Part 23
Insertion: Semidefinite Programming

Source: *Approximation Algorithms* (Vazirani, Springer Press)

Positive definite matrices

Definition (positive semidefinite Matrix)

A matrix $A \in \mathbb{R}^{n \times n}$ is called positive semi-definite if

$$\forall x \in \mathbb{R}^n : x^T A x \geq 0.$$

Theorem (Diagonalization)

Let $A \in \mathbb{R}^{n \times n}$ be symmetric (i.e. $a_{ij} = a_{ji}$), then A is diagonalizable, i.e. one can write

$$A = \underbrace{\begin{pmatrix} \boxed{\begin{matrix} \vdots \\ v_1 \\ \vdots \end{matrix}} & \dots & \boxed{\begin{matrix} \vdots \\ v_n \\ \vdots \end{matrix}} \end{pmatrix}}_{=L} \cdot \underbrace{\begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \dots & \dots & \ddots & \dots \\ 0 & 0 & \dots & \lambda_n \end{pmatrix}}_{=D} \cdot \underbrace{\begin{pmatrix} \boxed{\dots \quad v_1 \quad \dots} \\ \vdots \\ \boxed{\dots \quad v_n \quad \dots} \end{pmatrix}}_{=L^T}$$

where $v_i \in \mathbb{R}^n$ is orthonormal Eigenvector for Eigenvalue λ_i, i.e $Av_i = \lambda_i v_i$, $\|v_i\|_2 = 1, v_i^T v_j = 0 \; \forall i \neq j$.

Some useful results

Lemma

Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix (v_i orthonormal Eigenvector for λ_i). Then the following statements are equivalent

(1) $\forall x \in \mathbb{R}^n : x^T A x \geq 0$

(2) $\lambda_i \geq 0 \; \forall i$

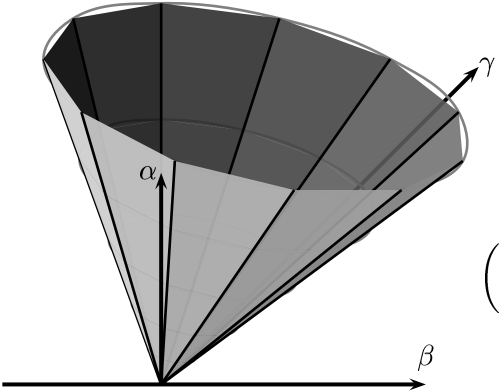
(3) *There is* $W \in \mathbb{R}^{n \times n}$ *with* $A = W^T W$

- (1) $\Rightarrow$ (2). $0 \leq v_i^T A v_i = v_i^T (\lambda_i v_i) = \lambda_i \underbrace{v_i^T v_i}_{=1} = \lambda_i$
- (2) $\Rightarrow$ (3). $A = LDL^T = L\sqrt{D}\sqrt{D}L^T = (\sqrt{D}L^T)^T \underbrace{(\sqrt{D}L^T)}_{=:W}$
- (3) $\Rightarrow$ (1). For any $x \in \mathbb{R}^n$:

$$x^T A x = x^T (W^T W) x = (Wx)^T \cdot (Wx) \geq 0$$

Remark: Matrix W can be found by Cholesky decomposition in $O(n^3)$ arithmetic operations (if $\sqrt{\ }$ counts as 1 operation).

The semidefinite cone



$$\begin{pmatrix} \alpha & \beta \\ \beta & \gamma \end{pmatrix} \succeq 0$$

- **Def.:** Write $Y \succeq 0$ if Y is positive semidefinite.
- **Fact:** The set

$$\{Y \in \mathbb{R}^{n\times n} \mid Y \succeq 0, Y \text{ symmetric}\} = \text{cone}\{xx^T \mid x \in \mathbb{R}^n\}$$

is a convex, non-polyhedral cone.

