MAX CUT and SEMIDEFINITE PROGRAMMING

14/11/2025

Problem: Given $G=(V,E)$, possibly with weights $w: E \to \mathbb{R}^+$, find a cut $(S, V\setminus S)$ maximizing $|E(S, V\setminus S)|$ (or $\sum_{e\in E(S,V\setminus S)} w(e)$ in the weighted version)

2-approximations

1. order vertices arbitrarily - $v_1, v_2, \dots, v_n$

   for $i=1$ to n: color v_i <u>white</u> if the number of black left neighbors of v_i is at least equal the number of the left black neighbors of v_i, otherwise color v_i <u>black</u>

   For each v_i, at least half of its left neighbors has different color than v_i

   $\Rightarrow$ For $S = \{v_i : v_i \text{ black}\}$, $(S, V\setminus S)$ is 2-approxim.

2. For each $v \in V$, color it white with prob. $\frac{1}{2}$, and black with prob $\frac{1}{2}$.

   $\Rightarrow \mathbb{E}[\#\text{edges in cut}] = \frac{|E|}{2} \Rightarrow$ 2-approximation.

It holds: It is NP-hard to approximate MAX CUT better than $\frac{16}{17} \doteq 0{,}941$.

Flows, ... 7/1

Quadratic programm for MAX CUT

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

s.t. $y_i^2 = 1$, $y_i \in \mathbb{R}$ $\forall i \in V$

For $y_i \in \mathbb{R}$, either $y_i = 1$ or $y_i = -1$ $y_i = y_j$ $y_i \neq y_j$

$\Rightarrow$ contribution of $\{i,j\}$ is either 0 or 1 (w_{ij})

$\Rightarrow$ gives the optimal solution.

As we can't solve quadratic programmes in poly time, we relax the constraints:

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

Vector programming relaxation for MAX CUT

(2) $\max \sum_{\{i,j\}\in E} \frac{1-v_i \cdot v_j}{2}$ dot product

s.t. $v_i^2 = 1$, $v_i \in \mathbb{R}^n$ $\forall i \in V$

For a solution $y_1, \dots, y_n$ of (1),

$\underbrace{(0,0,\dots,0,y_1)}_{=v_1}, \dots, \underbrace{(0,0,\dots,0,y_n)}_{=v_n}$ is a solution of (2)

with the same objective value.

$\Rightarrow$ (2) is a relaxation of (1), i.e., (2) provides an upper bound on (1)

Vectors seem more complicated than numbers – how does it help?

Flows, ...

SEMIDEFINITE PROGRAMMING

Let $C, D_1, \dots, D_k$ be symmetric $n \times n$ real matrices, $d_1, \dots, d_k \in \mathbb{R}$.

$\max C \cdot X$ — $\cdot$ is standard dot product (multiply component wise and sum)

s.t. $D_i \cdot X = d_i$ for $i = 1, \dots, k$ $\quad X \succeq 0$

X is positive semidefinite (i.e., $\forall x \in \mathbb{R}^n,\ u^T X u \geq 0$)

Recall: $X \succeq 0$ iff $\exists\, U$ s.t. $X = U^T U$ … Cholesky decomposition.

Efficiently computable (up to tiny factor… ignore).

THM. $\forall\, \varepsilon > 0$, the SDP problem can be solved with additive error ε in poly time with respect to n, k and $\log\left(\frac{1}{\varepsilon}\right)$.

$\rightsquigarrow$ we will ignore this tiny error.

What is the relation between vector and SDP programs?

Consider the following SDP:

$$(3) \quad \max \sum_{\{i,j\} \in E} \frac{1 - x_{ij}}{2}$$

$$\text{s.t.} \quad x_{ii} = 1 \quad \forall i \in V$$

$$X \succeq 0.$$

- For feasible solution $v_1, \dots, v_n \in \mathbb{R}^n$ of (2), define $x_{ij} = v_i^T v_j$, $\forall i, j$, i.e., $X = U^T U$ where $U = \{v_1\, v_2 \cdots v_n\}$. Then X is a feasible solution of (3) with the same objective value.
- On the other hand, for a feasible X of (3), let U be s.t. $X = U^T U$. Then the columns of U provide a feasible solution of (2) with the same objective value.

