Semidefinite Programming (SDP) and the Goemans-Williamson MAXCUT Paper

Robert M. Freund

September 8, 2003

This presentation is based on: Goemans, Michel X., and David P. Williamson. *Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. Journal of the ACM* 42(6), November 1995, pp. 1115-1145.

Outline

- Alternate View of Linear Programming
- Facts about Symmetric and Semidefinite Matrices
- SDP
- SDP Duality
- Approximately Solving MAXCUT using SDP and Random Vectors
- Interior-Point Methods for SDP

Alternative Perspective

$$LP: \text{minimize } \boldsymbol{c \cdot x}$$

$$\text{s.t.} \quad \boldsymbol{a_i \cdot x = b_i, \;\; i = 1, \ldots, m}$$

$$\boldsymbol{x} \in \Re^n_+.$$

"$\boldsymbol{c \cdot x}$" means the linear function "$\sum_{j=1}^{n} \boldsymbol{c_j x_j}$"

$\Re^n_+ := \{\boldsymbol{x} \in \Re^n \mid \boldsymbol{x} \geq \boldsymbol{0}\}$ is the nonnegative orthant.

$\Re^n_+$ is a *convex cone*.

$\boldsymbol{K}$ is convex cone if $\boldsymbol{x, w} \in \boldsymbol{K}$ and $\boldsymbol{\alpha, \beta} \geq \boldsymbol{0} \;\; \Rightarrow \;\; \boldsymbol{\alpha x + \beta w} \in \boldsymbol{K}$.

Linear Programming

Alternative Perspective

$$LP: \text{minimize } \ c \cdot x$$

$$\text{s.t.} \quad a_i \cdot x = b_i, \ \ i = 1, \ldots, m$$

$$x \in \Re^n_+.$$

"Minimize the linear function $c \cdot x$, subject to the condition that x must solve m given equations $a_i \cdot x = b_i, i = 1, \ldots, m$, and that x must lie in the convex cone $K = \Re^n_+$."

LP Dual Problem...

$$\begin{array}{lll} LD: \text{maximize} & \displaystyle\sum_{i=1}^{m} y_i b_i \\ \text{s.t.} & \displaystyle\sum_{i=1}^{m} y_i a_i + s = c \\ & s \in \Re^n_+ . \end{array}$$

For feasible solutions x of LP and (y, s) of LD, the duality gap is simply

$$c \cdot x - \sum_{i=1}^{m} y_i b_i = \left(c - \sum_{i=1}^{m} y_i a_i \right) \cdot x = s \cdot x \geq 0$$

Linear Programming

Alternative Perspective

...LP Dual Problem

If LP and LD are feasible, then there exists $\boldsymbol{x}^*$ and $(\boldsymbol{y}^*, \boldsymbol{s}^*)$ feasible for the primal and dual, respectively, for which

$$\boldsymbol{c} \cdot \boldsymbol{x}^* - \sum_{i=1}^{m} \boldsymbol{y}_i^* \boldsymbol{b}_i = \boldsymbol{s}^* \cdot \boldsymbol{x}^* = \boldsymbol{0}$$

Facts about the Semidefinite Cone

If $\boldsymbol{X}$ is an $\boldsymbol{n} \times \boldsymbol{n}$ matrix, then $\boldsymbol{X}$ is a symmetric positive semidefinite (SPSD) matrix if $\boldsymbol{X} = \boldsymbol{X}^{\boldsymbol{T}}$ and

$$\boldsymbol{v}^{\boldsymbol{T}} \boldsymbol{X} \boldsymbol{v} \geq \boldsymbol{0} \quad \text{for any} \quad \boldsymbol{v} \in \Re^{\boldsymbol{n}}$$

If $\boldsymbol{X}$ is an $\boldsymbol{n} \times \boldsymbol{n}$ matrix, then $\boldsymbol{X}$ is a symmetric positive definite (SPD) matrix if $\boldsymbol{X} = \boldsymbol{X}^{\boldsymbol{T}}$ and

$$\boldsymbol{v}^{\boldsymbol{T}} \boldsymbol{X} \boldsymbol{v} > \boldsymbol{0} \quad \text{for any} \quad \boldsymbol{v} \in \Re^{\boldsymbol{n}}, \boldsymbol{v} \neq \boldsymbol{0}$$

Facts about the Semidefinite Cone

S^n denotes the set of symmetric $n \times n$ matrices

S^n_+ denotes the set of (SPSD) $n \times n$ matrices.