A semidefinite program

Given:

- Obj. function vector $C = (c_{ij})_{1 \leq i,j \leq n} \in \mathbb{Q}^{n \times n}$
- Linear constraints $A_k = (a^k_{ij})_{1 \leq i,j \leq n} \in \mathbb{Q}^{n \times n}$, $b_k \in \mathbb{Q}$

$$
\begin{aligned}
\max \sum_{i,j} c_{ij} y_{ij} & & \\
\sum_{i,j} a^k_{ij} y_{ij} & \leq b_k \quad \forall k = 1, \ldots, m \\
Y & \quad \text{symmetric} \\
Y & \succeq 0
\end{aligned}
$$

- Frobenius inner product: $C \bullet Y := \sum_{i=1}^n \sum_{j=1}^n c_{ij} \cdot y_{ij}$

A semidefinite program

Given:

- Obj. function vector $C = (c_{ij})_{1 \leq i,j \leq n} \in \mathbb{Q}^{n \times n}$
- Linear constraints $A_k = (a_{ij}^k)_{1 \leq i,j \leq n} \in \mathbb{Q}^{n \times n}$, $b_k \in \mathbb{Q}$

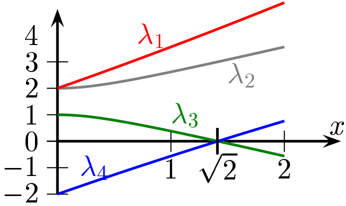
$$\begin{array}{rcl} \max C \bullet Y & & \\ A_k \bullet Y & \leq & b_k \quad \forall k = 1, \dots, m \\ Y & & \text{symmetric} \\ Y & \succeq & 0 \end{array}$$

- Frobenius inner product: $C \bullet Y := \sum_{i=1}^n \sum_{j=1}^n c_{ij} \cdot y_{ij}$

Pathological situations

- **Case: All solutions might be irrational.** $x = \sqrt{2}$ is the unique solution of

$$\begin{pmatrix} 1 & x & 0 & 0 \\ x & 2 & 0 & 0 \\ 0 & 0 & 2x & 2 \\ 0 & 0 & 2 & x \end{pmatrix} \succeq 0$$



- **Case: All sol. might have exponential encoding length.** Let $Q_1(x) = x_1 - 2, Q_i(x) := \begin{pmatrix} 1 & x_{i-1} \\ x_{i-1} & x_i \end{pmatrix}$. Then

$$Q(x) := \begin{pmatrix} Q_1(x) & 0 & \dots & 0 \\ 0 & Q_2(x) & \dots & 0 \\ \dots & \dots & \ddots & \vdots \\ 0 & 0 & \dots & Q_n(x) \end{pmatrix} \succeq 0$$

if and only if $Q_1(x), \dots, Q_n(x) \succeq 0$. I.e. $x_1 - 2 \geq 0$ and $x_i \geq x_{i-1}^2$, hence $x_n \geq 2^{2^n - 1}$.

Solvability of Semidefinite Programs

Theorem

Given rational input $A_1, \dots, A_m, b_1, \dots, b_m, C, R$ *and* $\varepsilon > 0$, *suppose*

$$SDP = \max\{C \bullet Y \mid A_k \bullet Y \leq b_k \ \forall k; \ Y \, symmetric; \ Y \succeq 0\}$$

is feasible and all feasible points are contained in $B(\mathbf{0}, R)$. *Then one can find a* Y^* *with*

$$A_k \bullet Y^* \leq b_k + \varepsilon, \ Y^* \ symmetric, \ Y^* \succeq 0$$

such that $C \bullet Y^* \geq SDP - \varepsilon$. *The running time is polynomial in the input length,* $\log(R)$ *and* $\log(1/\varepsilon)$ *(in the Turing machine model).*

Solving the separation problem

- **Remark:** We show that we can solve the separation problem, ignore numerical inaccuracies.
- Let infeasible Y be given, we have to find a separating hyperplane.

(1) *Case* $A_k \bullet Y < b_k$*:* return "$A_k \bullet Y \geq b_k$ violated"

(2) *Case* Y *not symmetric:* Find the i, j with $y_{ij} < y_{ji}$. Return "$y_{ij} \geq y_{ji}$ violated".

