CMPUT 675: Topics on Approximation Algorithms and Approximability **Fall 2015**

Lecture 17–18 Semidefinite Programming, Max-Cut, Max-2SAT: Oct 25–27

Lecturer: Mohammad R. Salavatipour *Scribe: Roshan Shariff*

17.1 Semidefinite Programming

Quadratic programming is concerned with optimizing a quadratic function of variables subject to quadratic constraints. A quadratic program is *strict* if the objective function and each of the constraints consist only of degree 0 or 2 monomials. Here we are concerned with a type of strict quadratic program called a *semidefinite program.*

Definition 17.1 *Let $x \in \mathbb{R}^{n\times n}$ be a symmetric $n \times n$ real matrix. We say that x is* positive semidefinite *(and write $x \succeq 0$) if $a^T x a \geq 0$ for all $a \in \mathbb{R}^n$.*

Theorem 17.2 *If $x \in \mathbb{R}^{n\times n}$, the following are equivalent:*

(a) $x \succeq 0$.

(b) x *has non-negative eigenvalues.*

(c) $x = v^T v$ *for some* $v \in \mathbb{R}^{m\times n}$ *with* $m \geq n$.

(d) $x = \sum_{i=1}^{m} \lambda_i w_i w_i^T$ *for some* $\lambda_i \geq 0$ *and* $w_i \in \mathbb{R}^n$ *with* $w_i^T w_i = 1$ *and* $w_i^T w_j = 0$ *for* $i \neq j$.

In the following, let $C, D_1, D_2, \ldots, D_k \in \mathbb{R}^{n\times n}$ be symmetric matrices and $d_1, d_2, \ldots, d_k \in \mathbb{R}$ be constants.

Definition 17.3 *A* semidefinite program *is an optimization problem of the form*

$$\begin{aligned} \max / \min \sum_{1\leq i,j\leq n} C_{ij} x_{ij}, & \qquad x \in \mathbb{R}^{n\times n}; \\ \textit{subject to: } \sum_{1\leq i,j\leq n} D_{l,ij} x_{ij} = d_l, & \quad \textit{for all } 1 \leq l \leq k; \\ x \succeq 0. & \end{aligned}$$

Using the notation $A \cdot B$ (for $A, B \in \mathbb{R}^{n\times n}$) to mean $\mathrm{tr}(A^T B) = \sum_i \sum_j A_{ij} B_{ij}$, we can also write a semidefinite program as

$$\begin{aligned} \max / \min\ C \cdot x & \qquad x \in \mathbb{R}^{n\times n}; \\ \textit{subject to: } D_l \cdot x = d_l, & \quad \textit{for all } 1 \leq l \leq k; \\ x \succeq 0. & \end{aligned}$$

If the matrices C and $D_1, D_2, \ldots, D_k$ are diagonal, then the above semidefinite program is a linear program.

Definition 17.4 *A* vector program *is an optimization problem of the form*

$$\begin{aligned} &\max / \min \sum_{1\le i,j\le n} C_{ij}\langle \vec{v}_i, \vec{v}_j\rangle, && \vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n \in \mathbb{R}^n; \\ &\textit{subject to: } \sum_{1\le i,j\le n} D_{l,ij}\langle \vec{v}_i, \vec{v}_j\rangle = d_l, && \textit{for all } 1 \le l \le k. \end{aligned}$$

The n vectors $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n \in \mathbb{R}^n$ give n^2 variables, with $Y_{ij} = \langle \vec{v}_i, \vec{v}_j\rangle$. The matrix Y is always positive semidefinite.

Lemma 17.5 *A vector program is equivalent to the corresponding semidefinite program defined by the matrix Y as above.*

Proof: *Given a solution $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n \in \mathbb{R}^n$ to the vector program, let $W \in \mathbb{R}^{n\times n}$ be defined as*

$$W = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \vec{v}_1 & \vec{v}_2 & \ldots & \vec{v}_n \\ \vdots & \vdots & & \vdots \end{bmatrix}$$

and let $x = W^T W$. By condition (c) of Theorem 17.2, $x \succeq 0$, so it is a feasible solution to the semidefinite program. Moreover, $x_{ij} = \langle \vec{v}_i, \vec{v}_j\rangle$, so it has the same objective value.

The converse proof is left as an exercise. ■

For any given $\varepsilon > 0$, we can find a solution to the semidefinite program with additive error ε.

