Optimisation Combinatoire et Convexe.

Semidefinite programming
A **linear program** (LP) is written

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]

where \( x \geq 0 \) means that the coefficients of the vector \( x \) are nonnegative.

- Starts with Dantzig’s simplex algorithm in the late 40s.
- First efficient algorithm with polynomial complexity derived by Karmarkar [1984], using interior point methods.
A semidefinite program (SDP) is written

\[
\begin{align*}
\text{minimize} & \quad \text{Tr}(CX) \\
\text{subject to} & \quad \text{Tr}(A_iX) = b_i, \quad i = 1, \ldots, m \\
& \quad X \succeq 0
\end{align*}
\]

where \( X \succeq 0 \) means that the matrix variable \( X \in S_n \) is positive semidefinite.

- Nesterov and Nemirovskii [1994] showed that the interior point algorithms used for linear programs could be extended to semidefinite programs.
- Key result: \textbf{self-concordance} analysis of Newton’s method (affine invariant smoothness bounds on the Hessian).
Introduction

- Modeling
  - Linear programming started as a toy problem in the 40s, many applications followed.
  - Semidefinite programming has much stronger expressive power, many new applications being investigated today (cf. this talk).
  - Similar conic duality theory.

- Algorithms
  - Robust solvers for solving large-scale linear programs are available today (e.g. MOSEK, CPLEX, GLPK).
  - Not (yet) true for semidefinite programs. Very active work now on first-order methods, motivated by applications in statistical learning (matrix completion, NETFLIX, structured MLE, . . . ).
Outline

- Introduction
- **Semidefinite programming**
  - Conic duality
  - A few words on algorithms
- Recent applications
  - Eigenvalue problems
  - Combinatorial relaxations
  - Ellipsoidal approximations
  - Distortion, embedding
  - Mixing rates for Markov chains & maximum variance unfolding
  - Moment problems & positive polynomials
Semidefinite Programming
Semidefinite programming: conic duality

Direct extension of LP duality results. Start from a semidefinite program

\[
\begin{align*}
\text{minimize} & \quad \text{Tr}(CX) \\
\text{subject to} & \quad \text{Tr}(A_iX) = b_i, \quad i = 1, \ldots, m \\
& \quad X \succeq 0
\end{align*}
\]

which is a convex minimization problem in \( X \in \mathbb{S}_n \). The cone of positive semidefinite matrices is \textbf{self-dual}, i.e.

\[
Z \succeq 0 \iff \text{Tr}(ZX) \geq 0, \quad \text{for all } X \succeq 0,
\]

so we can form the \textbf{Lagrangian}

\[
L(X, y, Z) = \text{Tr}(CX) + \sum_{i=1}^{m} y_i (b_i - \text{Tr}(A_iX)) - \text{Tr}(ZX)
\]

with \textbf{Lagrange multipliers} \( y \in \mathbb{R}^m \) and \( Z \in \mathbb{S}_n \) with \( Z \succeq 0 \).
Semidefinite programming: conic duality

Rearranging terms, we get

\[ L(X, y, Z) = \text{Tr} \left( X (C - \sum_{i=1}^{m} y_i A_i - Z) \right) + b^T y \]

hence, after minimizing this affine function in \( X \in S_n \), the dual can be written

\[
\begin{align*}
\text{maximize} & \quad b^T y \\
\text{subject to} & \quad Z = C - \sum_{i=1}^{m} y_i A_i \\
& \quad Z \succeq 0,
\end{align*}
\]

which is another semidefinite program in the variables \( y, Z \). Of course, the last two constraints can be simplified to

\[ C - \sum_{i=1}^{m} y_i A_i \succeq 0. \]
Semidefinite programming: conic duality

- Primal dual pair

\[
\begin{align*}
\text{minimize} & \quad \text{Tr}(CX) \\
\text{subject to} & \quad \text{Tr}(A_iX) = b_i \\
& \quad X \succeq 0,
\end{align*}
\]

\[
\begin{align*}
\text{maximize} & \quad b^T y \\
\text{subject to} & \quad C - \sum_{i=1}^m y_i A_i \succeq 0.
\end{align*}
\]

- Simple constraint qualification conditions guarantee strong duality.

