A = B^T B \).

Programming over vectors: \( v_1, v_2, \ldots, v_n \).
Semidefinite Programming: another view.

\[
\begin{align*}
\max \quad & A.C \\
A.X_i \quad & \geq \quad b_i \\
A \quad & \succeq \quad 0
\end{align*}
\]

Recall \( A = B^T B \).

Programming over vectors: \( v_1, v_2, \ldots, v_n \).

Linear Constraints over \( v_i \cdot v_j \)
Semidefinite Programming: another view.

\[
\begin{align*}
\text{max} & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*}
\]  \quad (2)

Recall \( A = B^T B \).

Programming over vectors: \( v_1, v_2, \ldots, v_n \).

Linear Constraints over \( v_i \cdot v_j \)

quadratic..kind of!
Semidefinite Programming: another view.

$$\max A.C$$

$$A.X_i \geq b_i$$

$$A \succeq 0$$ \hfill (2)

Recall $A = B^T B$.

Programming over vectors: $v_1, v_2, \ldots, v_n$.

- Linear Constraints over $v_i \cdot v_j$
- Quadratic..kind of!
Semidefinite Programming: another view.

\[
\begin{align*}
\text{max} & \quad A.C \\
A.X_i & \geq b_i \\
A & \succeq 0
\end{align*}
\] (2)

Recall \( A = B^T B \).

Programming over vectors: \( v_1, v_2, \ldots, v_n \).

Linear Constraints over \( v_i \cdot v_j \)

quadratic..kind of!
Max Cut

Given a graph \( G = (V, E) \), with \( w : E \rightarrow R \), find \( S \) where \( \sum_{e \in S \times S} w_e \) is maximized.
Max Cut

Given a graph $G = (V, E)$, with $w : E \rightarrow R$, find $S$ where \( \sum_{e \in S \times S} w_e \) is maximized.

Max Cut Size: 4
Factor half approximation?

Random: choose a side at random.
Random: choose a side at random.
Each edge has probability $\frac{1}{2}$ of being cut.
Factor half approximation?

Random: choose a side at random.
Each edge has probability \( \frac{1}{2} \) of being cut.
Expected value of solution is half total edge weight.
Factor half approximation?

Random: choose a side at random.

Each edge has probability $\frac{1}{2}$ of being cut.

Expected value of solution is half total edge weight.

Greedy: choose larger choice.
Factor half approximation?

Random: choose a side at random.
Each edge has probability $\frac{1}{2}$ of being cut.
Expected value of solution is half total edge weight.
Greedy: choose larger choice.
    When each node comes, cuts at least half previous edges.
Factor half approximation?

Random: choose a side at random.
Each edge has probability $\frac{1}{2}$ of being cut.
Expected value of solution is half total edge weight.

Greedy: choose larger choice.
  When each node comes, cuts at least half previous edges.
Can we do better?
Embedding problem.

Assign variables $x_1, \ldots, x_n$ to vertices.
Embedding problem.

Assign variables $x_1, \ldots, x_n$ to vertices.

$x_i$ are $\pm 1$. 
Embedding problem.

Assign variables $x_1, \ldots, x_n$ to vertices.

$x_i$ are $\pm 1$.

Maximize $\sum_{ij} w_{ij} \frac{1-x_i x_j}{2}$.
Embedding problem.

Assign variables $x_1, \ldots, x_n$ to vertices.

$x_i$ are $\pm 1$.

Maximize $\sum_{ij} w_{ij} \frac{1-x_i x_j}{2}$.

Cost of cut indicated by $\pm 1$ vector!
Embedding problem.

Assign variables $x_1, \ldots, x_n$ to vertices.

$x_i$ are $\pm 1$.

Maximize $\sum_{ij} w_{ij} \frac{1-x_i x_j}{2}$.

Cost of cut indicated by $\pm 1$ vector!

Integer?
Embedding problem.

Assign variables $x_1, \ldots, x_n$ to vertices.

$x_i$ are $\pm 1$.

Maximize $\sum_{ij} w_{ij} \frac{1-x_i x_j}{2}$.