17.2 Max-Cut

Given an undirected graph $G = (V, E)$ with weights $w : E \to \mathbb{Q}^+$, the Max-Cut problem is to find a maximal cut S:

$$\max_{S\subset V} \sum_{e\in\delta(S)} w(e),$$

where $\delta(S)$ is the set of edges with one vertex in S and the other not in S.

The randomized algorithm that independently picks each edge with probability 1/2 is a trivial 1/2-approximation for this problem. To try to do better, consider the following integer program formulation:

$$\begin{aligned} &\text{maximize:} && \frac{1}{2}\sum_{(i,j)\in E} w_{ij}(1 - y_i y_j), && y_i \in \mathbb{Z} \\ &\text{subject to:} && y_i^2 = 1, && \text{for all } i \in E. \end{aligned}$$

Since this is an integer program, the constraint ensures $y_i \in \{-1, 1\}$ for each $i \in E$. A vector program relaxation of this integer program is:

$$\begin{aligned} &\text{maximize:} && \frac{1}{2}\sum_{(i,j)\in E} w_{ij}(1 - \vec{v}_i \cdot \vec{v}_j), && \vec{v}_i \in \mathbb{R}^n \\ &\text{subject to:} && \vec{v}_i \cdot \vec{v}_i = 1, && \text{for all } i \in E. \end{aligned}$$

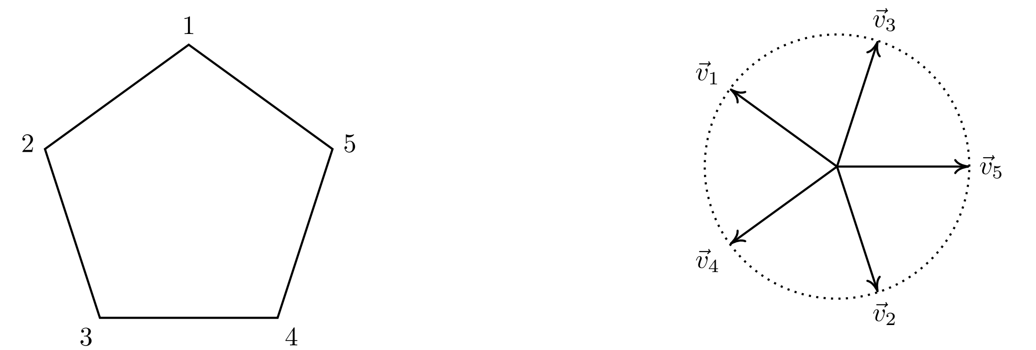
Given a solution y to the integer program, setting $\vec{v}_i = (y_i, 0, \ldots, 0)$ for each $i \in V$ gives a feasible solution to the vector program with the same objective value.

17.2.1 Example

Figure 17.1a shows a cyclic graph $G = (V, E)$ with 5 vertices. If each edge has weight 1, the maximum cut has a value of $\mathrm{OPT}_{\mathrm{MC}} = 4$. Figure 17.1b shows the vectors $\vec{v}_1, \ldots, \vec{v}_5$ that are the optimal solution to the above vector program relaxation. The angle between $\vec{v}_i$ and $\vec{v}_j$ for any $(i, j) \in E$ is $4\pi/2$, so $\vec{v}_i \cdot \vec{v}_j = \cos(4\pi/5)$. The value of the vector program objective is therefore

$$Z_{\mathrm{VP}} = \frac{5(1 - \cos(4\pi/2))}{2} \approx 4.52$$

Any rounding procedure that produces an integer solution based on this vector program solution will therefore incur an approximation ratio of at least $\mathrm{OPT}_{\mathrm{MC}}/Z_{\mathrm{VP}} \approx 0.885$. With a good rounding strategy, we could do better than the naive randomized algorithm which has a ratio of 1/2.



(a) A cyclic graph with 5 vertices.

(b) Optimal max-cut vector program solution.

Figure 17.1: An example of a graph and the optimal solution to the corresponding max-cut vector program relaxation.

17.2.2 Random Hyperplane Rounding

VP Max-Cut Rounding

let $\vec{v}_1, \ldots, \vec{v}_n \leftarrow$ optimal solution to above vector program
let $\vec{r} \leftarrow$ uniformly random from the unit n-sphere
return $S = \{i : \vec{v}_i \cdot r \geq 0\}$

Note: To sample the random vector $\vec{r}$ uniformly from the unit n-dimensional sphere, sample each of its components from a standard normal distribution. The resulting vector has a spherically symmetric distribution, so it is enough to then normalize it.