- We can write a conic version of the KKT optimality conditions

\[
\begin{align*}
C - \sum_{i=1}^m y_i A_i &= Z, \\
\text{Tr}(A_iX) &= b_i, \quad i = 1, \ldots, m, \\
\text{Tr}(XZ) &= 0, \\
X, Z &\succeq 0.
\end{align*}
\]
Semidefinite programming: conic duality

So what?

- Weak duality produces simple bounds on e.g. combinatorial problems.
- Consider the MAXCUT relaxation

\[
\begin{align*}
\text{max.} & \quad x^T C x \\
\text{s.t.} & \quad x_i^2 = 1
\end{align*}
\]

is bounded by

\[
\begin{align*}
\text{max.} & \quad \text{Tr}(XC') \\
\text{s.t.} & \quad \text{diag}(X) = 1 \\
& \quad X \succeq 0,
\end{align*}
\]

in the variables \( x \in \mathbb{R}^n \) and \( X \in \mathbb{S}_n \) (more later on these relaxations).

- The dual of the SDP on the right is written

\[
\min_y n\lambda_{\text{max}}(C - \text{diag}(y)) + 1^T y
\]

in the variable \( y \in \mathbb{R}^n \).

- By weak duality, plugging any value \( y \) in this problem will produce an upper bound on the optimal value of the combinatorial problem above.
**Semidefinite programming: algorithms**

**Algorithms** for semidefinite programming

- Following [Nesterov and Nemirovskii, 1994], most of the attention was focused on interior point methods.
- Newton’s method, with efficient linear algebra solving for the search direction.
- Fast, and robust on small problems ($n \sim 500$).
- Computing the Hessian is too hard on larger problems.

**Solvers**

- Open source solvers: SDPT3, SEDUMI, SDPA, CSDP, . . .
- Very powerful modeling systems: CVX
Solving the maxcut relaxation

\[
\begin{align*}
\text{max.} & \quad \text{Tr}(XC) \\
\text{s.t.} & \quad \text{diag}(X) = 1 \\
& \quad X \succeq 0,
\end{align*}
\]

is written as follows in CVX/MATLAB

```matlab
cvx_begin
\hspace{1cm}. \quad \text{variable } X(n,n) \text{ symmetric}
\hspace{1cm}. \quad \text{maximize } \text{trace}(C\ast X)
\hspace{1cm}. \quad \text{subject to}
\hspace{1cm}. \hspace{1cm}. \quad \text{diag}(X)==1
\hspace{1cm}. \hspace{1cm}. \quad X==\text{semidefinite}(n)
\hspace{1cm}cvx_end
```
Solving large-scale problems is a bit more problematic.

- No universal algorithm known yet. No CVX like modeling system.
- Performance and algorithmic choices heavily depends on problem structure.
- Very basic codes only require computing one leading eigenvalue per iteration, with complexity $O(n^2 \log n)$ using e.g. Lanczos.
- Each iteration requires about 300 matrix vector products, but making progress may require many iterations. Typically $O(1/\epsilon^2)$ or $O(1/\epsilon)$ in some cases.
- In general, most optimization algorithms are purely sequential, so only the linear algebra subproblems benefit from the multiplication of CPU cores.
Outline

- Introduction
- Semidefinite programming
  - Conic duality
  - A few words on algorithms
- Recent applications
  - Eigenvalue problems, Combinatorial relaxations
  - Ellipsoidal approximations
  - Distortion, embedding
  - Mixing rates for Markov chains & maximum variance unfolding
  - Moment problems & positive polynomials
  - Gordon-Slepian and the maximum of Gaussian processes
  - Collaborative prediction
Many classical problems can be cast as or approximated by semidefinite programs.

Recognizing this is not always obvious.

At reasonable scales, numerical solutions often significantly improve on classical closed-form bounds.

A few examples follow...
Eigenvalue problems
Eigenvalue problems

Start from a semidefinite program with constant trace

\[
\begin{align*}
\text{minimize} & \quad \text{Tr}(CX) \\
\text{subject to} & \quad \text{Tr}(A_iX) = b_i, \quad i = 1, \ldots, m \\
& \quad \text{Tr}(X) = 1 \\
& \quad X \succeq 0
\end{align*}
\]

in the variable \( X \in \mathbf{S}_n \). Because

\[
\max_{\text{Tr}(X)=1, \quad X \succeq 0} \quad \text{Tr}(CX) = \lambda_{\text{max}}(C),
\]

the dual semidefinite program is written

\[
\min_y \quad \lambda_{\text{max}} \left( C - \sum_{i=1}^m y_i \right) + b^T y
\]

in the variable \( y \in \mathbb{R}^m \).