Cost of cut indicated by $\pm 1$ vector!

Integer? Quadratic?
Embedding problem.

Assign variables $x_1, \ldots, x_n$ to vertices.

$x_i$ are $\pm 1$.

Maximize $\sum_{ij} w_{ij} \frac{1-x_i x_j}{2}$.

Cost of cut indicated by $\pm 1$ vector!

Integer? Quadratic?

Assign vectors $v_1, v_2, \ldots, v_n$ to vertices.
Embedding problem.

Assign variables $x_1, \ldots, x_n$ to vertices.
$x_i$ are $\pm 1$.
Maximize $\sum_{ij} w_{ij} \frac{1-x_i x_j}{2}$.
Cost of cut indicated by $\pm 1$ vector!
Integer? Quadratic?
Assign vectors $v_1, v_2, \ldots, v_n$ to vertices.
$|v_i| = 1$
Embedding problem.

Assign variables $x_1, \ldots, x_n$ to vertices.
$x_i$ are $\pm 1$.

Maximize $\sum_{ij} w_{ij} \frac{1-x_i x_j}{2}$.

Cost of cut indicated by $\pm 1$ vector!

Integer? Quadratic?

Assign vectors $v_1, v_2, \ldots, v_n$ to vertices.
$|v_i| = 1$

Maximize $\sum_{ij} w_{ij} \frac{1-v_i \cdot v_j}{2}$. 
Embedding problem.

Assign variables $x_1, \ldots, x_n$ to vertices.

$x_i$ are $\pm 1$.

Maximize $\sum_{ij} w_{ij} \frac{1 - x_i x_j}{2}$.

Cost of cut indicated by $\pm 1$ vector!

Integer? Quadratic?

Assign vectors $v_1, v_2, \ldots, v_n$ to vertices.

$|v_i| = 1$

Maximize $\sum_{ij} w_{ij} \frac{1 - v_i \cdot v_j}{2}$.

Semidefinite Program.
Embedding problem.

Assign variables $x_1, \ldots, x_n$ to vertices.
$x_i$ are $\pm 1$.
Maximize $\sum_{ij} w_{ij} \frac{1-x_i x_j}{2}$.
Cost of cut indicated by $\pm 1$ vector!
Integer? Quadratic?

Assign vectors $v_1, v_2, \ldots, v_n$ to vertices.
$|v_i| = 1$
Maximize $\sum_{ij} w_{ij} \frac{1-v_i \cdot v_j}{2}$.
Semidefinite Program. Can solve?
Embedding problem.

Assign variables $x_1, \ldots, x_n$ to vertices.
$x_i$ are $\pm 1$.
Maximize $\sum_{ij} w_{ij} \frac{1-x_i x_j}{2}$.
Cost of cut indicated by $\pm 1$ vector!
Integer? Quadratic?

Assign vectors $v_1, v_2, \ldots, v_n$ to vertices.

$|v_i| = 1$
Maximize $\sum_{ij} w_{ij} \frac{1-v_i \cdot v_j}{2}$.

Semidefinite Program. Can solve? (Basically.)
Assign vectors $v_1, v_2, \ldots, v_n$ to vertices.
Assign vectors $v_1, v_2, \ldots, v_n$ to vertices.

$|v_i| = 1$
Assign vectors $v_1, v_2, \ldots, v_n$ to vertices.

$|v_i| = 1$

Maximize $\sum_{ij} w_{ij} \frac{1 - v_i \cdot v_j}{2}$.
Assign vectors \(v_1, v_2, \ldots, v_n\) to vertices.

\[ |v_i| = 1 \]

Maximize \(\sum_{ij} w_{ij} \frac{1 - v_i \cdot v_j}{2} \).

Example?
Assign vectors $v_1, v_2, \ldots, v_n$ to vertices.

$|v_i| = 1$

Maximize $\sum_{ij} w_{ij} \frac{1 - v_i \cdot v_j}{2}$.

Example?
Assign vectors $v_1, v_2, \ldots, v_n$ to vertices.

$|v_i| = 1$

Maximize $\sum_{ij} w_{ij} \frac{1 - v_i \cdot v_j}{2}$.

