Introduction to Semidefinite Programming

Konstantinidis Orestis
M.P.L.A.
Network Algorithms

Review of Linear Programming

- LP:

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

Review of Linear Programming

- LP:

minimize	$c \cdot x$	
s.t.	$a_i \cdot x = b_i,$	$i = 1, \ldots, m$
	$x \in \Re^n{}_+$	

LD:

maximize	$\sum_{i=1}^{m} y_i b_i$
s.t.	$\sum_{i=1}^{m} y_i a_i + s = c,$
	$x \in \Re^n{}_+$

Review of Linear Programming

- LP:

minimize	$c \cdot x$
s.t.	$a_i \cdot x = b_i, \qquad i = 1, \dots, m$
	$x \in \Re^n{}_+$

LD:

maximize	$\sum_{i=1}^{m} y_i b_i$
s.t.	$\sum_{i=1}^{m} y_i a_i + s = c,$
	$x \in \Re^n{}_+$

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

Facts about matrices

PSD:

- If X is an $n \times n$ matrix, then X is a positive semidefinite (psd) matrix if

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

Facts about matrices

PSD:

- If X is an $n \times n$ matrix, then X is a positive semidefinite (psd) matrix if

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

PD:

If X is an $n \times n$ matrix, then X is a positive definite (pd) matrix if

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

Facts about matrices

PSD:

- If X is an $n \times n$ matrix, then X is a positive semidefinite (psd) matrix if

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

PD:

If X is an $n \times n$ matrix, then X is a positive definite (pd) matrix if

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

S^n: set of $n \times n$ symmetric matrices.

${S^n}_+$: set of positive semidefinite $n \times n$ symmetric matrices. $X \succcurlyeq 0$

${S^n}_{++}$: set of positive definite $n \times n$ symmetric matrices. $X \succ 0$

$$X \succcurlyeq Y \Leftrightarrow X - Y \succcurlyeq 0$$

Semidefinite Cone

- Closed Convex Cone:

K is a *closed convex cone* if:

- ❖ $x, w \in K \Longrightarrow \alpha x + \beta w \in K, \quad \forall\, \alpha, \beta \geq 0.$
- ❖ K is a closed set

Semidefinite Cone

- Closed Convex Cone:

K is a *closed convex cone* if:

- ❖ $x, w \in K \Longrightarrow \alpha x + \beta w \in K, \quad \forall\, \alpha, \beta \geq 0.$
- ❖ K is a closed set

Remark 1 :

${S^n}_+ = \{X \in S^n | X \succcurlyeq 0\}$ is a closed convex cone in $\Re^{n^2}$ of dimension $n \times (n+1)/2$.

Proof. Suppose that $X, W \in {S^n}_+$. $\forall\, \alpha, \beta \geq 0$, $\forall v \in \Re^n$:

$$v^T(\alpha \cdot X + \beta \cdot W)v = \alpha \cdot v^T X v + \beta \cdot v^T W v \geq 0,$$

Whereby $\alpha \cdot X + \beta \cdot W \in {S^n}_+$.

Properties of Symmetric Matrices

- $X \in S^n \Longrightarrow X = QDQ^T$
  (Q is orthonormal [$Q^T = Q^{-1}$], D is diagonal)
- The columns of Q form a set of n orthogonal eigenvectors of X, whose eigenvalues are the corresponding diagonal entries of D.

Properties of Symmetric Matrices

- $X \in S^n \Longrightarrow X = QDQ^T$
  (Q is orthonormal $[Q^T = Q^{-1}]$, D is diagonal)
- The columns of Q form a set of n orthogonal eigenvectors of X, whose eigenvalues are the corresponding diagonal entries of D.
- $X \succcurlyeq 0 \Longleftrightarrow X = QDQ^T$, where the eigenvalues are all nonnegative.
- $X \succ 0 \Longleftrightarrow X = QDQ^T$, where the eigenvalues are all positive.

Properties of Symmetric Matrices

- $X \in S^n \Longrightarrow X = QDQ^T$
  (Q is orthonormal $[Q^T = Q^{-1}]$, D is diagonal)
- The columns of Q form a set of n orthogonal eigenvectors of X, whose eigenvalues are the corresponding diagonal entries of D.
- $X \succcurlyeq 0 \Leftrightarrow X = QDQ^T$, where the eigenvalues are all nonnegative.
- $X \succ 0 \Leftrightarrow X = QDQ^T$, where the eigenvalues are all positive.
- $[(X \succcurlyeq 0) \wedge (X_{ii} = 0)] \Longrightarrow X_{ij} = X_{ji} = 0, \quad \forall j = 1, \dots, n.$

Properties of Symmetric Matrices

- $X \in S^n \Longrightarrow X = QDQ^T$
  (Q is orthonormal $[Q^T = Q^{-1}]$, D is diagonal)
- The columns of Q form a set of n orthogonal eigenvectors of X, whose eigenvalues are the corresponding diagonal entries of D.
- $X \succcurlyeq 0 \Leftrightarrow X = QDQ^T$, where the eigenvalues are all nonnegative.
- $X \succ 0 \Leftrightarrow X = QDQ^T$, where the eigenvalues are all positive.
- $[(X \succcurlyeq 0) \wedge (X_{ii} = 0)] \Longrightarrow X_{ij} = X_{ji} = 0, \quad \forall j = 1, \dots, n.$
- Matrix M defined as follows:
  - Where $P \succ 0$, v is a vector and d is a scalar.
  - Then $M \succ 0 \Leftrightarrow d - v^T P^{-1} v > 0$.