(3) *Case* Y *not positive semidefinite.* Find eigenvector v with Eigenvalue $\lambda < 0$, i.e. $Yv = \lambda v$. Then

$$\sum_{i,j} v_i^T v_j \cdot y_{ij} = v^T Y v < 0$$

hence return "$\sum_{i,j} v_i^T v_j \cdot y_{ij} \geq 0$ violated".

Vectorprograms

Idea:

$$\begin{aligned} & Y \text{ symmetric and } Y \succeq 0 \\ \Leftrightarrow \quad & \exists W = (v_1, \ldots, v_n) \in \mathbb{R}^{n \times n} : W^T W = Y \\ \Leftrightarrow \quad & \exists v_1, \ldots, v_n \in \mathbb{R}^n : y_{ij} = v_i^T v_j \end{aligned}$$

SDP:

$$\max \sum_{i,j} c_{ij} y_{ij}$$

$$\begin{aligned} \sum_{i,j} a_{ij}^k \cdot y_{ij} & \leq b_k \quad \forall k \\ Y & \quad \text{sym.} \\ Y & \succeq 0 \end{aligned}$$

Vector program:

$$\max \sum_{i,j} c_{ij} v_i^T v_j$$

$$\begin{aligned} \sum_{i,j} a_{ij}^k \cdot v_i^T v_j & \leq b_k \quad \forall k \\ v_i & \in \mathbb{R}^n \quad \forall i \end{aligned}$$

Observation

The SDP and the vector program are equivalent.

PART 24
MAXCUT

SOURCE:

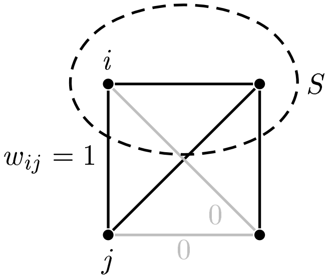
- *Approximation Algorithms* (Vazirani, Springer Press)
- *Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming* (Goemans, Williamson) (link)

Problem definition

Problem: MaxCut

- Given: Complete undirected graph $G = (V, E)$, edge weights $w : E \to \mathbb{Q}_+$
- Find: Cut maximizing the weight of separated edges

$$OPT = \max_{S \subseteq V} \left\{ \sum_{e \in \delta(S)} w(e) \right\}$$



A vector program

- Choose decision variable for any node $i \in V$:

$$v_i = \begin{cases} (\ \ \ 1, 0, \ldots, 0) & i \in S \\ (-1, 0, \ldots, 0) & i \notin S \end{cases}$$

- An exact MaxCut vector program:

$$\max \sum_{(i,j) \in E} \frac{w_{ij}}{2}(1 - v_i^T v_j)$$

$$\begin{aligned} v_i^T v_i &= 1 & \forall i = 1, \ldots, n \\ v_i &\in \mathbb{R}^n & \forall i = 1, \ldots, n \\ v_i &= (\pm 1, 0, \ldots, 0) & \forall i = 1, \ldots, n \end{aligned}$$

- Then

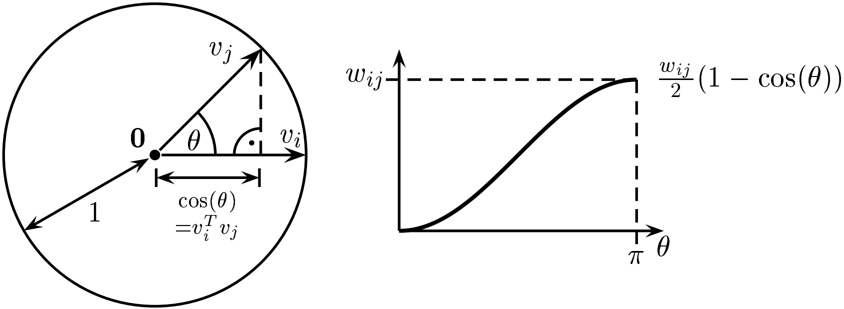
$$\sum_{(i.j) \in E} w_{ij} \cdot \overbrace{\frac{1}{2}(1 - \underbrace{v_i^T v_j}_{\substack{=-1 \text{ if } (i,j) \in \delta(S) \\ +1 \text{ o.w.}}})}^{=1 \text{ if } (i,j) \in \delta(S),\ 0 \text{ o.w.}} = \sum_{(i,j) \in \delta(S)} w_{ij}$$