Example?

Solution Value: $5^{\frac{(1 - \cos(4\pi/5))}{2}} \approx 4.52$
Assign vectors $v_1, v_2, \ldots, v_n$ to vertices.

$|v_i| = 1$

Maximize $\sum_{ij} w_{ij} \frac{1 - v_i \cdot v_j}{2}$.

Example?

Solution Value: $5 \frac{(1 - \cos(4\pi/5))}{2} \approx 4.52$

Higher than opt.
Assign vectors \( v_1, v_2, \ldots, v_n \) to vertices.

\[ |v_i| = 1 \]

Maximize \( \sum_{ij} w_{ij} \frac{1 - v_i \cdot v_j}{2} \).

Example?

Solution Value: \( 5 \left( \frac{1 - \cos(4\pi/5)}{2} \right) \approx 4.52 \)

Higher than opt.

Round and not lose too much?
Hyperplane rounding.

Normal to hyperplane: red does not separate! green does.
Hyperplane rounding.
Hyperplane rounding.

Normal to hyperplane:

red does not separate!

green does.
Hyperplane rounding.

Normal to hyperplane: red does not separate!
Hyperplane rounding.

Normal to hyperplane: red does not separate! green does.
Hyperplane rounding.

Take a random vector, \( w \)
Hyperplane rounding.

Take a random vector, $w$
Let $S = \{ w \cdot v \geq 0 \}$
Take a random vector, \( w \)

Let \( S = \{ w \cdot v \geq 0 \} \)

**Claim 1:** Expected weight of \((S, V - S)\) is at least 0.878 SDPOPT.
Hyperplane rounding.

Take a random vector, \( w \)
Let \( S = \{ w \cdot v \geq 0 \} \)

**Claim 1:** Expected weight of \((S, V - S)\) is at least 0.878 SDPOPT.
Hyperplane rounding.

Take a random vector, $w$
Let $S = \{ w \cdot v \geq 0 \}$

**Claim 1:** Expected weight of $(S, V - S)$ is at least 0.878 SDPOPT.

SDP value for edge $(i, j)$.

$$\frac{(1 - \cos \theta)}{2}$$
Hyperplane rounding.

Take a random vector, $w$

Let $S = \{ w \cdot v \geq 0 \}$

**Claim 1:** Expected weight of $(S, V - S)$ is at least 0.878 SDPOPT.

**SDP value for edge $(i,j)$:**

$$\frac{(1 - \cos \theta)}{2}$$

Prob. of cutting:

$1 - \cos \theta$
Hyperplane rounding.

Take a random vector, $w$
Let $S = \{ w \cdot v \geq 0 \}$

**Claim 1:** Expected weight of $(S, V - S)$ is at least 0.878 SDPOPT.

SDP value for edge $(i,j)$.

$$\frac{(1 - \cos \theta)}{2}$$

Prob. of cutting:

$$\frac{\theta}{\pi}$$

Expected value in rounding!
Hyperplane rounding.

Take a random vector, \( w \)

Let \( S = \{ w \cdot v \geq 0 \} \)

**Claim 1:** Expected weight of \( (S, V - S) \) is at least 0.878 SDPOPT.

SDP value for edge \((i,j)\).  
\[
\frac{(1 - \cos \theta)}{2}
\]

Prob. of cutting:  
\[
\frac{\theta}{\pi}
\]

Expected value in rounding!  

\[
\text{Ratio is } \frac{2\theta}{\pi (1 - \cos \theta)}.
\]
Hyperplane rounding.

Take a random vector, \( w \)

Let \( S = \{ w \cdot v \geq 0 \} \)

**Claim 1:** Expected weight of \((S, V - S)\) is at least 0.878 SDPOPT.

SDP value for edge \((i, j)\).

\[
\frac{(1 - \cos \theta)}{2}
\]

Prob. of cutting:

\[
\frac{\theta}{\pi}
\]

Expected value in rounding!

Ratio is

\[
\frac{2\theta}{\pi(1 - \cos \theta)}
\]

Always bigger than .878!
See you on Thursday.