Projections onto Spectral Matrix Cones

Daniel Cederberg

Joint work with Stephen Boyd

Talk given at Linköping University
November 30, 2024

Outline

Conic convex optimization

Spectral matrix cones

Projecting onto spectral matrix cones

Numerical experiments

Conic convex optimization

$$
\begin{array}{ll}
\text{minimize} & c^T x \\
\text{subject to} & Ax + s = b \\
& s \in K
\end{array}
$$

- variables $x \in \mathbf{R}^n$ and $s \in \mathbf{R}^m$, and problem data $A,\ b,\ c$ and convex cone K

Conic convex optimization

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax + s = b \\ & s \in K \end{array}$$

- variables $x \in \mathbf{R}^n$ and $s \in \mathbf{R}^m$, and problem data $A, \ b, \ c$ and convex cone K
- any convex problem you encounter in practice can be expressed using *standard cones*:
  - nonnegative orthant, second-order cone:

$$K = \{y \in \mathbf{R}^m \mid y \geq 0\}, \qquad K = \{(t, y) \in \mathbf{R} \times \mathbf{R}^m \mid \|y\|_2 \leq t\}$$

  - positive semidefinite cone:

$$K = \{Y \in \mathbf{S}^m \mid Y \succeq 0\}$$

  - exponential cones and power cones

Conic convex optimization

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax + s = b \\ & s \in K \end{array}$$

- variables $x \in \mathbf{R}^n$ and $s \in \mathbf{R}^m$, and problem data $A,\ b,\ c$ and convex cone K
- any convex problem you encounter in practice can be expressed using *standard cones*:
  - nonnegative orthant, second-order cone:

$$K = \{y \in \mathbf{R}^m \mid y \geq 0\}, \qquad K = \{(t, y) \in \mathbf{R} \times \mathbf{R}^m \mid \|y\|_2 \leq t\}$$

  - positive semidefinite cone:

$$K = \{Y \in \mathbf{S}^m \mid Y \succeq 0\}$$

  - exponential cones and power cones
- frameworks like CVXPY [Diamond & Boyd, 2016] translate convex problems to this form

Conic convex optimization in practice - an example

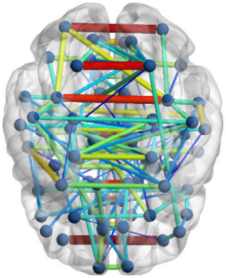
$$\text{minimize} - \log\det X + \mathbf{Tr}(SX) + \lambda\|X\|_1$$

- variable $X \in \mathbf{S}^n$ represents an inverse covariance matrix of a Gaussian vector
- the problem data is the sample covariance matrix $S \in \mathbf{S}^n_+$
- called *sparse inverse covariance estimation* (applications in graphical modeling)

Conic convex optimization in practice - an example

$$\text{minimize } -\log\det X + \mathbf{Tr}(SX) + \lambda\|X\|_1$$

- variable $X \in \mathbf{S}^n$ represents an inverse covariance matrix of a Gaussian vector
- the problem data is the sample covariance matrix $S \in \mathbf{S}^n_+$
- called *sparse inverse covariance estimation* (applications in graphical modeling)



Conic convex optimization in practice - an example

$$\text{minimize} \; -\log\det X + \mathbf{Tr}(SX) + \lambda\|X\|_1$$

What the user sees:

```
import cvxpy as cp
S, lmbda = ...
X = cp.Variable((n, n), symmetric=True)
obj = -cp.logdet(X) + cp.trace(S @ X) +
    lmbda * cp.norm(X, 1)
prob = cp.Problem(cp.Minimize(obj))
prob.solve()
```

What the solver sees:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax + s = b \\ & s \in K \end{array}$$

where A, b, c, K and x are defined on the next slide

Conic convex optimization in practice - an example

$$\text{minimize} \; -\log\det X + \mathbf{Tr}(SX) + \lambda \|X\|_1$$

What the user sees:

```
import cvxpy as cp
S, lmbda = ...
X = cp.Variable((n, n), symmetric=True)
obj = -cp.logdet(X) + cp.trace(S @ X) +
     lmbda * cp.norm(X, 1)
prob = cp.Problem(cp.Minimize(obj))
prob.solve()
```

What the solver sees:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax + s = b \\ & s \in K \end{array}$$

where A, b, c, K and x are defined on the next slide

- transforming a problem to standard form is called *canonicalization*

What the solver sees after canonicalization

- A is extremely sparse; for $n = 50$ the density is less than 0.05%
- error prone to canonicalize by hand (even for professors...)