S^n_{++} denotes the set of (SPD) $n \times n$ matrices.

Facts about the Semidefinite Cone

Let $\boldsymbol{X}, \boldsymbol{Y} \in \boldsymbol{S}^n$.

"$\boldsymbol{X} \succeq \mathbf{0}$" denotes that $\boldsymbol{X}$ is SPSD

"$\boldsymbol{X} \succeq \boldsymbol{Y}$" denotes that $\boldsymbol{X} - \boldsymbol{Y} \succeq \mathbf{0}$

"$\boldsymbol{X} \succ \mathbf{0}$" to denote that $\boldsymbol{X}$ is SPD, etc.

Remark: $\boldsymbol{S}^n_+ = \{\boldsymbol{X} \in \boldsymbol{S}^n \mid \boldsymbol{X} \succeq \mathbf{0}\}$ is a convex cone.

Facts about Eigenvalues and Eigenvectors

If M is a square $n \times n$ matrix, then λ is an eigenvalue of M with corresponding eigenvector q if

$$Mq = \lambda q \text{ and } q \neq 0 \ .$$

Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ enumerate the eigenvalues of M.

Facts about Eigenvalues and Eigenvectors

The corresponding eigenvectors $q^1, q^2, \ldots, q^n$ of M can be chosen so that they are orthonormal, namely

$$\left(q^i\right)^T\left(q^j\right) = 0 \text{ for } i \neq j, \quad \text{and} \quad \left(q^i\right)^T\left(q^i\right) = 1$$

Define:

$$Q := \left[q^1 \; q^2 \; \cdots \; q^n\right]$$

Then Q is an *orthonormal* matrix:

$$Q^T Q = I, \quad \text{equivalently} \quad Q^T = Q^{-1}$$

Facts about Eigenvalues and Eigenvectors

$\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of M

$q^1, q^2, \ldots, q^n$ are the corresponding orthonormal eigenvectors of M

$$Q := \left[q^1 \; q^2 \; \cdots \; q^n \right]$$

$$Q^T Q = I, \text{ equivalently } Q^T = Q^{-1}$$

Define D:

$$D := \begin{pmatrix} \lambda_1 & 0 & & 0 \\ 0 & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{pmatrix} .$$

Property: $M = QDQ^T$.

Facts about Eigenvalues and Eigenvectors

The decomposition of M into $M = QDQ^T$ is called its *eigendecomposition*.

Facts about Symmetric Matrices

- If $\boldsymbol{X} \in \boldsymbol{S}^{\boldsymbol{n}}$, then $\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^T$ for some orthonormal matrix $\boldsymbol{Q}$ and some diagonal matrix $\boldsymbol{D}$. The columns of $\boldsymbol{Q}$ form a set of $\boldsymbol{n}$ orthogonal eigenvectors of $\boldsymbol{X}$, whose eigenvalues are the corresponding entries of the diagonal matrix $\boldsymbol{D}$.
- $\boldsymbol{X} \succeq \mathbf{0}$ if and only if $\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^T$ where the eigenvalues (i.e., the diagonal entries of $\boldsymbol{D}$) are all nonnegative.
- $\boldsymbol{X} \succ \mathbf{0}$ if and only if $\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^T$ where the eigenvalues (i.e., the diagonal entries of $\boldsymbol{D}$) are all positive.

Facts about Symmetric Matrices

- If M is symmetric, then

$$\det(M) = \prod_{j=1}^{n} \lambda_j$$

Facts about Symmetric Matrices

- Consider the matrix M defined as follows:

$$M = \begin{pmatrix} P & v \\ v^T & d \end{pmatrix},$$

where $P \succ 0$, v is a vector, and d is a scalar. Then $M \succeq 0$ if and only if $d - v^T P^{-1} v \geq 0$.

