Intro to semidefinite programming

linear programming where vars are entries in a psd matrix

Defn: If A is a symmetric n by n matrix then the following statements are equivalent:

1. X is positive semidefinite $=$ psd $\equiv X \succcurlyeq 0$
2. $\forall y \in \mathbb{R}^n \quad y^T X y \geq 0$
3. X has nonnegative eigenvalues
4. $X = V^T V$ for some $m \times n$ matrix V $\quad m \leq n$
5. $X = \sum_{i=1}^{n} \lambda_i w_i w_i^T$ for some $\lambda_i \geq 0$ and orthonormal vectors $w_i \in \mathbb{R}^n$

Semidefinite program (SDP)

max or min $\sum_{i,j} c_{ij} x_{ij}$

subject to $\sum_{i,j} a_{ijk} x_{ij} = b_k \quad \forall k$

$x_{ij} = x_{ji} \quad \forall i,j$

$X = (x_{ij}) \succeq 0$

max or min $\sum_{i,j} c_{ij} (v_i \cdot v_j)$

subject to $\sum_{i,j} a_{ijk} (v_i \cdot v_j) = b_k$

$v_i \in \mathbb{R}^n \quad i = 1, .., n$

given $X \Rightarrow X = V^T V$, set v_i to be i^{th} col of V

Key Fact:

SDPs can be solved to within additive error ε in time

$\text{poly}(\text{size of input}, \log(\frac{1}{\varepsilon}))$

(in our discussions, we ignore additive error ε

MAXCUT

Input : $G=(V,E)$ $\quad w_{ij}\ \forall (i,j)\in E$

Goal: partition vertex set so as to max weight of endpts crossing cut.

IP formulation of MAXCUT

x_i encodes side of partition $\quad i\in V$

$$z_{ij} = \begin{cases} 1 & \text{edge } (i,j) \text{ cut.} \\ 0 & \text{o.w.} \end{cases}$$

$$\max \sum_{(i,j)\in E} w_{ij}\, z_{ij}$$

$$z_{ij} \le x_i + x_j \qquad \forall (i,j)\in E$$

$$z_{ij} \le 2-(x_i+x_j) \qquad \forall (i,j)\in E$$

LP opt $=1$ $\forall$ graphs!

$$x_i \in \{0,1\} \qquad i\in V$$
$$z_{ij} \in \{0,1\} \qquad \forall (i,j)\in E$$

Quadratic Programming Formulation

$$\max \frac{1}{2} \sum_{(i,j) \in E} w_{ij} [1 - y_i y_j]$$

$$y_i \cdot y_i = 1 \quad \forall i \in V$$

OPT is exactly optimal soln to MAXCUT

still NP-hard

Vector Programming Relaxation

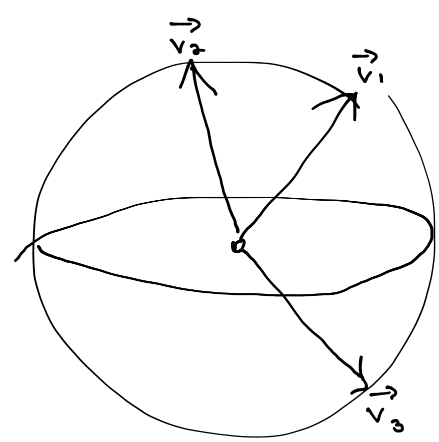
$$\max \quad \frac{1}{2} \sum_{(i,j) \in E} w_{ij} (1 - \vec{v}_i \cdot \vec{v}_j)$$

Z^* optimum of SDP

$$\vec{v}_i \cdot \vec{v}_i = 1 \quad \forall i \in V$$

$$v_i \in \mathbb{R}^n$$

Claim: $Z^* \geq OPT$



Can solve SDP, but how to round?

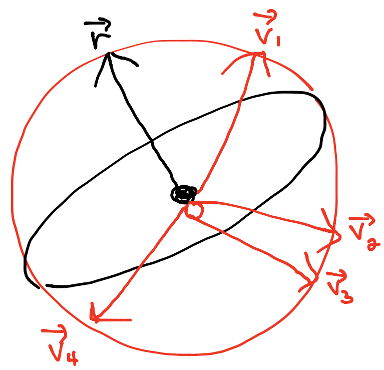
get large contribution to OPT when $v_i \cdot v_j$ very -ve

Random hyperplane rounding

Solve SDP $\rightarrow v_1^*, \ldots, v_n^*$

pick random hyperplane thru origin.