- $x = (\text{vec}(X), \text{Lower}(Z), t, v) \in \mathbf{R}^{\frac{n(n+1)}{2}} \times \mathbf{R}^{\frac{n(n+1)}{2}} \times \mathbf{R}^{n} \times \mathbf{R}^{\frac{n(n+1)}{2}}$
- $K = \mathbf{R}_+^{2n} \times \mathbf{R}_+^{n(n-1)} \times \text{vec}(\mathbf{S}_+^{2n}) \times \underbrace{K_{\text{exp}} \times \cdots \times K_{\text{exp}}}_{n \text{ exponential cones}}$
- $c = (\text{vec}(S), 0, \mathbf{1}, \mathbf{1})$
- $b[n(n+1)/2 + n(2n+1) + 1 + 3i] = 1$ for $i = 1, \ldots, n$, otherwise 0
- $A \sim (3n^2 + 5n) \times (n(n+2) + n(n+1)/2)$

Which cones are important in practice?

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax + s = b \\ & s \in K \end{array}$$

- most convex problems in practice are LPs or QPs
  - can be canonicalized using nonnegative orthant and second-order cone

Which cones are important in practice?

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax + s = b \\ & s \in K \end{array}$$

- most convex problems in practice are LPs or QPs
  - can be canonicalized using nonnegative orthant and second-order cone
- some problems contain logarithms, exponentials, and powers of scalar variables
  - can be canonicalized using the exponential and power cones

Which cones are important in practice?

$$
\begin{array}{ll}
\text{minimize} & c^T x \\
\text{subject to} & Ax + s = b \\
 & s \in K
\end{array}
$$

- most convex problems in practice are LPs or QPs
  - can be canonicalized using nonnegative orthant and second-order cone
- some problems contain logarithms, exponentials, and powers of scalar variables
  - can be canonicalized using the exponential and power cones
- the above cones cover more than $\approx 97\%$ of applications
  - the remaining problems are canonicalized using the positive semidefinite cone

Which cones are important in practice?

$$
\begin{array}{ll}
\text{minimize} & c^T x \\
\text{subject to} & Ax + s = b \\
& s \in K
\end{array}
$$

- most convex problems in practice are LPs or QPs
  - can be canonicalized using nonnegative orthant and second-order cone
- some problems contain logarithms, exponentials, and powers of scalar variables
  - can be canonicalized using the exponential and power cones
- the above cones cover more than $\approx$ 97% of applications
  - the remaining problems are canonicalized using the positive semidefinite cone
- in this talk we'll focus on solving semidefinite programs (SDPs) faster...

Solving conic convex optimization problems

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax + s = b \\ & s \in K \end{array}$$

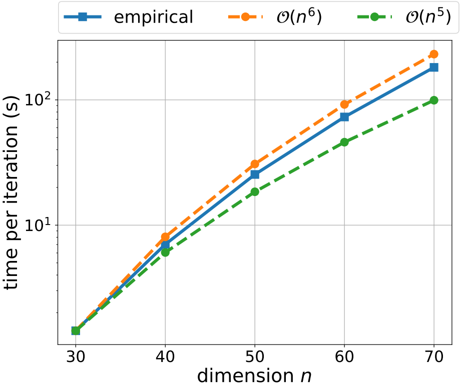
- interior-point methods (Mosek, Clarabel [Goulart 2024]):
  - reliable and accurate solutions in 10–100 iterations
  - the KKT system has a dense $m \times m$ block for SDPs (m is the number of rows of A)
- first-order solvers (SCS [O'Donoghue 2016], COSMO [Garstka 2021]):
  - less reliable and computes solutions with low to modest accuracy
  - must project onto K in each iteration

Solving conic convex optimization problems

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax + s = b \\ & s \in K \end{array}$$

- interior-point methods (Mosek, Clarabel [Goulart 2024]):
  - reliable and accurate solutions in 10–100 iterations
  - the KKT system has a dense $m \times m$ block for SDPs (m is the number of rows of A)
- first-order solvers (SCS [O'Donoghue 2016], COSMO [Garstka 2021]):
  - less reliable and computes solutions with low to modest accuracy
  - must project onto K in each iteration
- IPMs are often the preferred choice but SDPs can be challenging
  - for a matrix variable $X \in \mathbf{S}^p$, the canonicalization may introduce $\mathcal{O}(p^2)$ rows to A
  - results in iteration complexity of order $\mathcal{O}(p^6)$

Empirical scaling interior-point method (Clarabel)



- problem is to minimize $\mathbf{Tr}(SX) - \log\det X$ over $X \in \mathbf{S}^n$
- first-order solvers are very relevant for SDPs!