A vector program (2)

The relaxed vector program:

$$\max \sum_{(i,j)\in E} \frac{w_{ij}}{2}(1 - v_i^T v_j)$$

$$\begin{aligned} v_i^T v_i &= 1 \quad \forall i = 1, \dots, n \\ v_i &\in \mathbb{R}^n \quad \forall i = 1, \dots, n \end{aligned}$$

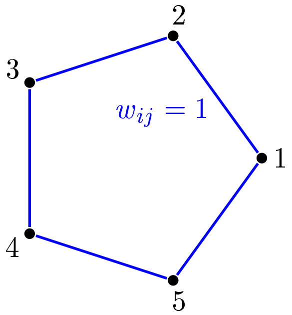


A physical interpretation

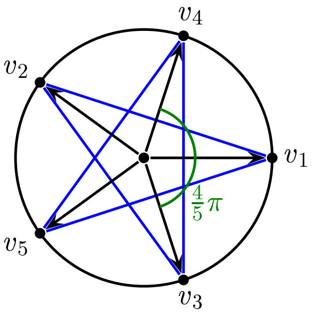
- n vectors on n-dim unit ball.
- Repulsion force of w_{ij} between v_i and v_j

Example:

Graph G



SDP solution:



- $OPT = 4$
- For SDP solution, place $v_1, \ldots, v_5$ equidistantly on 2-dim. subspace. $SDP = 5 \cdot \frac{1}{2}(1 - \cos(\frac{4}{5}\pi)) \approx 4.52$
- Hence integrality gap ≥ 1.13.

The algorithm

Algorithm:

(1) Solve MaxCut vector program $\rightarrow v_1, \ldots, v_n \in \mathbb{Q}^n$
(More precisely: Solve the equivalent SDP, obtain a matrix $Y \in \mathbb{Q}^{n \times n}$. Apply Cholesky decomposition to Y to obtain $v_1, \ldots, v_n$)

(2) Choose randomly a vector r from n-dimensional unit ball

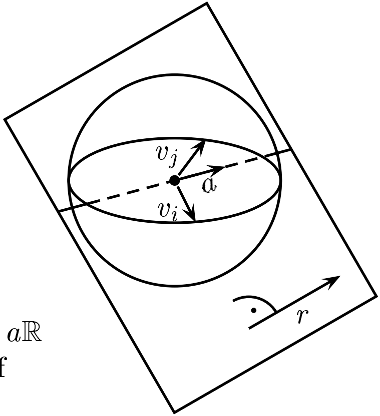
(3) Choose cut $S := \{i \mid v_i \cdot r \geq 0\}$

Theorem

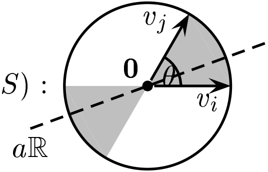
$E[\sum_{(i,j)\in\delta(S)} w_{ij}] \geq 0.87 \cdot OPT$ *(i.e. the algorithm gives an expected* 1.13*-apx).*

Proof

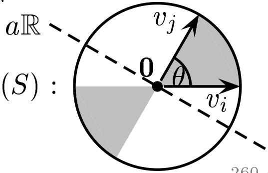
- Consider 2 vectors v_i, v_j with angle $\theta \in [0, \pi]$. Let $\mathbb{R} \cdot a$ be the 1-dim. intersection of the $n-1$-dim. hyperplane $x \cdot r = 0$ with the plane spanned by v_i, v_j
- a has a random direction
- v_i, v_j are separated
  $\Leftrightarrow$ they lie on different sides of line $a\mathbb{R}$
  $\Leftrightarrow$ a lies in one of the 2 gray arcs of angle θ
- $\Pr[v_i \text{ and } v_j \text{ separated}] = 2 \cdot \frac{\theta}{2\pi} = \frac{\theta}{\pi}$
- Expected contribution to APX is $w_{ij} \cdot \frac{\theta}{\pi}$