Partition vertices based on which side of hyperplane corresponding vectors are.



Let $\vec{r}$ by random unit vector

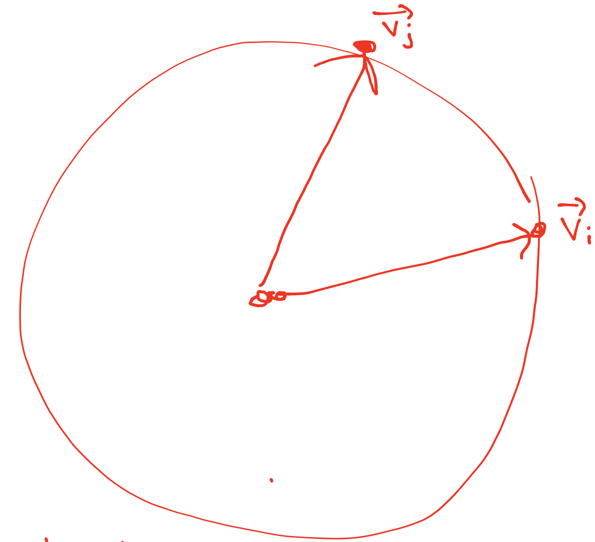
(e.g. Let $\vec{g} = (r_1, \ldots, r_n)$ $\quad r_i \sim N(0,1)$ iid.

$$\vec{r} = \frac{\vec{g}}{\|\vec{g}\|}$$

Put $i \rightarrow U$ if $\vec{v}_i \cdot \vec{r} \geq 0$

$i \rightarrow \overline{U}$ if $\vec{v}_i \cdot \vec{r} < 0$

$\Pr((i,j) \text{ gets cut}) = ?$



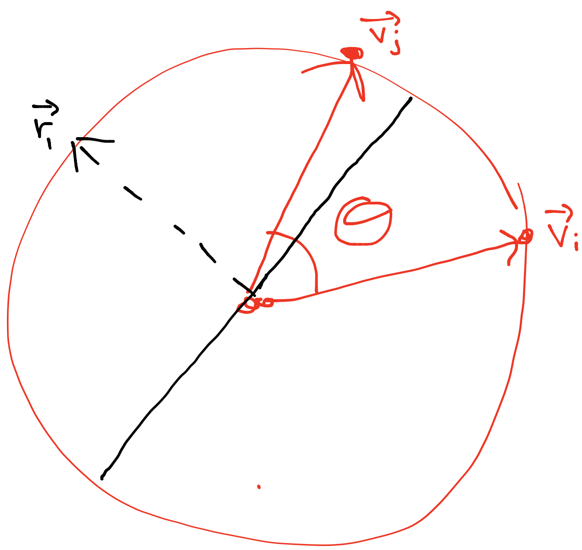
2D plane containing $\vec{v}_i, \vec{v}_j$

Since hyperplane selected from rotationally symmetric distn, the prob that hyperplane cuts these 2 vectors = Pr random diam lies in between

$$\vec{r} = \underset{\substack{\uparrow \\ \text{in 2D plane}}}{\vec{r}'} + \underset{\substack{\uparrow \\ \text{orthogonal}}}{\vec{r}''}$$

$$\vec{v}_i \cdot \vec{r}'' = \vec{v}_j \cdot \vec{r}'' = 0$$

$\frac{r'}{\|r'\|}$ uniformly dist'd on circle



$\Pr(i \,\&\, j \text{ separated})$

$$= \frac{2\theta}{2\pi} = \frac{\theta}{\pi} = \frac{\arccos(\vec{v}_i \cdot \vec{v}_j)}{\pi}$$

$$E(\text{wt of cut}) = \sum_{(i,j)\in E} w_{ij} \Pr(i \,\&\, j \text{ separated})$$

$$= \sum_{(i,j)\in E} w_{ij} \frac{\arccos(v_i \cdot v_j)}{\pi} \geq \alpha \sum_{(i,j)\in E} w_{ij} \left(\frac{1 - v_i \cdot v_j}{2}\right)$$

?

$$\alpha \geq \min_{-1 \leq x \leq 1} \frac{\frac{1}{\pi}\arccos(x)}{\frac{1}{2}(1-x)} \geq 0.878$$

$$E(\text{cut produced by alg}) \geq 0.878 \text{ OPT}$$