CS672: Approximation Algorithms
Spring 2020
Intro to Semidefinite Programming

Instructor: Shaddin Dughmi

Outline

Symmetric Matrices

A matrix $A \in \mathbb{R}^{n \times n}$ is symmetric if and only if it is square and $A_{ij} = A_{ji}$ for all i, j.

- We denote the cone of $n \times n$ symmetric matrices by S^n.

Symmetric Matrices

A matrix $A \in \mathbb{R}^{n \times n}$ is symmetric if and only if it is square and $A_{ij} = A_{ji}$ for all i, j.

- We denote the cone of $n \times n$ symmetric matrices by S^n.

Fact

A matrix $\mathcal{A} \in \mathbb{R}^{n \times n}$ is symmetric if and only if it is orthogonally diagonalizable.

Symmetric Matrices

A matrix $A \in \mathbb{R}^{n \times n}$ is symmetric if and only if it is square and $A_{ij} = A_{ji}$ for all i, j.

- We denote the cone of $n \times n$ symmetric matrices by S^n.

Fact

A matrix $A \in \mathbb{R}^{n \times n}$ is symmetric if and only if it is orthogonally diagonalizable.

- i.e. $A = QDQ^\intercal$ where Q is an orthogonal matrix and $D = \mathbf{diag}(\lambda_1, \ldots, \lambda_n)$.
- The columns of Q are the (normalized) eigenvectors of A, with corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$
- Equivalently: As a linear operator, A scales the space along an orthonormal basis Q
- The scaling factor λ_i along direction q_i may be negative, positive, or 0.

Positive Semi-Definite Matrices

A matrix $A \in \mathbb{R}^{n \times n}$ is positive semi-definite if it is symmetric and moreover all its eigenvalues are nonnegative.

- We denote the cone of $n \times n$ positive semi-definite matrices by S_+^n
- We use $A \succeq 0$ as shorthand for $A \in S_+^n$

Positive Semi-Definite Matrices

A matrix $A \in \mathbb{R}^{n \times n}$ is positive semi-definite if it is symmetric and moreover all its eigenvalues are nonnegative.

- We denote the cone of $n \times n$ positive semi-definite matrices by S_+^n
- We use $A \succeq 0$ as shorthand for $A \in S_+^n$

- $A = QDQ^\intercal$ where Q is an orthogonal matrix and $D = \mathbf{diag}(\lambda_1, \ldots, \lambda_n)$, where $\lambda_i \geq 0$.
- As a linear operator, A performs nonnegative scaling along an orthonormal basis Q

Positive Semi-Definite Matrices

A matrix $A \in \mathbb{R}^{n \times n}$ is positive semi-definite if it is symmetric and moreover all its eigenvalues are nonnegative.

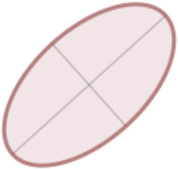
- We denote the cone of $n \times n$ positive semi-definite matrices by S_+^n
- We use $A \succeq 0$ as shorthand for $A \in S_+^n$

- $A = QDQ^\intercal$ where Q is an orthogonal matrix and $D = \mathbf{diag}(\lambda_1, \ldots, \lambda_n)$, where $\lambda_i \geq 0$.
- As a linear operator, A performs nonnegative scaling along an orthonormal basis Q

Note

Positive definite, negative semi-definite, and negative definite defined similarly.

Geometric Intuition for PSD Matrices



- For $A \succeq 0$, let $q_1, \ldots, q_n$ be the orthonormal eigenbasis for A, and let $\lambda_1, \ldots, \lambda_n \geq 0$ be the corresponding eigenvalues.
- The linear operator $x \rightarrow Ax$ scales the q_i component of x by λ_i
- When applied to every x in the unit ball, the image of A is an ellipsoid centered at the origin with principal directions $q_1, \ldots, q_n$ and corresponding diameters $2\lambda_1, \ldots, 2\lambda_n$
  - When A is positive definite ($i.e. \lambda_i > 0$), and therefore invertible, the ellipsoid is the set $\{y : y^T (AA^T)^{-1} y \leq 1\}$

Useful Properties of PSD Matrices

If $A \succeq 0$, then

- $x^T A x \geq 0$ for all x
- A has a positive semi-definite square root $A^{\frac{1}{2}}$
  - $A^{\frac{1}{2}} = Q\,\mathbf{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_n})Q^\intercal$
- $A = B^T B$ for some matrix B.
  - Interpretation: PSD matrices encode the "pairwise similarity" relationships of a family of vectors. A_{ij} is dot product of the ith and jth columns of B.
  - Interpretation: The quadratic form $x^T A x$ is the length of a linear transformation of x, namely $||Bx||_2^2$
- The quadratic function $x^T A x$ is convex
- A can be expressed as a sum of vector outer-products
  - e.g., $A = \sum_{i=1}^n v_i v_i^T$ for $\vec{v_i} = \sqrt{\lambda_i}\vec{q_i}$