- For a given column vector a, the matrix $X := aa^T$ is SPSD, i.e., $X = aa^T \succeq 0$.

- If $M \succeq 0$, then there is a matrix N for which $M = N^T N$. To see this, simply take $N = D^{\frac{1}{2}} Q^T$.

SDP

Think about X

Let $X \in S^n$. Think of X as:

- a matrix
- an array of n^2 components of the form $(x_{11}, \ldots, x_{nn})$
- an object (a vector) in the space S^n.

All three different equivalent ways of looking at X will be useful.

SDP

Linear Function of X

Let $X \in S^n$. What will a linear function of X look like?

If $C(X)$ is a linear function of X, then $C(X)$ can be written as $C \bullet X$, where

$$C \bullet X := \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij}.$$

There is no loss of generality in assuming that the matrix C is also symmetric.

$$SDP: \text{minimize } C \bullet X$$

$$\text{s.t.} \quad A_i \bullet X = b_i \ , i = 1, \ldots, m,$$

$$X \succeq 0,$$

"$X \succeq 0$" is the same as "$X \in S^n_+$"

The data for SDP consists of the symmetric matrix C (which is the data for the objective function) and the m symmetric matrices $A_1, \ldots, A_m$, and the $m-$vector b, which form the m linear equations.

SDP

Example...

$$A_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 3 & 7 \\ 1 & 7 & 5 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 2 & 8 \\ 2 & 6 & 0 \\ 8 & 0 & 4 \end{pmatrix}, \; b = \begin{pmatrix} 11 \\ 19 \end{pmatrix}, \text{ and } C = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 9 & 0 \\ 3 & 0 & 7 \end{pmatrix},$$

The variable X will be the 3×3 symmetric matrix:

$$X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix},$$

$$\begin{array}{llrcl} SDP: & \text{minimize} & x_{11} + 4x_{12} + 6x_{13} + 9x_{22} + 0x_{23} + 7x_{33} & & \\ & \text{s.t.} & x_{11} + 0x_{12} + 2x_{13} + 3x_{22} + 14x_{23} + 5x_{33} & = & 11 \\ & & 0x_{11} + 4x_{12} + 16x_{13} + 6x_{22} + 0x_{23} + 4x_{33} & = & 19 \end{array}$$

$$X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} \succeq 0.$$

SDP

...Example

$$\begin{array}{rrcl} SDP: \text{ minimize} & x_{11} + 4x_{12} + 6x_{13} + 9x_{22} + 0x_{23} + 7x_{33} & & \\ \text{s.t.} & x_{11} + 0x_{12} + 2x_{13} + 3x_{22} + 14x_{23} + 5x_{33} & = & 11 \\ & 0x_{11} + 4x_{12} + 16x_{13} + 6x_{22} + 0x_{23} + 4x_{33} & = & 19 \end{array}$$

$$X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} \succeq 0.$$

It may be helpful to think of "$X \succeq 0$" as stating that each of the n *eigenvalues* of X must be nonnegative.

SDP

Semidefinite Programming

$LP \subset SDP$

$$
\begin{array}{lll}
LP: & \text{minimize} & c \cdot x \\
 & \text{s.t.} & a_i \cdot x = b_i, \;\; i = 1, \ldots, m \\
 & & x \in \Re^n_+ .
\end{array}
$$

Define:

$$
A_i = \begin{pmatrix} a_{i1} & 0 & \ldots & 0 \\ 0 & a_{i2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & a_{in} \end{pmatrix}, \;\; i = 1, \ldots, m, \;\; \text{and} \;\; C = \begin{pmatrix} c_1 & 0 & \ldots & 0 \\ 0 & c_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & c_n \end{pmatrix}.
$$

$$
\begin{array}{lll}
SDP: & \text{minimize} & C \bullet X \\
 & \text{s.t.} & A_i \bullet X = b_i \;\; , i = 1, \ldots, m, \\
 & & X_{ij} = 0, \;\; i = 1, \ldots, n, \;\; j = i+1, \ldots, n, \\
 & & X = \begin{pmatrix} x_1 & 0 & \ldots & 0 \\ 0 & x_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & x_n \end{pmatrix} \succeq 0,
\end{array}
$$