Lemma 17.6 *For any distinct $i, j \in V$, the probability that i and j are separated by the cut is θ_{ij}/π, where θ_{ij} is the angle between $\vec{v}_i$ and $\vec{v}_j$ in the vector program solution.*

Proof: Let $\vec{s}$ be the projection of $\vec{r}$ onto the plane containing $\vec{v}_i$ and $\vec{v}_j$. Then $\vec{r} - \vec{s}$ is perpendicular to both $\vec{v}_i$ and $\vec{v}_j$, so

$$\begin{aligned} \vec{v}_i \cdot \vec{r} &= \vec{v}_i \cdot (\vec{s} + \vec{r} - \vec{s}) \\ &= (\vec{v}_i \cdot \vec{s}) + \vec{v}_i \cdot (\vec{r} - \vec{s}) \\ &= \vec{v}_i \cdot \vec{s}. \end{aligned}$$

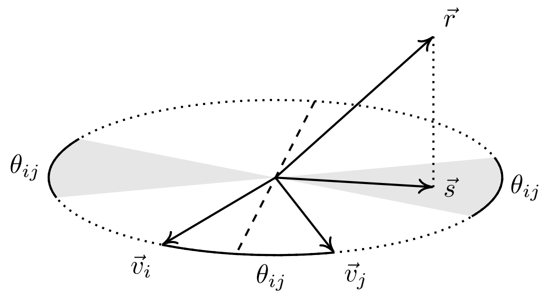


Figure 17.2: Vectors $\vec{v}_i$ and $\vec{v}_j$ are separated by the dashed line perpendicular to $\vec{r}$ whenever $\vec{s}$ lies in either of the two shaded regions, each subtending an angle of θ_{ij}.

Similarly, $\vec{v}_j \cdot \vec{r} = \vec{v}_j \cdot \vec{s}$. Consider expressing $\vec{v}_i$, $\vec{v}_j$, and $\vec{s}$ using polar coordinates. Without loss of generality, $\vec{v}_i$ has an angular coordinate of 0, $\vec{v}_j$ has angular coordinate θ_{ij}, and $\vec{s}$ has angular coordinate ϕ. Now $\vec{s}$ separates $\vec{v}_i$ and $\vec{v}_j$ if and only if $\pi/2 \leq \phi \leq \pi/2 + \theta_{ij}$ or $3\pi/2 \leq \phi \leq 3\pi/2 + \theta_{ij}$. Because $\vec{r}$ has a spherically symmetric distribution on the n-dimensional sphere, the angular coordinate of $\vec{s}$ is uniformly distributed in $[0, 2\pi)$. Thus the above condition is satisfied with probability $2 \cdot \theta_{ij}/2\pi = \theta_{ij}/\pi$. ■

Theorem 17.7 *The above algorithm is a 0.8785-approximation for Max-Cut.*

Proof: We define

$$\alpha = \frac{2}{\pi} \min_{0 \leq \theta \leq \pi} \frac{\theta}{1 - \cos\theta} \approx 0.8785,$$

so that for any θ we have

$$\frac{\theta}{\pi} \geq \alpha \left(\frac{1 - \cos\theta}{2} \right).$$

If X_{ij} is the indicator random variable that is 1 if vertices $i, j \in V$ are separated by the cut and 0 otherwise, the expected weight of the cut produced by the above algorithm is

$$\begin{aligned}
E[W] &= E\Big[\sum_{(i,j)\in E} w_{ij} X_{ij} \Big] \\
&= \sum_{(i,j)\in E} w_{ij} \frac{\theta_{ij}}{\pi} \\
&\geq \alpha \cdot \frac{1}{2} \sum_{(i,j)\in E} w_{ij} (1 - \cos\theta_{ij}) \\
&= \alpha \cdot \frac{1}{2} \sum_{(i,j)\in E} w_{ij} (1 - \vec{v}_i \cdot \vec{v}_j) \\
&= \alpha \cdot Z_{\mathrm{VP}} \geq \alpha \cdot \mathrm{OPT}_{\mathrm{MC}}.
\end{aligned}$$

We use the fact that $\vec{v}_i \cdot \vec{v}_j = \|\vec{v}_i\| \cdot \|\vec{v}_j\| \cdot \cos\theta_{ij}$, and $\|\vec{v}_i\| = \|\vec{v}_j\| = 1$. ■

Theorem 17.8 (Hasdard, 1997) *Unless* P = NP*, Max-Cut has no* β*-approximation where* $\beta > 16/17 \approx$ 0.941.