Maximum eigenvalue minimization problems are usually easier to solve using first-order methods.
Combinatorial relaxations
Semidefinite programs with constant trace often arise in **convex relaxations** of combinatorial problems. Use MAXCUT as an example here.

The problem is written

$$\max \quad x^T C x$$

$$\text{s.t.} \quad x \in \{-1, 1\}^n$$

in the binary variables $x \in \{-1, 1\}^n$, with parameter $C \in S_n$ (usually $C \succeq 0$). This problem is known to be **NP-Hard**. Using

$$x \in \{-1, 1\}^n \iff x_i^2 = 1, \quad i = 1, \ldots, n$$

we get

$$\max \quad x^T C x$$

$$\text{s.t.} \quad x_i^2 = 1, \quad i = 1, \ldots, n$$

which is a nonconvex quadratic program in the variable $x \in \mathbb{R}^n$. 
Combinatorial relaxations

We now do a simple change of variables, setting $X = xx^T$, with

$$X = xx^T \iff X \in \mathbf{S}_n, \ X \succeq 0, \ \text{Rank}(X) = 1$$

and we also get

$$\text{Tr}(CX) = x^T C x$$
$$\text{diag}(X) = 1 \iff x_i^2 = 1, \ i = 1, \ldots, n$$

so the original combinatorial problem is equivalent to

$$\max. \ \text{Tr}(CX)$$
$$\text{s.t.} \ \text{diag}(X) = 1$$
$$X \succeq 0, \ \text{Rank}(X) = 1$$

which is now a nonconvex problem in $X \in \mathbf{S}_n$. 
Combinatorial relaxations

- If we simply drop the rank constraint, we get the following relaxation:

\[
\begin{align*}
\text{max.} \quad & x^T C x \\
\text{s.t.} \quad & x \in \{-1, 1\}^n \\
\end{align*}
\]

is bounded by

\[
\begin{align*}
\text{max.} \quad & \text{Tr}(CX) \\
\text{s.t.} \quad & \text{diag}(X) = 1 \\
& X \succeq 0,
\end{align*}
\]

which is a semidefinite program in \( X \in S_n \).

- **Rank constraints** in semidefinite programs are usually hard. All semi-algebraic optimization problems can be formulated as rank constrained SDPs.

- Randomization techniques produce bounds on the approximation ratio. When \( C \succeq 0 \) for example, we have

\[
\frac{2}{\pi} \text{SDP} \leq \text{OPT} \leq \text{SDP}
\]

for the MAXCUT relaxation (more details in [Ben-Tal and Nemirovski, 2001]).

- Applications in graph, matrix approximations (CUT-Norm, \( \| \cdot \|_{1 \rightarrow 2} \)) [Frieze and Kannan, 1999, Alon and Naor, 2004, Nemirovski, 2005]
Ellipsoidal approximations
Ellipsoidal approximations

**Minimum volume ellipsoid** $\mathcal{E}$ s.t. $C \subseteq \mathcal{E}$ (Löwner-John ellipsoid).

- Parametrize $\mathcal{E}$ as $\mathcal{E} = \{ v \mid \|Av + b\|_2 \leq 1 \}$ with $A \succ 0$.
- Vol $\mathcal{E}$ is proportional to $\det A^{-1}$; to compute minimum volume ellipsoid,

$$\begin{align*}
\text{minimize (over } A, b) & \log \det A^{-1} \\
\text{subject to } & \sup_{v \in C} \|Av + b\|_2 \leq 1
\end{align*}$$

convex, but the constraint can be hard (for general sets $C$).