SDP Duality

$$SDD: \text{maximize} \sum_{i=1}^{m} y_i b_i$$

s.t. $$\sum_{i=1}^{m} y_i A_i + S = C$$

$$S \succeq 0.$$

Notice

$$S = C - \sum_{i=1}^{m} y_i A_i \succeq 0$$

SDP Duality

and so equivalently:

$$\boldsymbol{SDD}: \ \textbf{maximize} \ \sum_{i=1}^{m} y_i b_i$$

$$\text{s.t.} \qquad C - \sum_{i=1}^{m} y_i A_i \succeq 0$$

SDP Duality

Example

$$A_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 3 & 7 \\ 1 & 7 & 5 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 2 & 8 \\ 2 & 6 & 0 \\ 8 & 0 & 4 \end{pmatrix}, \ b = \begin{pmatrix} 11 \\ 19 \end{pmatrix}, \ \text{and} \ C = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 9 & 0 \\ 3 & 0 & 7 \end{pmatrix}$$

$$SDD: \ \text{maximize} \ 11y_1 + 19y_2$$

s.t.

$$y_1 \begin{pmatrix} 1 & 0 & 1 \\ 0 & 3 & 7 \\ 1 & 7 & 5 \end{pmatrix} + y_2 \begin{pmatrix} 0 & 2 & 8 \\ 2 & 6 & 0 \\ 8 & 0 & 4 \end{pmatrix} + S = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 9 & 0 \\ 3 & 0 & 7 \end{pmatrix}$$

$$S \succeq 0$$

SDP Duality

Example

$$SDD: \text{ maximize } \; 11y_1 + 19y_2$$

s.t.

$$y_1 \begin{pmatrix} 1 & 0 & 1 \\ 0 & 3 & 7 \\ 1 & 7 & 5 \end{pmatrix} + y_2 \begin{pmatrix} 0 & 2 & 8 \\ 2 & 6 & 0 \\ 8 & 0 & 4 \end{pmatrix} + S = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 9 & 0 \\ 3 & 0 & 7 \end{pmatrix}$$

$$S \succeq 0$$

is the same as:

$$SDD: \text{ maximize } \quad 11y_1 + 19y_2$$

s.t.

$$\begin{pmatrix} 1 - 1y_1 - 0y_2 & 2 - 0y_1 - 2y_2 & 3 - 1y_1 - 8y_2 \\ 2 - 0y_1 - 2y_2 & 9 - 3y_1 - 6y_2 & 0 - 7y_1 - 0y_2 \\ 3 - 1y_1 - 8y_2 & 0 - 7y_1 - 0y_2 & 7 - 5y_1 - 4y_2 \end{pmatrix} \succeq 0.$$

Weak Duality

Weak Duality Theorem: Given a feasible solution X of SDP and a feasible solution (y, S) of SDD, the duality gap is

$$C \bullet X - \sum_{i=1}^{m} y_i b_i = S \bullet X \geq 0 \; .$$

If

$$C \bullet X - \sum_{i=1}^{m} y_i b_i = 0 \; ,$$

then X and (y, S) are each optimal solutions to SDP and SDD, respectively, and furthermore, $SX = 0$.

SDP Duality

Strong Duality Theorem: Let z_P^* and z_D^* denote the optimal objective function values of SDP and SDD, respectively. Suppose that there exists a feasible solution $\hat{X}$ of SDP such that $\hat{X} \succ 0$, and that there exists a feasible solution $(\hat{y}, \hat{S})$ of SDD such that $\hat{S} \succ 0$. Then both SDP and SDD attain their optimal values, and

$$z_P^* = z_D^* \, .$$

Some Important Weaknesses of SDP

- There may be a finite or infinite duality gap.
- The primal and/or dual may or may not attain their optima.
- Both programs will attain their common optimal value if both programs have feasible solutions that are SPD.
- There is no finite algorithm for solving SDP.
- There is a simplex algorithm, but it is not a finite algorithm. There is no direct analog of a "basic feasible solution" for SDP.

The MAX CUT Problem

M. Goemans and D. Williamson, *Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems using Semidefinite Programming,* J. ACM 42 1115-1145, 1995.

