CMPUT 498/501: Advanced Algorithms **Fall 2025**

Lecture 19 (Nov 17, 2025): Semidefinite Programming: Max-Cut & Max-2SAT

Lecturer: Mohammad R. Salavatipour *Scribe: Kaining Yang*

19.1 Introduction to Semidefinite Programming

Semidefinite programming (SDP) is a powerful generalization of linear programming. While LP optimizes over the cone of non-negative vectors, SDP allows us to optimize over the much more expressive cone of positive semidefinite matrices.

Quadratic Programs (QP): A problem of optimizing a quadratic function of variables subject to a set of quadratic constraints.

19.1.1 Strict Quadratic Programming

Strict QP: If each of the constraints and the objective function consists only of degree zero or degree 2 monomials.

Definition 1 *Let $x \in \mathbb{R}^{n\times n}$ be a symmetric real matrix. We say x is PSD (Positive Semidefinite) if $\forall a \in \mathbb{R}^n$: $a^T x a \geq 0$.*

Theorem 1 *If $x \in \mathbb{R}^{n\times n}$, then the following are equivalent:*

1. *$x \succeq 0$ (is PSD)*
2. *x has only non-negative eigenvalues*
3. *$\exists V \in \mathbb{R}^{m\times n}$, $m \leq n$ such that $x = V^T V$*
4. *$x = \sum_{i=1}^{n} \lambda_i \omega_i \cdot \omega_i^T$ for $\lambda_i \geq 0$, $\omega_i \in \mathbb{R}^n$ with $\omega_i^T \omega_i = 1$ and $\omega_i^T \omega_j = 0$ for $i \neq j$*

Trace of $A \in \mathbb{R}^{n\times n}$: $\mathrm{Tr}(A)$ is the sum of diagonal entries.

Definition 2 (Frobenius Inner Product) *For matrices $A, B \in \mathbb{R}^{n\times n}$:*

$$A \bullet B = \mathrm{Tr}(A^T B) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{ij}$$

Let M_n be the cone of symmetric $\mathbb{R}^{n\times n}$ matrices. $X \succeq 0$ denotes X is PSD. Let $C, D_1, \ldots, D_k \in M_n$ and $d_1, \ldots, d_k \in \mathbb{R}$.

SDP Formulation:

$$\begin{aligned} \min/\max \quad & \sum_{ij} c_{ij} \cdot x_{ij} && \text{or} \quad C \bullet X \\ \text{subject to} \quad & \sum_{i,j} d_{ij}^{(\ell)} \cdot x_{ij} = d_\ell && \text{or} \quad D_\ell \bullet X = d_\ell, \quad \forall 1 \le \ell \le k \\ & X \succeq 0 \\ & X \in M_n \end{aligned}$$

If $C, D_1, \ldots, D_k$ are diagonal, then this turns into an LP. SDP is a convex program. They can be solved in polynomial time with additive error ε (for any $\varepsilon > 0$). Polynomial in n and $\log(\frac{1}{\varepsilon})$.

19.2 Vector Programming

SDP is equivalent to vector programming. Let $V_1, \ldots, V_n$ be n-dimensional vectors $V_i \in \mathbb{R}^n$.

$$\begin{aligned} \min/\max \quad & \sum_{i,j} c_{ij} (\vec{v}_i \cdot \vec{v}_j) \\ \text{subject to} \quad & \sum_{i,j} a_{ij,\ell} (\vec{v}_i \cdot \vec{v}_j) = b_\ell, \quad \forall 1 \le \ell \le k \\ & \vec{v}_i \in \mathbb{R}^n \end{aligned}$$

The corresponding SDP is defined as follows: it has n^2 variables y_{ij}; replace $\vec{v}_i \cdot \vec{v}_j$ by y_{ij}. Additionally require Y be PSD.

Lemma 1 *The Vector Program and the corresponding SDP are equivalent.*

Proof. We show any feasible solution of V corresponds to a feasible solution of SDP with the same value.