Useful Properties of PSD Matrices

If $A \succeq 0$, then

- $x^T A x \geq 0$ for all x
- A has a positive semi-definite square root $A^{\frac{1}{2}}$
  - $A^{\frac{1}{2}} = Q\,\mathbf{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_n})Q^\intercal$
- $A = B^T B$ for some matrix B.
  - Interpretation: PSD matrices encode the "pairwise similarity" relationships of a family of vectors. A_{ij} is dot product of the ith and jth columns of B.
  - Interpretation: The quadratic form $x^T A x$ is the length of a linear transformation of x, namely $||Bx||_2^2$
- The quadratic function $x^T A x$ is convex
- A can be expressed as a sum of vector outer-products
  - e.g., $A = \sum_{i=1}^n v_i v_i^T$ for $\vec{v_i} = \sqrt{\lambda_i}\vec{q_i}$

As it turns out, each of the above is also sufficient for $A \succeq 0$ (assuming A is symmetric).

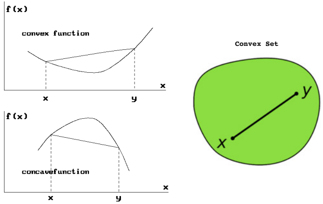
Properties of PSD Matrices Relevant for Computation

- The set of PSD matrices is convex
  - Follows from the characterization: $x^T Ax \geq 0$ for all x
- The set of PSD matrices admits an efficient separation oracle
  - Given A , find eigenvector v with negative eigenvalue: $v^T Av < 0$.
- A PSD matrix $A \in \mathcal{R}^{n \times n}$ implicitly encodes the "pairwise similarities" of a family of vectors $b_1, \ldots, b_n \in R^n$.
  - Follows from the characterization $A = B^T B$ for some B
  - $A_{ij} = \langle b_i, b_j \rangle$
- Can convert between A and B efficiently.
  - B to A: Matrix multiplication
  - A to B: B can be expressed in terms of eigenvectors/eigenvalues of A, which can be easily computed to arbitrary precision via powering methods. Alternatively: Cholesky decomposition, SVD, ...

Outline

Convex Optimization

min (or max)	$f(x)$
subject to	$x \in \mathcal{X}$



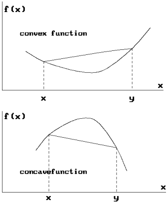
Convex Optimization Problem

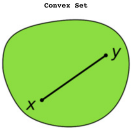
Generalization of LP where

- Feasible set $\mathcal{X}$ convex: $\alpha x + (1-\alpha)y \in \mathcal{X}$, for all $x, y \in \mathcal{X}$ and $\alpha \in [0,1]$
- Objective function f is convex in case of minimization
  - $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y)$ for all $x, y \in \mathcal{X}$ and $\alpha \in [0,1]$
- Objective function f is concave in case of maximization

Convex Optimization

min (or max) $f(x)$
subject to $x \in \mathcal{X}$





Convex Optimization Problems Solvable efficiently (i.e. in polynomial time) to arbitrary precision under mild conditions

- Separation oracle for $\mathcal{X}$
- First-order oracle for evaluating $f(x)$ and $\bigtriangledown f(x)$.

For more detail

Take CSCI 675!

Semidefinite Programs

These are Optimization problems where the feasible set is the cone of PSD cone, possibly intersected with linear constraints.

- Generalization of LP.
- Special case of Convex Optimization.

$$
\begin{array}{ll}
\text{maximize} & c^\mathsf{T} x \\
\text{subject to} & Ax \preceq b \\
 & x_1 F_1 + x_2 F_2 \dots x_n F_n + G \text{ is PSD}
\end{array}
$$

- $F_1, \dots, F_n$, G, and A are given matrices, and c, b are given vectors.

Semidefinite Programs

These are Optimization problems where the feasible set is the cone of PSD cone, possibly intersected with linear constraints.

- Generalization of LP.
- Special case of Convex Optimization.

$$
\begin{array}{ll}
\text{maximize} & c^{\mathsf{T}} x \\
\text{subject to} & Ax \preceq b \\
& x_1 F_1 + x_2 F_2 \ldots x_n F_n + G \text{ is PSD}
\end{array}
$$

- $F_1, \ldots, F_n$, G, and A are given matrices, and c, b are given vectors.