The MAX CUT Problem

G is an undirected graph with nodes $N = \{1, \ldots, n\}$ and edge set E.

Let $w_{ij} = w_{ji}$ be the weight on edge (i, j), for $(i, j) \in E$.

We assume that $w_{ij} \geq 0$ for all $(i, j) \in E$.

The MAX CUT problem is to determine a subset S of the nodes N for which the sum of the weights of the edges that cross from S to its complement $\bar{S}$ is maximized ($\bar{S} := N \setminus S$).

The MAX CUT Problem

The MAX CUT problem is to determine a subset S of the nodes N for which the sum of the weights w_{ij} of the edges that cross from S to its complement $\bar{S}$ is maximized ($\bar{S} := N \setminus S$).

Let $x_j = 1$ for $j \in S$ and $x_j = -1$ for $j \in \bar{S}$.

$$MAXCUT: \text{maximize}_x \ \frac{1}{4} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}(1 - x_i x_j)$$

$$\text{s.t.} \qquad x_j \in \{-1, 1\}, \ \ j = 1, \ldots, n.$$

The MAX CUT Problem

Formulations

$$MAXCUT: \text{maximize}_x \ \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(1 - x_i x_j)$$

s.t. $\quad x_j \in \{-1, 1\}, \ \ j = 1, \ldots, n.$

Let

$$Y = xx^T \ .$$

Then

$$Y_{ij} = x_i x_j \quad i = 1, \ldots, n, \ \ j = 1, \ldots, n.$$

The MAX CUT Problem

Also let W be the matrix whose $(i,j)^{\mathrm{th}}$ element is w_{ij} for $i = 1, \ldots, n$ and $j = 1, \ldots, n$. Then

$$MAXCUT: \text{maximize}_{Y,x} \ \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\left(1 - Y_{ij}\right)$$

$$\text{s.t.} \qquad x_j \in \{-1, 1\}, \quad j = 1, \ldots, n$$

$$Y = xx^T.$$

The MAX CUT Problem

Formulations

$$MAXCUT: \text{maximize}_{Y,x} \ \frac{1}{4} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (1 - Y_{ij})$$

s.t. $$x_j \in \{-1, 1\}, \ \ j = 1, \ldots, n$$

$$Y = xx^T.$$

The MAX CUT Problem

Formulations

The first set of constraints are equivalent to $Y_{jj} = 1, j = 1, \ldots, n$.

$$MAXCUT: \text{maximize}_{Y,x} \ \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\,(1 - Y_{ij})$$

$$\text{s.t.} \qquad Y_{jj} = 1, \ \ j = 1, \ldots, n$$

$$Y = xx^T.$$

The MAX CUT Problem

Formulations

$$MAXCUT: \text{maximize}_{Y,x} \; \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\,(1 - Y_{ij})$$

$$\text{s.t.} \qquad Y_{jj} = 1, \;\; j = 1, \ldots, n$$

$$Y = xx^T.$$

Notice that the matrix $Y = xx^T$ is a rank-1 SPSD matrix.

The MAX CUT Problem

We *relax* this condition by removing the rank-1 restriction:

$$\boldsymbol{RELAX:}\ \ \mathbf{maximize}_{\boldsymbol{Y}}\ \ \tfrac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\,(1 - Y_{ij})$$

$$\text{s.t.}\qquad Y_{jj} = 1,\ \ j = 1, \ldots, n$$

$$Y \succeq 0.$$

It is therefore easy to see that RELAX provides an upper bound on MAXCUT, i.e.,

$$MAXCUT \le RELAX.$$

The MAX CUT Problem

Computing a Good Solution

$$RELAX: \text{maximize}_Y \ \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\,(1 - Y_{ij})$$

s.t. $$Y_{jj} = 1, \ \ j = 1, \ldots, n$$

$$Y \succeq 0.$$

Let $\hat{Y}$ solve $RELAX$

Factorize $\hat{Y} = \hat{V}^T \hat{V}$

$\hat{V} = [\hat{v}_1 \ \hat{v}_2 \ \cdots \ \hat{v}_n]$ and $\hat{Y}_{ij} = \left(\hat{V}^T \hat{V}\right)_{ij} = \hat{v}_i^T \hat{v}_j$