**Finite set** $C = \{x_1, \ldots, x_m\}$, or polytope with polynomial number of vertices:

$$\begin{align*}
\text{minimize (over } A, b) & \log \det A^{-1} \\
\text{subject to } & \|Ax_i + b\|_2 \leq 1, \quad i = 1, \ldots, m
\end{align*}$$

also gives Löwner-John ellipsoid for polyhedron $\text{Co}\{x_1, \ldots, x_m\}$
Ellipsoidal approximations

**Maximum volume ellipsoid** $\mathcal{E}$ inside a convex set $C \subseteq \mathbb{R}^n$

- parametrize $\mathcal{E}$ as $\mathcal{E} = \{Bu + d \mid \|u\|_2 \leq 1\}$ with $B > 0$.
- vol $\mathcal{E}$ is proportional to $\det B$, we can compute $\mathcal{E}$ by solving

$$
\begin{align*}
\text{maximize} & \quad \log \det B \\
\text{subject to} & \quad \sup_{\|u\|_2 \leq 1} I_C(Bu + d) \leq 0
\end{align*}
$$

(where $I_C(x) = 0$ for $x \in C$ and $I_C(x) = \infty$ for $x \not\in C$) again, this is a convex problem, but evaluating the constraint can be hard (for general $C$)

**Polyhedron** given by its facets $\{x \mid a_i^T x \leq b_i, \ i = 1, \ldots, m\}$:

$$
\begin{align*}
\text{maximize} & \quad \log \det B \\
\text{subject to} & \quad \|Ba_i\|_2 + a_i^T d \leq b_i, \quad i = 1, \ldots, m
\end{align*}
$$

(constRAINT follows from $\sup_{\|u\|_2 \leq 1} a_i^T (Bu + d) = \|Ba_i\|_2 + a_i^T d$)
Ellipsoidal approximations

$C \subseteq \mathbb{R}^n$ convex, bounded, with nonempty interior

- Löwner-John ellipsoid, shrunk by a factor $n$, lies inside $C$
- maximum volume inscribed ellipsoid, expanded by a factor $n$, covers $C$

**example** (for two polyhedra in $\mathbb{R}^2$)

factor $n$ can be improved to $\sqrt{n}$ if $C$ is symmetric. See [Boyd and Vandenberghe, 2004] for further examples.
Distortion, embedding problems, . . .
Distortion, embedding problems, . . .

We cannot hope to always get low rank solutions, unless we are willing to admit some distortion. . . The following result from [Ben-Tal, Nemirovski, and Roos, 2003] gives some guarantees.

**Theorem**

**Approximate $S$-lemma.** Let $A_1, \ldots, A_N \in \mathbb{S}_n$, $\alpha_1, \ldots, \alpha_N \in \mathbb{R}$ and a matrix $X \in \mathbb{S}_n$ such that

\[ A_i, X \succeq 0, \quad \text{Tr}(A_iX) = \alpha_i, \quad i = 1, \ldots, N \]

Let $\epsilon > 0$, there exists a matrix $X_0$ such that

\[ \alpha_i(1 - \epsilon) \leq \text{Tr}(A_iX_0) \leq \alpha_i(1 + \epsilon) \quad \text{and} \quad \text{Rank}(X_0) \leq 8 \log 4N/\epsilon^2 \]

**Proof.** Randomization, concentration results on Gaussian quadratic forms.

A particular case: Given $N$ vectors $v_i \in \mathbb{R}^d$, construct their Gram matrix $X \in S_N$, with

$$X \succeq 0, \quad X_{ii} - 2X_{ij} + X_{jj} = \|v_i - v_j\|_2^2, \quad i, j = 1, \ldots, N.$$ 

The matrices $D_{ij} \in S_n$ such that

$$\text{Tr}(D_{ij}X) = X_{ii} - 2X_{ij} + X_{jj}, \quad i, j = 1, \ldots, N$$

satisfy $D_{ij} \succeq 0$. Let $\epsilon > 0$, there exists a matrix $X_0$ with

$$m = \text{Rank}(X_0) \leq 16 \frac{\log 2N}{\epsilon^2},$$

from which we can extract vectors $u_i \in \mathbb{R}^m$ such that

$$\|v_i - v_j\|_2^2 (1 - \epsilon) \leq \|u_i - u_j\|_2^2 \leq \|v_i - v_j\|_2^2 (1 + \epsilon).$$

In this setting, the Johnson-Lindenstrauss lemma is a particular case of the approximate $S$ lemma. . .
The problem of reconstructing an $N$-point Euclidean metric, given partial information on pairwise distances between points $v_i$, $i = 1, \ldots, N$ can also be cast as an SDP, known as and **Euclidean Distance Matrix Completion** problem.

\[
\begin{align*}
\text{find} & \quad D \\
\text{subject to} & \quad 1v^T + v1^T - D \succeq 0 \\
& \quad D_{ij} = \|v_i - v_j\|_2^2, \quad (i, j) \in S \\
& \quad v \geq 0
\end{align*}
\]

in the variables $D \in \mathbf{S}_n$ and $v \in \mathbb{R}^n$, on a subset $S \subset [1, N]^2$.

