Convex Optimization M2

Semidefinite Programming Applications
Distortion, embedding problems, . . .
We cannot hope to always get low rank solutions to SDPs, 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 $A_1, \ldots, A_N \in S_n$, $\alpha_1, \ldots, \alpha_N \in \mathbb{R}$ and a matrix $X \in S_n$ such that

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

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

$$\alpha_i(1 - \epsilon) \leq \text{Tr}(A_i X_0) \leq \alpha_i(1 + \epsilon) \quad \text{and} \quad \text{Rank}(X_0) \leq 8 \frac{\log 4N}{\epsilon^2}$$

**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.

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

in the variables \( D \in S_n \) 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. . .
Distortion, embedding problems, . . .

Figure 140: Map of United States of America showing some state boundaries and the Great Lakes. All plots made by connecting 5020 points. Any difference in scale in (a) through (d) is artifact of plotting routine.

(a) Shows original map made from decimated (latitude, longitude) data.

(b) Original map data rotated (freehand) to highlight curvature of Earth.

(c) Map isometrically reconstructed from an EDM (from distance only).

(d) Same reconstructed map illustrating curvature.

(e),(f) Two views of one isotonic reconstruction (from comparative distance); problem (1181) with no sort constraint $\Pi_d$ (and no hidden line removal).

[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
Mixing rates for Markov chains & unfolding

[Sun, Boyd, Xiao, and Diaconis, 2006]

- 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 $L \in S_n$ satisfies $L \succeq 0$ and its smallest eigenvalue is zero.
Mixing rates for Markov chains & unfolding

- With
  \[ \pi(t) = e^{-Lt}\pi(0) \]
  the \textbf{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
  \[ \lambda_2(L) \geq t \iff L(w) \succeq t(I - (1/n)\mathbf{1}\mathbf{1}^T), \]
- Maximizing the mixing rate of the Markov chain means solving
  \[
  \begin{align*}
  \text{maximize} & \quad t \\
  \text{subject to} & \quad L(w) \succeq t(I - (1/n)\mathbf{1}\mathbf{1}^T) \\
  & \quad \sum_{(i,j) \in V} d_{ij}^2 w_{ij} \leq 1 \\
  & \quad w \geq 0
  \end{align*}
  \]
  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 \textbf{Metropolis-Hastings}.
Mixing rates for Markov chains & unfolding

- 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 \(X \in S_n\).

- 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

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

- 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

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


This forms exponentially large, ill-conditioned semidefinite programs however.
Collaborative prediction
Collaborative prediction

- Users assign **ratings** to a certain number of movies:

  ![Rating Matrix]

- 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

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

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:

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

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


A. d’Aspremont. Convex Optimization M2, MVA.