Let $\hat{Y}$ solve $RELAX$

Factorize $\hat{Y} = \hat{V}^T \hat{V}$

$\hat{V} = [\hat{v}_1\ \hat{v}_2\ \cdots\ \hat{v}_n]$ and $\hat{Y}_{ij} = \left(\hat{V}^T \hat{V}\right)_{ij} = \hat{v}_i^T \hat{v}_j$

Let r be a random uniform vector on the unit n-sphere S^n

$S := \{i \mid r^T \hat{v}_i \geq 0\}$

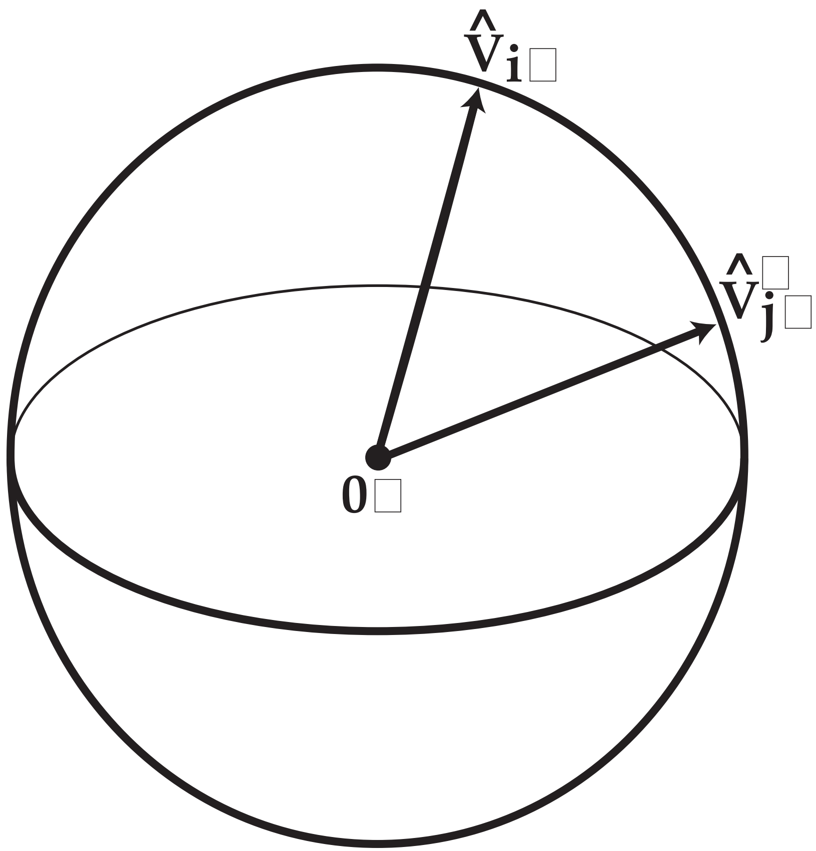
$\overline{S} := \{i \mid r^T \hat{v}_i < 0\}$

Proposition:

$$P\left(\text{sign}(\boldsymbol{r}^T\hat{\boldsymbol{v}}_i) \neq \text{sign}(\boldsymbol{r}^T\hat{\boldsymbol{v}}_j)\right) = \frac{\mathbf{arccos}(\hat{\boldsymbol{v}}_i^T\hat{\boldsymbol{v}}_j)}{\boldsymbol{\pi}} .$$

The MAX CUT Problem

Computing a Good Solution



Let r be a random uniform vector on the unit n-sphere S^n

$S := \{i \mid r^T \hat{v}_i \geq 0\}$

$\overline{S} := \{i \mid r^T \hat{v}_i < 0\}$

Let $E[\text{Cut}]$ denote the expected value of this cut.