- We can add further constraints to this problem given additional structural info on the configuration.

- Applications in sensor networks, molecular conformation reconstruction etc. . .
[Dattorro, 2005] 3D map of the USA reconstructed from pairwise distances on 5000 points. Distances reconstructed from Latitude/Longitude data.
Mixing rates for Markov chains & maximum variance unfolding
Let $G = (V, E)$ be an undirected graph with $n$ vertices and $m$ edges.

We define a Markov chain on this graph, and let $w_{ij} \geq 0$ be the transition rate for edge $(i, j) \in V$.

Let $\pi(t)$ be the state distribution at time $t$, its evolution is governed by the heat equation

$$d\pi(t) = -L\pi(t) dt$$

with

$$L_{ij} = \begin{cases} 
-w_{ij} & \text{if } i \neq j, (i, j) \in V \\
0 & \text{if } (i, j) \notin V \\
\sum_{(i, k) \in V} w_{ik} & \text{if } i = j 
\end{cases}$$

the graph Laplacian matrix, which means

$$\pi(t) = e^{-Lt}\pi(0).$$

The matrix $L \in \mathbb{S}_n$ satisfies $L \succeq 0$ and its smallest eigenvalue is zero.
Mixing rates for Markov chains & unfolding

- With
  \[ \pi(t) = e^{-Lt}\pi(0) \]
  the **mixing rate** is controlled by the second smallest eigenvalue \( \lambda_2(L) \).

- Since the smallest eigenvalue of \( L \) is zero, with eigenvector \( \mathbf{1} \), we have
  \[ \lambda_2(L) \geq t \iff L(w) \succeq t(\mathbf{I} - (1/n)\mathbf{1}\mathbf{1}^T), \]

- Maximizing the mixing rate of the Markov chain means solving
  \[
  \begin{align*}
  \text{maximize} & \quad t \\
  \text{subject to} & \quad L(w) \succeq t(\mathbf{I} - (1/n)\mathbf{1}\mathbf{1}^T) \\
  & \quad \sum_{(i,j) \in V} d_{ij}^2 w_{ij} \leq 1 \\
  & \quad w \geq 0
  \end{align*}
  \]
  in the variable \( w \in \mathbb{R}^m \), with (normalization) parameters \( d_{ij}^2 \geq 0 \).

- Since \( L(w) \) is an affine function of the variable \( w \in \mathbb{R}^m \), this is a semidefinite program in \( w \in \mathbb{R}^m \).

- Numerical solution usually performs better than **Metropolis-Hastings**.
We can also form the dual of the maximum MC mixing rate problem.

The dual means solving

\[
\begin{align*}
\text{maximize} & \quad \text{Tr}(X(I - (1/n)11^T)) \\
\text{subject to} & \quad X_{ii} - 2X_{ij} + X_{jj} \leq d_{ij}^2, \quad (i, j) \in V \\
& \quad X \succeq 0,
\end{align*}
\]

in the variable \( X \in \mathbb{S}_n \).

Here too, we can interpret \( X \) as the gram matrix of a set of \( n \) vectors \( v_i \in \mathbb{R}^d \).

The program above maximizes the variance of the vectors \( v_i \)

\[
\text{Tr}(X(I - (1/n)11^T)) = \sum_i \|v_i\|_2^2 - \|\sum_i v_i\|_2^2
\]

while the constraints bound pairwise distances

\[
X_{ii} - 2X_{ij} + X_{jj} \leq d_{ij}^2 \iff \|v_i - v_j\|_2^2 \leq d_{ij}^2
\]