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

We can think of X as….

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

We can think of X as….

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

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

Linear Function of X

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

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

Linear Function of X

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

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

If X is a symmetric matrix, w.l.o.g. matrix C is also symmetric.

Linear Function of X

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

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

If X is a symmetric matrix, w.l.o.g. matrix C is also symmetric.

With this notation,
we are now ready to define a
semidefinite programm

Semidefinite program (SDP)

•

SDP:

minimize $\quad C * X$

s.t. $\quad A_i * X = b_i, \quad i = 1, \ldots, m$

$X \succcurlyeq 0$

An example $[n = 3,\ m = 2]$ (1)

- $$A_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 3 & 7 \\ 1 & 7 & 5 \end{pmatrix}, \qquad A_2 = \begin{pmatrix} 0 & 2 & 8 \\ 2 & 6 & 0 \\ 8 & 0 & 4 \end{pmatrix},$$

$$C = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 9 & 0 \\ 3 & 0 & 7 \end{pmatrix},\ b_1 = 11 \text{ and } b_2 = 19.$$

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

$$C * X = x_{11} + 4x_{12} + 6x_{13} + 9x_{22} + 0x_{23} + 7x_{33}.$$

An example $[n = 3,\ m = 2]$ (2)

- minimize $\quad x_{11} + 4x_{12} + 6x_{13} + 9x_{22} + 0x_{23} + 7x_{33}$

  s.t. $\quad A_i * X = b_i, \quad i = 1, \dots, m$

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

An example $[n = 3,\ m = 2]$ (2)

minimize $\quad x_{11} + 4x_{12} + 6x_{13} + 9x_{22} + 0x_{23} + 7x_{33}$

s.t. $\quad A_i * X = b_i, \quad i = 1, \ldots, m$

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

Notice that SDP looks remarkably similar to a linear program.

LP : Special case of SDP

- Intuitively,

$$[(x \geq 0) \Leftrightarrow (x_i \geq 0)]$$

$$\leftrightsquigarrow$$

$$[(X \succcurlyeq 0) \Leftrightarrow (\text{each of the } n \textit{ eigenvalues} \geq 0)]$$

LP : Special case of SDP

- Intuitively,

$$[(x \geq 0) \Leftrightarrow (x_i \geq 0)]$$

$$\leftrightsquigarrow$$

$$[(X \succcurlyeq 0) \Leftrightarrow (\text{each of the } n \textit{ eigenvalues} \geq 0)]$$

If $A_i = diag(\alpha_{i1}, \ldots, \alpha_{in}), i = 1, \ldots, m$ and $C = diag(c_1, \ldots, c_n)$:

minimize	$C * X$		
s.t.	$A_i * X = b_i,$	$i = 1, \ldots, m$	
	$X_{ij} = 0,$	$i = 1, \ldots, n,$	$j = i + 1, \ldots, n.$
	$X \succcurlyeq 0$		

LP : Special case of SDP

- Intuitively,

  $[(X \succcurlyeq 0) \Leftrightarrow$ each of the n eigenvalues $\geq 0)]$

  $[(x \geq 0) \Leftrightarrow (x_i \geq 0)]$

- If $A_i = diag(\alpha_{i1}, \ldots, \alpha_{in})$, $i = 1, \ldots, m$ and $C = diag(c_1, \ldots, c_n)$:

  minimize $C * X$

  s.t. $A_i * X = b_i, \quad i = 1, \ldots, m$

  $X_{ij} = 0, i = 1, \ldots, n, \quad j = i + 1, \ldots, n.$

  $X \succcurlyeq 0$

In practice, one would never want this kind of conversion

Semidefinite Programming Duality

SDD:

maximize	$\sum_{i=1}^{m} y_i b_i$
s.t.	$\sum_{i=1}^{m} y_i A_i + S = C,$
	$S \succcurlyeq 0$

Semidefinite Programming Duality

- SDD:

maximize	$\sum_{i=1}^{m} y_i b_i$
s.t.	$\sum_{i=1}^{m} y_i A_i + S = C,$
	$S \succcurlyeq 0$

The constraints of SDD state that:

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

must be positive semidefinite. That is,

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

The Dual of the example

maximize $11y_1 + 19y_2$

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

$$S \succcurlyeq 0$$

The Dual of the example

maximize $\quad 11y_1 + 19y_2$

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

$S \succcurlyeq 0$

maximize $\quad 11y_1 + 19y_2$

s.t.

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

Weak Duality

Proposition.