Let $W = \begin{bmatrix} \vec{x}_1 & \vec{x}_2 & \cdots & \vec{x}_n \end{bmatrix}$ where $\vec{x}_1, \ldots, \vec{x}_n$ is a solution to V. Then $X = W^T W$ is feasible to SDP with the same value (other direction is similar). ■

19.3 Max-Cut

Input: Undirected graph $G(V, E)$, edge weights $w : E \to \mathbb{R}^+$.
Goal: Find a subset $S \subseteq V$ that maximizes $w(\delta(S)) = \sum_{e \in \delta(S)} w_e$.

The integrality gap for all known LP's is 2. Random partition gives a $\frac{1}{2}$-approximation for Max-Cut. We can formulate Max-Cut exactly using variables $y_i \in \{-1, 1\}$:

$$\begin{aligned} \max \quad & \frac{1}{2} \sum_{1 \le i < j \le n} w_{ij} (1 - y_i y_j) \qquad \text{(IP)} \\ \text{subject to} \quad & y_i^2 = 1 \\ & y_i \in \mathbb{Z} \end{aligned}$$

We relax this to a Vector Program (VP) by replacing $y_i y_j$ with $\vec{v}_i \cdot \vec{v}_j$:

$$
\begin{aligned}
(VP) \quad \max \quad & \frac{1}{2}\sum_{i<j} w_{ij}(1 - \vec{v}_i \cdot \vec{v}_j) \\
\text{s.t.} \quad & \vec{v}_i \cdot \vec{v}_i = 1 \\
& \vec{v}_i \in \mathbb{R}^n
\end{aligned}
$$

Since $\vec{v}_i \cdot \vec{v}_i = 1$, all vectors lie on the n-dimensional sphere S_n.

Example 1 (Integrality Gap on C_5) *Consider a 5-cycle (C_5) with edge weights $w(e) = 1$.*

- ***OPT:*** *The maximum cut removes all but one edge, so* $\text{OPT} = 4$.
- ***VP Solution:*** *The optimal vector solution arranges the vectors in $\mathbb{R}^2$ such that the angle between adjacent vertices is $\theta = 4\pi/5$.*

$$
Z_{VP} = 5 \cdot \left(\frac{1 - \cos(4\pi/5)}{2}\right) = \frac{25 + 5\sqrt{5}}{8} \approx 4.52
$$

Thus, the integrality gap is $\frac{\text{OPT}}{Z_{VP}} \approx 0.884$. The larger the angle θ_{ij} (closer to π), the larger the contribution to the cut.

19.3.1 Max-Cut SDP Rounding

Algorithm 1 Max-Cut SDP Rounding

1: Solve the VP and get vectors $\vec{v}$
2: Choose a random vector r on unit sphere S_n
3: Let $S = \{i \mid \vec{v}_i \cdot r \geq 0\}$
4: Return S

Vector r is normal to a hyperplane. All vectors on the same side as r will be selected (to S).

$$
\vec{v}_i \cdot \vec{v}_j = \cos(\theta_{ij})|\vec{v}_i||\vec{v}_j|
$$

$\vec{v}_i \cdot r$ and $\vec{v}_j \cdot r$ have different signs.

Lemma 2 $\Pr[v_i$ *and* v_j *are separated by* $r] = \dfrac{\theta_{ij}}{\pi}$

Proof. Consider the plane defined by $\vec{v}_i, \vec{v}_j$ and project r onto this plane. $\vec{v}_i$ and $\vec{v}_j$ are separated iff r lies between the arcs of θ_{ij}:

$$
\begin{aligned}
r = r' + r'' \quad & \text{so } \vec{v}_i \cdot r = \vec{v}_i \cdot (r' + r'') \\
& = \vec{v}_i \cdot r' \quad \text{since } r'' \text{ is orthogonal} \\
\vec{v}_j \cdot r = \vec{v}_j \cdot r' &
\end{aligned}
$$

Let $X_{ij} = \begin{cases} 1 & \text{if } e = ij \text{ is across the cut} \\ 0 & \text{otherwise} \end{cases}$. Then $W = \sum_{ij} w_{ij} \cdot X_{ij}$. ∎

$$\mathbb{E}[W] = \sum_{ij} w_{ij} \cdot \Pr[v_i \text{ and } v_j \text{ are sep}] = \sum_{ij} w_{ij} \cdot \frac{\theta_{ij}}{\pi}$$