Examples

- Fitting a distribution, say a Gaussian, to observed data. Variable is a positive semi-definite covariance matrix.
- As a relaxation to combinatorial problems that encode pairwise relationships: e.g. finding the maximum cut of a graph.

Semidefinite Programs

These are Optimization problems where the feasible set is the cone of PSD cone, possibly intersected with linear constraints.

- Generalization of LP.
- Special case of Convex Optimization.

$$\begin{array}{ll} \text{maximize} & c^\intercal x \\ \text{subject to} & Ax \preceq b \\ & x_1F_1 + x_2F_2 \ldots x_nF_n + G \text{ is PSD} \end{array}$$

- $F_1, \ldots, F_n$, G, and A are given matrices, and c, b are given vectors.

Fact

SDP can be solved in polytime to arbitrary precision, since PSD constraints admit a polytime separation oracle.

Outline

The Max Cut Problem

Given an undirected graph $G = (V, E)$, find a partition of V into $(S, V \setminus S)$ maximizing number of edges with exactly one end in S.

$$\begin{array}{ll} \text{maximize} & \sum_{(i,j)\in E} \frac{1 - x_i x_j}{2} \\ \text{subject to} & x_i \in \{-1, 1\}, \qquad \text{for } i \in V. \end{array}$$

The Max Cut Problem

Given an undirected graph $G = (V, E)$, find a partition of V into $(S, V \setminus S)$ maximizing number of edges with exactly one end in S.

$$\begin{array}{lll} \text{maximize} & \sum_{(i,j)\in E} \frac{1-x_i x_j}{2} & \\ \text{subject to} & x_i \in \{-1, 1\}\,, & \text{for } i \in V. \end{array}$$

Instead of requiring x_i to be on the 1 dimensional sphere, we relax and permit it to be in the n-dimensional sphere, where $n = |V|$.

Vector Program relaxation

$$\begin{array}{lll} \text{maximize} & \sum_{(i,j)\in E} \frac{1-\vec{v}_i\cdot\vec{v}_j}{2} & \\ \text{subject to} & ||\vec{v}_i||_2 = 1, & \text{for } i \in V. \\ & \vec{v}_i \in \mathbb{R}^n, & \text{for } i \in V. \end{array}$$

SDP Relaxation

- Recall: A symmetric $n \times n$ matrix Y is PSD iff $Y = V^T V$ for $n \times n$ matrix V
- Equivalently: PSD matrices encode pairwise dot products of columns of V
- When diagonal entries of Y are 1, V has unit length columns
- Recall: Y and V can be recovered from each other efficiently

SDP Relaxation

- Recall: A symmetric $n \times n$ matrix Y is PSD iff $Y = V^T V$ for $n \times n$ matrix V
- Equivalently: PSD matrices encode pairwise dot products of columns of V
- When diagonal entries of Y are 1, V has unit length columns
- Recall: Y and V can be recovered from each other efficiently

Vector Program relaxation

$$\begin{array}{lll} \text{maximize} & \sum_{(i,j)\in E} \frac{1-\vec{v}_i\cdot\vec{v}_j}{2} & \\ \text{subject to} & ||\vec{v}_i||_2 = 1, & \text{for } i \in V. \\ & \vec{v}_i \in \mathbb{R}^n, & \text{for } i \in V. \end{array}$$

SDP Relaxation

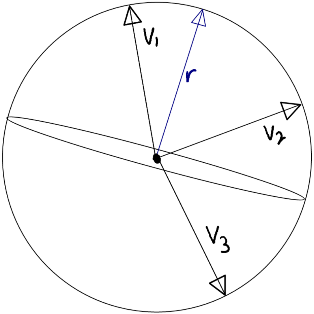
$$\begin{array}{lll} \text{maximize} & \sum_{(i,j)\in E} \frac{1-Y_{ij}}{2} & \\ \text{subject to} & Y_{ii} = 1, & \text{for } i \in V. \\ & Y \in S_+^n & \end{array}$$

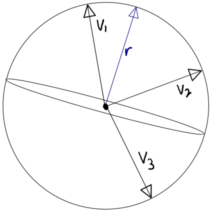
Goemans Williamson Algorithm for Max Cut

1. Solve the SDP to get $Y \succeq 0$
2. Decompose Y to VV^T
3. Draw random vector r on unit sphere
4. Place nodes i with $v_i \cdot r \geq 0$ on one side of cut, the rest on the other side