APPROXIMATING MAX CUT

Algorithm:

1. Solve the SDP relaxation (3) of MAX CUT
2. Transform it (Cholesky decomposition) into a vector solution $v_1, \dots, v_n \in \mathbb{R}^n$ of (2).
3. Sample uniformly at random $r \in \{x \in \mathbb{R}^n : x^2 = 1\}$ and OUTPUT $S = \{i \in V : v_i^T r \geq 0\}$.

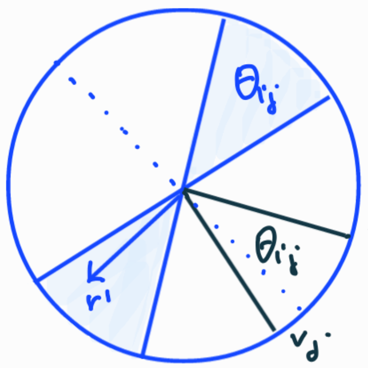
Note: in step 3, we are separating the unit vectors v_i by a random hyperplane - - the one that is perpendicular to r.

Let θ_{ij} denote the angle between v_i and v_j. Then $\cos\theta_{ij} = v_i v_j$.

The contribution of the edge $\{i,j\} \in E$ to the objective value of (2) is $\frac{1 - \cos\theta_{ij}}{2}$. note: the larger $|\theta_{ij}|$, the larger contribution

Lemma: For $\{i,j\} \in E$, $\Pr[i \text{ and } j \text{ separated}] = \frac{\theta_{ij}}{\pi}$.

Proof: Consider the projection r' of the vector r to the plane spanned by v_i, v_j.

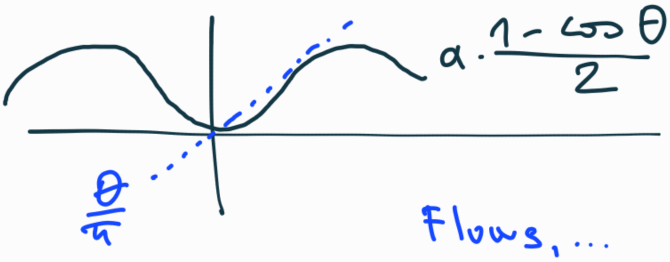


The edge $\{i,j\}$ is cut if r' lies in the "double wedge". As r is chosen uniformly at random, this happens with prob. $\frac{\theta_{ij}}{\pi}$

(note: $r \cdot v_i = r' \cdot v_i$, $r \cdot v_j = r' \cdot v_j$ ... $r = r' + r''$, $r'' \perp v_i$, $r'' \perp v_j$).

Note: For every $\theta \in \langle 0, \pi \rangle$, $\frac{\theta}{\pi} \geq \alpha \cdot \frac{1 - \cos\theta}{2}$ where (*)

$$\alpha = \frac{2}{\pi} \min_{0 \leq \theta \leq \pi} \frac{\theta}{1 - \cos\theta}.$$



Lemma: $\alpha \doteq 0{,}878567$

(math. analysis)

Flows, ...

Let W be the size of the constructed solution S, $W = |E(S, V\setminus S)|$.
Then
$$\mathbb{E}[W] = \sum_{\{i,j\}\in E} \frac{\theta_{ij}}{\pi} \geq \alpha \cdot \sum_{\{i,j\}\in E} \frac{1-\cos\theta_{ij}}{2} = \alpha \cdot OPT$$

THM: The expected size of the cut constructed by the algorithm is at least $0{,}878567 \cdot OPT$.

MAX 2SAT

given a formula s.t. each clause contains ≤ 2 literals, find a truth-assignment maximizing the number of satisfied clauses.