$(i,j) \in \delta(S)$:



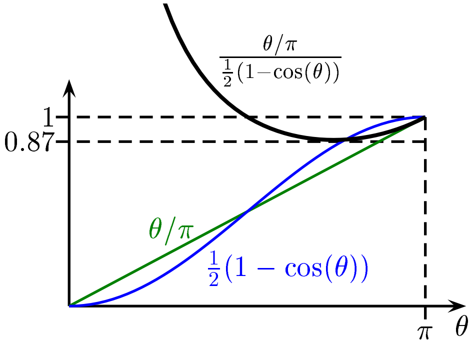
$(i,j) \notin \delta(S)$:



Proof (2)

- Expected contribution of edge (i, j) to APX is $w_{ij} \cdot \frac{\theta}{\pi}$
- Contribution of edge (i, j) to SDP is $w_{ij} \cdot \frac{1}{2}(1 - \cos(\theta))$

$$\frac{E[APX]}{SDP} \geq \min_{0 \leq \theta \leq \pi} \frac{\theta/\pi}{\frac{1}{2}(1 - \cos(\theta))} \approx 0.878. \quad \square$$



State of the art

Theorem (Khot, Kindler, Mossel, O'Donnell '05)

There is no polynomial time < 1.138*-approximation algorithm (unless the Unique Games Conjecture is false).*

- That means the presented approximation is the best possible.

PART 25
MAX2SAT

SOURCE: *Approximation Algorithms* (Vazirani, Springer Press)

Problem definition

Problem: MAX2SAT

- Given: SAT formula $\bigwedge_{C \in \mathcal{C}} C$ on variables $x_1, \ldots, x_n$. Each clause C contains at most 2 literals.
- Find: Truth assignment maximizing the number of satisfied clauses

$$OPT = \max_{a=(a_1,\ldots,a_n)\in\{0,1\}^n} \left|\{C \in \mathcal{C} \mid C \text{ true for assignment } a\}\right|$$

- **Example:**

$$\underbrace{(\bar{x}_1 \vee x_2)}_{\text{clause}} \wedge (x_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2) \wedge (x_1 \vee x_2) \wedge \bar{x}_1$$

  Optimal assignment: $a = (0, 1)$ with 4 satisfied clauses.
- **Remark:** Problem is **NP**-hard though testing wether *all* clauses can be satisfied is easy.

A quadratic program

- **Goal:** Write MAX2SAT as quadratic program

$$\max \sum_{i,j} a_{ij}(1 + y_i y_j) + b_{ij}(1 - y_i y_j)$$

$$y_i^2 = 1$$

$$y_i \in \mathbb{Z}$$

  for suitable coefficients a_{ij}, b_{ij}.
- Here $y_i = 1 \equiv x_i$ true, $y_i = -1 \equiv x_i$ false
- Let $y_0 := 1$ be auxiliary variable.
- Write

$$v(C) = \begin{cases} 1 & \text{if clause } C \text{ true for } y \\ 0 & \text{otherwise} \end{cases}$$

- For clauses with 1 literal

$$v(x_i) = \frac{1 + y_0 y_i}{2}, v(\bar{x}_i) = \frac{1 - y_0 y_i}{2}$$

A quadratic program (2)

- For clause $x_i \vee x_j$

$$\begin{aligned} v(x_i \vee x_j) &= 1 - v(\bar{x}_i) \cdot v(\bar{x}_j) = 1 - \frac{1 - y_0 y_i}{2} \cdot \frac{1 - y_0 y_j}{2} \\ &= \frac{1}{4}(3 + y_0 y_i + y_0 y_j - \overbrace{y_0^2}^{=1} y_i y_j) \\ &= \frac{1 + y_0 y_i}{4} + \frac{1 + y_0 y_j}{4} + \frac{1 - y_i y_j}{4} \end{aligned}$$

- Similar for $\bar{x}_i \vee x_j$ and $\bar{x}_i \vee \bar{x}_j$.
- We obtain promised coefficients a_{ij}, b_{ij} by summing up $\sum_{C \in \mathcal{C}} v(C)$.
- Now: Relax the quadratic program to a (solvable) vector program.