SDP Relaxation

$$\begin{array}{ll} \text{maximize} & \sum_{(i,j)\in E} \frac{1-Y_{ij}}{2} \\ \text{subject to} & Y_{ii} = 1 \ \forall i \\ & Y \in S_+^n \end{array}$$

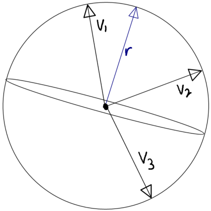




We will prove the following Lemma

Lemma

The random hyperplane cuts each edge (i, j) with probability at least $0.878\frac{1-Y_{ij}}{2}$



We will prove the following Lemma

Lemma

The random hyperplane cuts each edge (i, j) with probability at least $0.878\frac{1-Y_{ij}}{2}$

Therefore, by linearity of expectations, and the fact that $OPT_{SDP} \geq OPT$ (i.e. relaxation).

Theorem

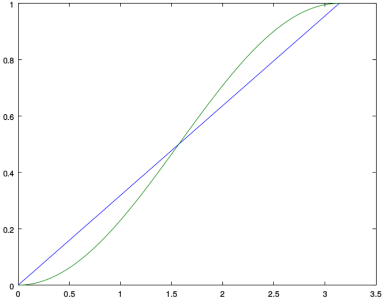
The Goemans Williamson algorithm outputs a random cut of expected size at least $0.878\,OPT$.

We use the following fact

Fact

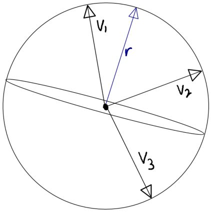
For all angles $\theta \in [0, \pi]$,

$$\frac{\theta}{\pi} \geq 0.878 \cdot \frac{1 - cos(\theta)}{2}$$



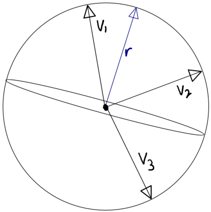
Lemma

The random hyperplane cuts each edge (i, j) with probability at least $0.878\frac{1-Y_{ij}}{2}$



Lemma

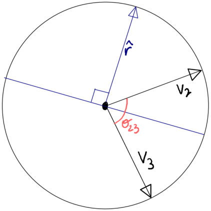
The random hyperplane cuts each edge (i, j) with probability at least $0.878\frac{1-Y_{ij}}{2}$



- (i, j) is cut iff $sign(r \cdot v_i) \neq sign(r \cdot v_j)$

Lemma

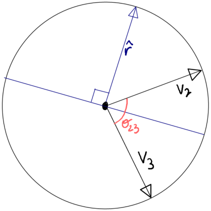
The random hyperplane cuts each edge (i, j) with probability at least $0.878\frac{1-Y_{ij}}{2}$



- (i, j) is cut iff $sign(r \cdot v_i) \neq sign(r \cdot v_j)$
- Can zoom in on the 2-d plane which includes v_i and v_j
  - Discard component r perpendicular to that plane, leaving $\widehat{r}$
  - Direction of $\widehat{r}$ is uniform in the plane

Lemma

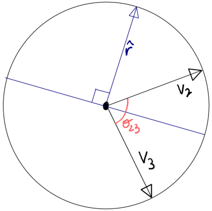
The random hyperplane cuts each edge (i, j) with probability at least $0.878\frac{1-Y_{ij}}{2}$



- (i, j) is cut iff $sign(r \cdot v_i) \neq sign(r \cdot v_j)$
- Can zoom in on the 2-d plane which includes v_i and v_j
  - Discard component r perpendicular to that plane, leaving $\widehat{r}$
  - Direction of $\widehat{r}$ is uniform in the plane
- Let θ_{ij} be angle between v_i and v_j. Note $Y_{ij} = v_i \cdot v_j = \cos(\theta_{ij})$

Lemma

The random hyperplane cuts each edge (i, j) with probability at least $0.878\frac{1-Y_{ij}}{2}$



- (i, j) is cut iff $sign(r \cdot v_i) \neq sign(r \cdot v_j)$
- Can zoom in on the 2-d plane which includes v_i and v_j
  - Discard component r perpendicular to that plane, leaving $\widehat{r}$
  - Direction of $\widehat{r}$ is uniform in the plane
- Let θ_{ij} be angle between v_i and v_j. Note $Y_{ij} = v_i \cdot v_j = \cos(\theta_{ij})$
- $\widehat{r}$ cuts (i, j) w.p.

$$\frac{2\theta_{ij}}{2\pi} = \frac{\theta_{ij}}{\pi} \geq 0.878\frac{1 - \cos\theta_{ij}}{2} = 0.878\frac{1 - Y_{ij}}{2}$$