Theorem: $E[\text{Cut}] \geq 0.87856 \times MAXCUT$

The MAX CUT Problem

Computing a Good Solution

$$
\begin{aligned}
E[\text{Cut}] &= \frac{1}{2}\sum_{i,j} w_{ij} \times P\left(\text{sign}(r^T \hat{v}_i) \neq \text{sign}(r^T \hat{v}_j)\right) \\
&= \frac{1}{2}\sum_{i,j} w_{ij} \frac{\arccos(\hat{v}_i^T \hat{v}_j)}{\pi} \\
&= \frac{1}{2}\sum_{i,j} w_{ij} \frac{\arccos(\hat{Y}_{ij})}{\pi} \\
&= \frac{1}{2\pi}\sum_{i,j} w_{ij} \arccos(\hat{Y}_{ij})
\end{aligned}
$$

The MAX CUT Problem

Computing a Good Solution

$$
\begin{aligned}
E[\text{Cut}] &= \frac{1}{2\pi}\sum_{i,j} w_{ij}\arccos(\hat{Y}_{ij}) \\
&= \frac{1}{4}\sum_{i,j} w_{ij}\left(1-\hat{Y}_{ij}\right)\frac{2}{\pi}\frac{\arccos(\hat{Y}_{ij})}{1-\hat{Y}_{ij}} \\
&\geq \frac{1}{4}\sum_{i,j} w_{ij}\left(1-\hat{Y}_{ij}\right)\min_{-1\leq t\leq 1}\frac{2}{\pi}\frac{\arccos(t)}{1-t} \\
&= RELAX \times \min_{0\leq\theta\leq\pi}\frac{2}{\pi}\frac{\theta}{1-\cos\theta} \\
&\geq RELAX \times 0.87856
\end{aligned}
$$

The MAX CUT Problem

Computing a Good Solution

So we have

$$\boldsymbol{MAXCUT} \geq \boldsymbol{E}[\text{Cut}] \geq \boldsymbol{RELAX} \times \mathbf{0.87856} \geq \boldsymbol{MAXCUT} \times \mathbf{0.87856}$$

This is an impressive result, in that it states that the value of the semidefinite relaxation is guaranteed to be no more than $\mathbf{12.2\%}$ higher than the value of $\boldsymbol{NP}$-hard problem MAXCUT.

The Logarithmic Barrier Function for SPD Matrices

Let $X \succeq 0$, equivalently $X \in S^n_+$.

X will have n nonnegative eigenvalues, say $\lambda_1(X), \ldots, \lambda_n(X) \geq 0$ (possibly counting multiplicities).

$$\partial S^n_+ = \{X \in S^n \mid \lambda_j(X) \geq 0, j = 1, \ldots, n, \text{ and } \lambda_j(X) = 0 \text{ for some } j \in \{1, \ldots, n\}\}.$$

The Logarithmic Barrier Function for SPD Matrices

$$\partial S^n_+ = \{X \in S^n \mid \lambda_j(X) \geq 0, j = 1, \ldots, n,$$
$$\text{and } \lambda_j(X) = 0 \text{ for some } j \in \{1, \ldots, n\}\}.$$

A natural barrier function is:

$$B(X) := -\sum_{j=1}^{n} \ln(\lambda_i(X)) = -\ln\left(\prod_{j=1}^{n} \lambda_i(X)\right) = -\ln(\det(X)).$$

This function is called the log-determinant function or the logarithmic barrier function for the semidefinite cone.

The Logarithmic Barrier Function for SPD Matrices

$$B(X) := -\sum_{j=1}^{n} \ln(\lambda_i(X)) = -\ln\left(\prod_{j=1}^{n} \lambda_i(X)\right) = -\ln(\det(X)).$$

Quadratic Taylor expansion at $X = \bar{X}$:

$$B(\bar{X} + \alpha D) \approx B(\bar{X}) + \alpha \bar{X}^{-1} \bullet D + \frac{1}{2}\alpha^2 \left(\bar{X}^{-\frac{1}{2}} D \bar{X}^{-\frac{1}{2}}\right) \bullet \left(\bar{X}^{-\frac{1}{2}} D \bar{X}^{-\frac{1}{2}}\right) .$$

$B(X)$ has the same remarkable properties in the context of interior-point methods for SDP as the barrier function $-\sum_{j=1}^{n} \ln(x_j)$ does in the context of linear optimization.