The algorithm

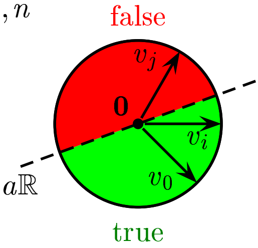
Algorithm:

(1) Solve MaxCut vector program

$$\max \sum_{0 \leq i < j \leq n} \Big(a_{ij}(1 + v_i v_j) + b_{ij}(1 - v_i v_j) \Big)$$
$$v_i^2 = 1 \ \forall i = 0, \ldots, n$$
$$v_i \in \mathbb{R}^{n+1}$$

(2) Choose randomly a vector r from n-dimensional unit ball

(3) Let $y_i := 1$ for all i that are on the same side of the hyperplane $x \cdot r = 0$ as v_0 (the "truth" vector)



Theorem

Let APX := #satisfied clauses. Then $E[APX] \geq 0.87 \cdot SDP$.

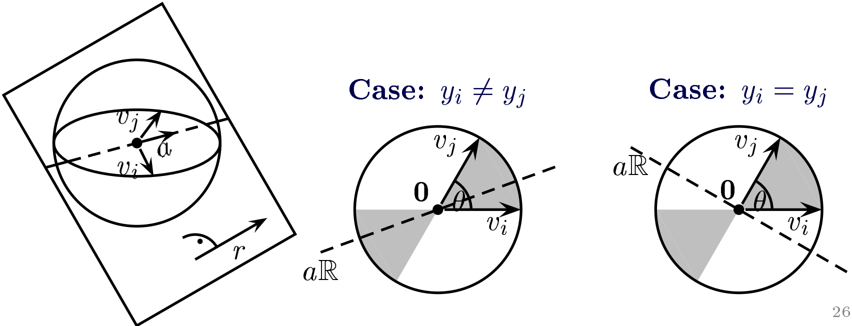
Analysis

Case: Term $b_{ij}(1 - v_i v_j)$ with angle θ between v_i, v_j

- Contribution to $E[APX]$: $2b_{ij} \cdot \Pr[y_i \neq y_j] = 2b_{ij}\frac{\theta}{\pi}$
- Contribution to Vector program: $b_{ij}(1 - \cos(\theta))$
- Gap: $\min_{0 \leq \theta \leq \pi} \frac{2\theta/\pi}{1-\cos(\theta))} \approx 0.878$

Case: Term $a_{ij}(1 + v_i v_j)$ with angle θ between v_i, v_j

- Contribution to $E[APX]$: $2a_{ij} \cdot \Pr[y_i = y_j] = 2a_{ij}(1 - \frac{\theta}{\pi})$
- Contribution to Vector program: $a_{ij}(1 + \cos(\theta))$
- Gap: $\min_{0 \leq \theta \leq \pi} \frac{2(1-\theta/\pi)}{1+\cos(\theta))} \approx 0.878$



State of the art

Theorem (Feige, Goemans '95)

There is a 1.0741*-apx for* MAX2SAT.

Theorem (Lewin, Livnat, Zwick '02)

There is a 1.064*-apx for* MAX2SAT.

Theorem (Hastad '97)

There is no 1.0476*-apx for* MAX2SAT *(unless* $\mathbf{NP} = \mathbf{P}$*).*

Theorem (Khot, Kindler, Mossel, O'Donnell '05)

There is no polynomial time 1.063*-apx for* MAX2SAT *(unless the Unique Games Conjecture is false).*