This is a maximum variance unfolding problem [Weinberger and Saul, 2006, Sun et al., 2006].
From [Sun et al., 2006]: we are given pairwise 3D distances for $k$-nearest neighbors in the point set on the right. We plot the maximum variance point set satisfying these pairwise distance bounds on the right.
Moment problems & positive polynomials
[Nesterov, 2000]. Hilbert’s 17th problem has a positive answer for univariate polynomials: a polynomial is nonnegative iff it is a sum of squares

\[ p(x) = x^{2d} + \alpha_{2d-1}x^{2d-1} + \ldots + \alpha_0 \geq 0, \text{ for all } x \iff p(x) = \sum_{i=1}^{N} q_i(x)^2 \]

We can formulate this as a linear matrix inequality, let \( v(x) \) be the moment vector

\[ v(x) = (1, x, \ldots, x^d)^T \]

we have

\[ \sum_i \lambda_i u_i u_i^T = M \succeq 0 \iff p(x) = v(x)^T M v(x) = \sum_i \lambda_i (u_i^T v(x))^2 \]

where \((\lambda_i, u_i)\) are the eigenpairs of \( M \).
The dual to the cone of Sum-of-Squares polynomials is the cone of moment matrices

\[ \mathbf{E}_\mu[x^i] = q_i, \ i = 0, \ldots, d \iff \begin{pmatrix} q_0 & q_1 & \cdots & q_d \\ q_1 & q_2 & \cdots & q_{d+1} \\ \vdots & \vdots & \ddots & \vdots \\ q_d & q_{d+1} & \cdots & q_{2d} \end{pmatrix} \succeq 0 \]


This forms exponentially large, ill-conditioned semidefinite programs however.
Collaborative prediction
Users assign **ratings** to a certain number of movies:

Objective: make recommendations for other movies...
Collaborative prediction

- Infer **user preferences** and **movie features** from user ratings.
- We use a linear prediction model:

\[ \text{rating}_{ij} = u_i^T v_j \]

where \( u_i \) represents user characteristics and \( v_j \) movie features.
- This makes collaborative prediction a **matrix factorization** problem.
- Overcomplete representation...
Collaborative prediction

- **Inputs**: a matrix of ratings $M_{ij} = \{-1, +1\}$ for $(i, j) \in S$, where $S$ is a subset of all possible user/movies combinations.

- We look for a linear model by factorizing $M \in \mathbb{R}^{n \times m}$ as:

  $$M = U^T V$$

  where $U \in \mathbb{R}^{n \times k}$ represents user characteristics and $V \in \mathbb{R}^{k \times m}$ movie features.

- **Parsimony**: We want $k$ to be as small as possible.

- **Output**: a matrix $X \in \mathbb{R}^{n \times m}$ which is a low-rank approximation of the ratings matrix $M$. 
Choose Means Squared Error as measure of discrepancy.

Suppose $S$ is the full set, our problem becomes:

$$\min_{\{X: \text{rank}(X)=k\}} \|X - M\|^2$$

This is just a **singular value decomposition** (SVD).

Problem: Not true when $S$ is not the full set (partial observations). Also, MSE not a good measure of prediction performance.
Soft Margin

\[
\text{minimize} \quad \text{Rank}(X) + c \sum_{(i,j) \in S} \max(0, 1 - X_{ij} M_{ij})
\]

non-convex and numerically hard. . .

- Relaxation result in Fazel et al. [2001]: replace \text{Rank}(X) by its convex envelope on the spectahedron to solve:

\[
\text{minimize} \quad \|X\|_* + c \sum_{(i,j) \in S} \max(0, 1 - X_{ij} M_{ij})
\]

where \(\|X\|_*\) is the **nuclear norm**, \(i.e.\) sum of the singular values of \(X\).

- Srebro [2004]: This relaxation also corresponds to multiple large margin SVM classifications.
The dual of this program:

\[
\begin{align*}
\text{maximize} & \quad \sum_{ij} Y_{ij} \\
\text{subject to} & \quad \|Y \odot M\|_2 \leq 1 \\
& \quad 0 \leq Y_{ij} \leq c
\end{align*}
\]

in the variable \( Y \in \mathbb{R}^{n \times m} \), where \( Y \odot M \) is the Schur (componentwise) product of \( Y \) and \( M \) and \( \|Y\|_2 \) the largest singular value of \( Y \).

This problem is \textbf{sparse}: \( Y_{ij}^* = c \) for \((i, j) \in S^c\)
References