Primal and Dual SDP

$$\begin{array}{lll} SDP: & \text{minimize} & C \bullet X \\ & \text{s.t.} & A_i \bullet X = b_i \ \ , i = 1, \ldots, m, \\ & & X \succeq 0 \end{array}$$

and

$$\begin{array}{lll} SDD: & \text{maximize} & \sum_{i=1}^{m} y_i b_i \\ & \text{s.t.} & \sum_{i=1}^{m} y_i A_i + S = C \\ & & S \succeq 0 \ . \end{array}$$

If X and (y, S) are feasible for the primal and the dual, the duality gap is:

$$C \bullet X - \sum_{i=1}^{m} y_i b_i = S \bullet X \geq 0 \ .$$

Also,

$$S \bullet X = 0 \iff SX = 0 \ .$$

$$B(X) = -\sum_{j=1}^{n} \ln(\lambda_i(X)) = -\ln\left(\prod_{j=1}^{n} \lambda_i(X)\right) = -\ln(\det(X)) \; .$$

Consider:

$$BSDP(\mu): \text{ minimize } \; C \bullet X - \mu \ln(\det(X))$$

$$\text{s.t.} \qquad A_i \bullet X = b_i \;\; , i = 1, \ldots, m,$$

$$X \succ 0.$$

Let $f_\mu(X)$ denote the objective function of $BSDP(\mu)$. Then:

$$-\nabla f_\mu(X) = C - \mu X^{-1}$$

Primal and Dual SDP

$$BSDP(\mu): \text{ minimize } C \bullet X - \mu \ln(\det(X))$$

$$\text{s.t.} \quad A_i \bullet X = b_i \ , i = 1, \ldots, m,$$

$$X \succ 0.$$

$$\nabla f_\mu(X) = C - \mu X^{-1}$$

Karush-Kuhn-Tucker conditions for $BSDP(\mu)$ are:

$$\begin{cases} A_i \bullet X = b_i \ , i = 1, \ldots, m, \\ X \succ 0, \\ C - \mu X^{-1} = \sum_{i=1}^{m} y_i A_i. \end{cases}$$

$$\begin{cases} A_i \bullet X = b_i \quad , i = 1, \ldots, m, \\ X \succ 0, \\ C - \mu X^{-1} = \sum_{i=1}^{m} y_i A_i. \end{cases}$$

Define

$$S = \mu X^{-1} \ ,$$

which implies

$$XS = \mu I \ ,$$

and rewrite KKT conditions as:

$$\begin{cases} A_i \bullet X = b_i \ \ , i = 1, \ldots, m, \quad X \succ 0 \\ \sum_{i=1}^{m} y_i A_i + S = C \\ XS = \mu I. \end{cases}$$

$$\begin{cases} A_i \bullet X = b_i \;\; , i = 1, \ldots, m, \quad X \succ 0 \\ \sum_{i=1}^{m} y_i A_i + S = C \\ XS = \mu I. \end{cases}$$

If (X, y, S) is a solution of this system, then X is feasible for SDP, (y, S) is feasible for SDD, and the resulting duality gap is

$$S \bullet X = \sum_{i=1}^{n} \sum_{j=1}^{n} S_{ij} X_{ij} = \sum_{j=1}^{n} (SX)_{jj} = \sum_{j=1}^{n} (\mu I)_{jj} = n\mu.$$

Primal and Dual SDP

$$\begin{cases} A_i \bullet X = b_i \ \ , i = 1, \ldots, m, \ \ X \succ 0 \\ \sum_{i=1}^{m} y_i A_i + S = C \\ XS = \mu I. \end{cases}$$

If (X, y, S) is a solution of this system, then X is feasible for SDP, (y, S) is feasible for SDD, the duality gap is

$$S \bullet X = n\mu.$$

Interior-point Methods for SDP

This suggests that we try solving $BSDP(\mu)$ for a variety of values of μ as $\mu \rightarrow \mathbf{0}$.

Interior-point methods for SDP are very similar to those for linear optimization, in that they use Newton's method to solve the KKT system as $\mu \rightarrow \mathbf{0}$.

Website for SDP

A good website for semidefinite programming is:

`http://www-user.tu-chemnitz.de/ helmberg/semidef.html`.