Consider a clause $C = x_i \vee x_j$, and define the expression
$$v(C) = \frac{1+y_0y_i}{4} + \frac{1+y_0y_j}{4} + \frac{1-y_iy_j}{4} = \ldots \overset{\text{if } y_0^2=1}{=} 1 - \frac{1-y_0y_i}{2} \cdot \frac{1-y_0y_j}{2}$$
Similarly for other clauses.

Given a truth assignment for the x_i variables,

define $y_0 = 1$ and $\forall i \geq 1$, $y_i = \begin{cases} 1 & \text{if } x_i = \text{true} \\ -1 & x_i = \text{false} \end{cases}$

Then C satisfied iff $v(C) = 1$.

x_1	x_2	$v(C)$
f	f	0
t	f	1
f	t	1
t	t	1

On the other hand, given $y \in \{-1,1\}^{n+1}$,
define $x_i = \begin{cases} \text{true} & \text{if } y_i = y_0 \\ \text{false} & \text{if } y_i \neq y_0 \end{cases}$

Then C satisfied iff $v(C) = 1$.

$$\Rightarrow \quad \max \sum_{C \in \varphi} v(C)$$
$$y_i^2 = 1 \quad i = 0, 1, \ldots, n$$

vector program:
$$\max \sum_{0 \leq i < j \leq n} \left[a_{ij} \cdot (1 + v_i \cdot v_j) + b_{ij} \cdot (1 - v_i \cdot v_j)\right]$$
$$v_i^2 = 1 \quad i = 0, 1, \ldots, n$$

is a quadratic program for MAX 2SAT

$\rightsquigarrow$ vector relaxation $\rightsquigarrow$ SDP $\rightsquigarrow$ vector solution
$v_0, v_1, \ldots, v_n \in \mathbb{R}^{n+1}$.

Rounding for MAX 2SAT:

pick uniformly at random a unit vector $r \in \mathbb{R}^{n+1}$

set $y_i = \begin{cases} 1 & \text{if } r^T \cdot v_i \geq 0 \\ -1 & r^T v_i < 0 \end{cases}$, for $i = 0, 1, \dots, n$

set $x_i = \begin{cases} \text{true} & \text{if } y_0 = y_i \\ \text{false} & y_0 = -y_i \end{cases}$

Lemma: The expected number of satisfied clauses is $\alpha \cdot OPT_{SDP}$.

Proof: each term $1 - v_i v_j$ is rounded to 2 in the $y_0, y_1, \dots, y_n$ solution with probability $\frac{\theta_{ij}}{\pi}$, and to 0 otherwise - by MAX CUT analysis.

For the same reasons, each term $1 + v_i v_j$ is rounded to 2 with prob. $1 - \frac{\theta_{ij}}{\pi}$, and to 0 otherwise.

$1 - \frac{\theta_{ij}}{\pi} \geq \alpha \cdot \frac{1 + \cos \theta_{ij}}{2}$. by (*) - substitute $\theta' = \pi - \theta$, and use $\cos(\pi - \theta) = -\cos\theta$

Let W be the number of satisfied clauses by x.

$$\mathbb{E}[W] = \sum_{0 \leq i < j \leq n} \left[2 \cdot a_{ij} \left(1 - \frac{\theta_{ij}}{\pi}\right) + 2 b_{ij} \frac{\theta_{ij}}{\pi} \right] \geq$$

$$\geq \sum_{0 \leq i < j \leq n} \left[a_{ij} \cdot \alpha \cdot (1 + \cos\theta_{ij}) + b_{ij} \cdot \alpha \cdot (1 - \cos\theta_{ij}) \right] =$$

$$\geq \alpha \cdot \sum_{0 \leq i < j \leq n} \left[a_{ij} \cdot (1 + v_i v_j) + b_{ij} (1 - v_i v_j) \right] = \alpha \cdot OPT_{SDP} \quad \blacksquare$$

Plows, ...

7/6