Today

Semidefinite Programming

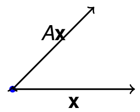
...for Approximating MaxCut.

Positive Semidefinite Matrices

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

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.

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

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

Possibly many such representations.

Semidefinite Programming.

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

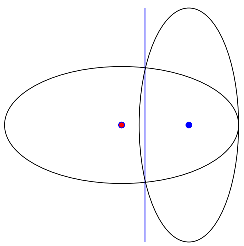
$A.X$ is matrix inner product: $\sum_{ij} a_{ij} x_{ij}$.
view A and X as n^2 dimensional vector.

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.



Enclosing Ellipse.
Center point not feasible.
New Ellipsoid.
$\leq (1 - 1/\text{poly}(n))$ 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!
$O(\log \frac{1}{\varepsilon})$ dependence on closeness.

Semidefinite Programming: another view.

$$\begin{aligned} \max \quad & A.C \\ A.X_i \quad & \geq \quad b_i \\ A \quad & \succeq \quad 0 \end{aligned} \tag{2}$$

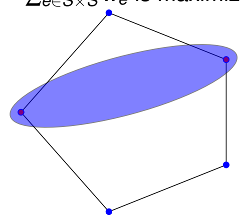
Recall $A = B^T B$.

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 Size: 4

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.

x_i are ± 1.

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

Cost of cut indicated by ± 1 vector!

Integer? Quadratic?

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

$|v_i| = 1$

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

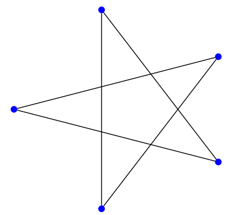
Semidefinite Program. Can solve? (Basically.)

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

$|v_i| = 1$

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

Example?

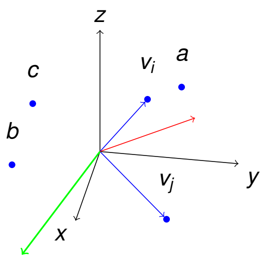


Solution Value: $5 \frac{(1 - cos(4\pi/5))}{2} \approx 4.52$

Higher than opt.

Round and not lose too much?

Hyperplane rounding.



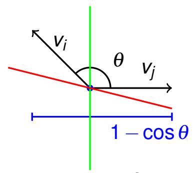
Normal to hyperplane:
red does not separate!
green does.

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.