Theorem 17.9 *Assuming the Unique Games Conjecture (UGC), there is no* $(\alpha + \varepsilon)$*-approximation for Max-Cut.*

17.3 Max-2SAT

The Max-2SAT problem is concerned with logical formulae in 2-conjunctive normal form (2-CNF), which is a formula like:

$$(x_1 \vee x_2) \wedge (\overline{x}_3 \vee x_2) \wedge \cdots .$$

There are n literals $x_i, \ldots, x_n$ and m clauses in the conjunction, and each clause is the disjunction of at most two literals and their negations. The Max-2SAT problem is to find an assignment of truth values to the literals that maximizes the number of satisfied clauses; it is NP-hard.

The natural linear program relaxation of the problem has an integrality gap of 4/3, which is no better than random assignment. Instead, we look at an SDP relaxation:

$$\begin{aligned} y_i &= \pm 1, && \text{for } i = 0, \ldots, m; \\ y_0 &= y_i, && \text{if and only if } x_i \text{ is true.} \end{aligned}$$

To define the objective function, we want each clause C to have a value $v(C)$ that is 1 if the clause is satisfied, and 0 otherwise:

$$\begin{aligned} v(x_i) &= \frac{1 + y_i y_0}{2}, && \text{for clauses of one variable.} \\ v(\overline{x}_i) &= \frac{1 - y_i y_0}{2} \\ v(x_i \vee x_j) &= 1 - v(\overline{x}_i) v(\overline{x}_j) && \text{for clauses of two variables.} \\ &= 1 - \frac{1 - y_i y_0}{2} \cdot \frac{1 - y_j y_0}{2} \\ &= \frac{3 + y_i y_0 + y_j y_0 - y_i y_j y_0^2}{4} \\ &= \frac{1 + y_i y_0}{4} + \frac{1 + y_j y_0}{4} + \frac{1 - y_i y_j}{4} \\ v(\overline{x}_i \vee x_j) &= \frac{1 - y_i y_0}{4} + \frac{1 + y_j y_0}{4} + \frac{1 + y_i y_j}{4} \\ v(x_i \vee \overline{x}_j) &= \frac{1 + y_i y_0}{4} + \frac{1 - y_j y_0}{4} + \frac{1 + y_i y_j}{4} \\ v(\overline{x}_i \vee \overline{x}_j) &= \frac{1 - y_i y_0}{4} + \frac{1 - y_j y_0}{4} + \frac{1 - y_i y_j}{4} \end{aligned}$$

We see that the terms in the value function are of the form $c(1 + y_i y_j)$ or $c(1 - y_i y_j)$, so by collecting the coefficients of like terms we can write the objective function as:

$$\max \sum_{0 \leq i,j \leq n} a_{ij}(1 + y_i y_j) + b_{ij}(1 - y_i y_j), \quad y_i = \pm 1.$$

As we did before for MAX CUT, we relax the above into a vector program:

$$\max \sum_{0 \leq i,j \leq n} a_{ij}(1 + \vec{v}_i \cdot \vec{v}_j) + a_{ij}(1 - \vec{v}_i \cdot \vec{v}_j), \quad v_i \in \mathbb{R}^{n+1}, \vec{v}_i \cdot \vec{v}_i = 1.$$

VP Max-2SAT Rounding

let $\vec{v}_0, \ldots, \vec{v}_n \leftarrow$ optimal solution to above vector program.
let $\vec{r} \leftarrow$ uniformly random from the unit n-sphere.
let $y_i \leftarrow 1$ if $\vec{v}_i \cdot \vec{r} \geq 0$, $y_i \leftarrow 0$ otherwise.

4 **let** $x_i \leftarrow$ TRUE if and only if $y_i = y_0$.

Theorem 17.10 *The above algorithm is a 0.8785-approximation for Max-2SAT.*

Proof: The expected weight of a cut produced by the above algorithm is

$$E[W] = \sum_{0 \le i,j \le n} a_{ij} P[y_i = y_j] + b_{ij} P[y_i \neq y_j].$$

From the argument given for Max-Cut above, we have

$$\begin{aligned} P[y_i \neq y_j] &= \theta_{ij}/\pi \geq \alpha(1 - \cos\theta_{ij})/2, \\ P[y_i = y_j] &= 1 - \theta_{ij}/\pi \geq \alpha(1 - \cos\theta_{ij})/2. \end{aligned}$$

Thus

$$E[W] \geq \alpha Z_{\text{SDP}} \approx 0.8785 Z_{\text{SDP}}.$$

■

Note: A result of Livnat, Lewin, and Zwick (2002) improves the approximation ratio to 0.940. There is also an upper bound on the ratio of 0.943.