Let $\alpha_{GW} = \frac{2}{\pi} \min_{\theta \in (0,\pi]} \frac{\theta}{1 - \cos\theta}$. For any θ, $\frac{\theta}{\pi} \geq \alpha \left(\frac{1-\cos\theta}{2}\right)$.

$$\Rightarrow \mathbb{E}[W] \geq \alpha \sum_{ij} \frac{1}{2} w_{ij}(1 - \cos\theta_{ij}) = \alpha \cdot Z_{\text{SDP}}$$

Lemma 3 $\alpha_{GW} \geq 0.8785$

Theorem 2 (Khot et al.) *Assuming UGC, there is no $(\alpha_{GW} - \varepsilon)$-approximation for Max-Cut for any $\varepsilon > 0$.*

19.4 Max-2SAT using SDP

We saw that LP gives a $\frac{3}{4}$-approximation for Max-2SAT and the gap was $\frac{3}{4}$ (tight). We show how we can do better using SDP.

$$\begin{aligned} y_i &= \pm 1 \quad \text{for } x_i \\ y_0 &= y_i \quad \text{iff } x_i \text{ is True} \end{aligned}$$

Value of a clause C: $v(C) = 1$ iff C is satisfied.

$$\begin{aligned} v(x_i) &= \frac{1 + y_i y_0}{2} \\ v(\overline{x}_i) &= \frac{1 - y_i y_0}{2} \\ v(x_i \vee x_j) &= 1 - v(\overline{x}_i) v(\overline{x}_j) = 1 - \frac{1 - y_i y_0}{2} \cdot \frac{1 - y_j y_0}{2} \end{aligned}$$

$$\begin{aligned} &= \frac{1}{4}\left(3 + y_i y_0 + y_j y_0 - y_i y_j y_0^2\right) \\ &= \frac{1 + y_i y_0}{4} + \frac{1 + y_j y_0}{4} + \frac{1 - y_i y_j}{4} \quad (\text{since } y_0^2 = 1) \end{aligned}$$

So each clause is a linear combination of $1 + y_i y_j$ and $1 - y_i y_j$. The objective is to maximize the weighted sum $\sum w_C v(C)$:

$$\max \sum \left[a_{ij}(1 + y_i y_j) + b_{ij}(1 - y_i y_j)\right]$$

subject to:

$$y_i^2 = 1$$
$$y_i = \pm 1$$

$$\max \sum a_{ij}(1 + \vec{v}_i \cdot \vec{v}_j) + b_{ij}(1 - \vec{v}_i \cdot \vec{v}_j)$$

subject to:

$$\vec{v}_i \cdot \vec{v}_i = 1$$
$$\vec{v}_i \in \mathbb{R}^{n+1}$$

Like Max-Cut: take a random $r \in S_{n+1}$, round each y_i to ± 1 iff $r \cdot \vec{v}_i \geq 0$.

Lemma 4 $\mathbb{E}[W] \geq \alpha \cdot Z_{SDP}$

$$\mathbb{E}[W] = 2 \sum [a_{ij} \cdot \Pr[y_i = y_j] + b_{ij} \cdot \Pr[y_i \neq y_j]]$$

where $\vec{v}_i \cdot \vec{v}_j = \theta_{ij}$:

$$\Pr[y_i \neq y_j] = \frac{\theta_{ij}}{\pi} \geq \frac{\alpha}{2}(1 - \cos\theta_{ij})$$
$$\Pr[y_i = y_j] = 1 - \frac{\theta_{ij}}{\pi} \geq \frac{\alpha}{2}(1 + \cos\theta_{ij})$$

$$\Rightarrow \mathbb{E}[W] \geq \alpha \sum a_{ij}(1 + \cos\theta_{ij}) + b_{ij}(1 - \cos\theta_{ij}) = \alpha \cdot Z_{\text{SDP}}$$

Theorem 3 (LLZ/JNZ) *Max-2SAT has an SDP rounding algorithm with ratio 0.94016 and this is best possible under UGC.*

19.5 General CSP

General CSP with range of values $\{1, \ldots, k\}$:

$$(x_1, x_5, x_6) \in \left\{ \begin{pmatrix} 2 \\ 1 \\ 6 \end{pmatrix}, \begin{pmatrix} 5 \\ 5 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 8 \\ 11 \end{pmatrix} \right\}$$