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

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

If $C * X - \sum_{i=1}^{m} y_i b_i = 0$, then X and (y, S) are each optimal solutions to SDP and SDD, respectively, and furthermore,

$$S * X = 0.$$

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 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

$$z_P^* = z_D^*.$$

Key properties not extended

- There may be a finite or infinite duality gap. The primal or dual may or may not attain their optima.

Key properties not extended

- There may be a finite or infinite duality gap. The primal or dual may or may not attain their optima.
- 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.

Key properties not extended

- There may be a finite or infinite duality gap. The primal or dual may or may not attain their optima.
- 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.
- Given rational data, the feasible region may have no rational solutions. The optimal solution may not have rational components or rational eigenvalues.

Key properties not extended

- There may be a finite or infinite duality gap. The primal or dual may or may not attain their optima.
- 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.
- Given rational data, the feasible region may have no rational solutions. The optimal solution may not have rational components or rational eigenvalues.
- Given rational data whose binary encoding is size L, the norms of any feasible and/or optimal solutions may exceed 2^{2^L}(or worse).
- Given rational data whose binary encoding is size L, the norms of any feasible and/or optimal solutions may be less than 2^{-2^L}(or worse).

MAX CUT as Integer Program

- Let G be an undirected graph with nodes $N = \{1, \ldots, n\}$, and edge set E. Let $w_{ij} = w_{ji}$ be the weight on edge $(i, j) \in E$. 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 (where $(\bar{S} := \mathrm{N} \backslash \mathrm{S})$).

MAX CUT as Integer Program

Let G be an undirected graph with nodes $N = \{1, \dots, n\}$, and edge set E. Let $w_{ij} = w_{ji}$ be the weight on edge $(i, j) \in E$. 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 (where $(\bar{S} := \mathrm{N} \backslash \mathrm{S})$).

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

MAX CUT:

maximize $_x$	$\frac{1}{4} \sum_{i=1}^{n} \sum_{i=1}^{n} w_{ij} (1 - x_i x_j)$
s.t.	$x_j \in \{-1, 1\}, \quad \mathrm{j} = 1, \dots, n$

Proper Transformation

- Let $Y = xx^T$, whereby

$$Y_{ij} = x_i x_j, \qquad i = 1, \dots, n, \qquad j = 1, \dots, n$$

Proper Transformation

- Let $Y = xx^T$, whereby

$$Y_{ij} = x_i x_j, \qquad i = 1, \dots, n, \qquad j = 1, \dots, n$$

Then *MAX CUT* can be equivalently formulated as:

maximize $_{Y,x}$	$\frac{1}{4}\sum_{i=1}^{n}\sum_{i=1}^{n} w_{ij} - W * Y$
s.t.	$x_j \in \{-1,1\}, \quad j = 1, \dots, n$
	$Y = xx^T.$

Proper Transformation

- Let $Y = xx^T$, whereby

$$Y_{ij} = x_i x_j, \qquad i = 1, \dots, n, \qquad j = 1, \dots, n$$

Then *MAX CUT* can be equivalently formulated as:

$$\begin{aligned} \text{maximize}_{Y,x} \quad & \frac{1}{4}\sum_{i=1}^{n}\sum_{i=1}^{n} w_{ij} - W * Y \\ \text{s.t.} \quad & Y_{jj} = 1, \qquad j = 1, \dots, n \\ & Y = xx^T. \end{aligned}$$

Relaxation

- The matrix Y is a symmetric rank-1 positive semidefinite matrix. If we relax this condition by removing the rank-1 restriction, we obtain the following relaxation of *MAX CUT*:

RELAX:

maximize $_Y$	$\frac{1}{4}\sum_{i=1}^{n}\sum_{i=1}^{n} w_{ij} - W * Y$	
s.t.	$Y_{jj} = 1,$	$\mathrm{j} = 1, \ldots, n$
	$Y \succcurlyeq 0.$	

.

Relaxation

- The matrix Y is a symmetric rank-1 positive semidefinite matrix. If we relax this condition by removing the rank-1 restriction, we obtain the following relaxation of *MAX CUT*:

RELAX:

maximize $_Y$	$\frac{1}{4}\sum_{i=1}^{n}\sum_{i=1}^{n} w_{ij} - W * Y$	
s.t.	$Y_{jj} = 1,$	$\mathrm{j} = 1, \ldots, n$
	$Y \succcurlyeq 0.$	

.

which is a semidefinite program.

Upper bound

- It is easy to see that *RELAX* provides an upper bound on *MAX CUT*, i.e.

$$MAX\ CUT \leq RELAX.$$

Upper bound

- It is easy to see that *RELAX* provides an upper bound on *MAX CUT*, i.e.

$$MAX\ CUT \leq RELAX.$$

As it turns out, one can also prove:

$$0.87856\ RELAX \leq MAX\ CUT \leq RELAX.$$

This is an impressive result, in that it states that the value of the semidefinite relaxation is guaranteed to be no more than 12% higher than the value of *NP*-hard problem *MAX CUT.*

Applications

- Combinatorial Optimization

Applications

- Combinatorial Optimization
- Convex Optimization

Applications

- Combinatorial Optimization
- Convex Optimization
- Control Theory
  - Interior point methods

KONIEC