Outline

Conic convex optimization

Spectral matrix cones

Projecting onto spectral matrix cones

Numerical experiments

A motivating example

$$\text{minimize } \mathbf{Tr}(SX) - \log\det X$$

- variable $X \in \mathbf{S}^n$ and problem data $S \in \mathbf{S}^n_{++}$

A motivating example

$$\text{minimize } \mathbf{Tr}(SX) - \log\det X$$

- variable $X \in \mathbf{S}^n$ and problem data $S \in \mathbf{S}^n_{++}$
- first step of canonicalization is to rewrite it as

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(SX) - \sum_{i=1}^n \log Z_{ii} \\ \text{subject to} & \begin{bmatrix} X & Z \\ Z^T & \mathbf{diag}(Z) \end{bmatrix} \succeq 0, \\ & Z \text{ lower triangular}, \end{array}$$

with variables $X \in \mathbf{S}^n$ and $Z \in \mathbf{R}^{n \times n}$

A motivating example

$$\text{minimize } \mathbf{Tr}(SX) - \log\det X$$

- variable $X \in \mathbf{S}^n$ and problem data $S \in \mathbf{S}^n_{++}$
- first step of canonicalization is to rewrite it as

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(SX) - \sum_{i=1}^n \log Z_{ii} \\ \text{subject to} & \begin{bmatrix} X & Z \\ Z^T & \mathbf{diag}(Z) \end{bmatrix} \succeq 0, \\ & Z \text{ lower triangular,} \end{array}$$

  with variables $X \in \mathbf{S}^n$ and $Z \in \mathbf{R}^{n \times n}$
- canonicalization to PSD cone significantly increases the size of the problem

A motivating example

$$\text{minimize } \mathbf{Tr}(SX) - \log\det X$$

- variable $X \in \mathbf{S}^n$ and problem data $S \in \mathbf{S}^n_{++}$
- first step of canonicalization is to rewrite it as

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(SX) - \sum_{i=1}^n \log Z_{ii} \\ \text{subject to} & \begin{bmatrix} X & Z \\ Z^T & \mathbf{diag}(Z) \end{bmatrix} \succeq 0, \\ & Z \text{ lower triangular,} \end{array}$$

  with variables $X \in \mathbf{S}^n$ and $Z \in \mathbf{R}^{n \times n}$
- canonicalization to PSD cone significantly increases the size of the problem
- **can we define cones that offer more compact canonicalization?**

Spectral functions

- **Definition:** a function $F : \mathbf{S}^n \to \mathbf{R}$ is *spectral* if $F(X) = F(UXU^T)$ for all orthogonal matrices $U \in \mathbf{R}^{n\times n}$ [Chandler 1957, Lewis 1995].
- a function F is spectral if the value $F(X)$ only depends on the spectrum of X
- **Examples**:

$$\log\det X = \sum_{i=1}^n \log \lambda_i(X) \qquad \mathbf{Tr}\, X = \sum_{i=1}^n \lambda_i(X)$$

Spectral matrix cones

- given a spectral function $F : \mathbf{S}^n \to \mathbf{R}$ we define the set $K_F \subset \mathbf{R} \times \mathbf{R}_{++} \times \mathbf{S}^n$ as

$$K_F = \mathbf{cl}\{(t, v, X) \in \mathbf{R} \times \mathbf{R}_{++} \times \mathbf{S}^n \mid vF(X/v) \leq t\}$$

- we call cones constructed this way *spectral matrix cones*
- constructing cones from the epigraph of a perspective function is a classic trick [Rockafellar 1970]

Spectral matrix cones

- given a spectral function $F : \mathbf{S}^n \to \mathbf{R}$ we define the set $K_F \subset \mathbf{R} \times \mathbf{R}_{++} \times \mathbf{S}^n$ as

$$K_F = \mathbf{cl}\{(t, v, X) \in \mathbf{R} \times \mathbf{R}_{++} \times \mathbf{S}^n \mid vF(X/v) \leq t\}$$

- we call cones constructed this way *spectral matrix cones*
- constructing cones from the epigraph of a perspective function is a classic trick [Rockafellar 1970]
- **Example**: $F(X) = -\log\det X$ induces the log-determinant cone

$$K_{\text{logdet}} = \mathbf{cl}\{(t, v, X) \in \mathbf{R} \times \mathbf{R}_{++} \times \mathbf{S}^n \mid -v\log\det(X/v) \leq t\}$$

Canonicalization using log-determinant cone

$$\text{minimize } \mathbf{Tr}(SX) - \log\det X$$

with variable $X \in \mathbf{S}^n$

Canonicalization based on PSD cone:

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(SX) - \sum_{i=1}^n \log Z_{ii} \\ \text{subject to} & \begin{bmatrix} X & Z \\ Z^T & \mathbf{diag}(Z) \end{bmatrix} \succeq 0, \\ & Z \text{ lower triangular,} \end{array}$$

- canonicalization *not* done

Canonicalization using log-determinant cone

$$\text{minimize } \mathbf{Tr}(SX) - \log\det X$$

with variable $X \in \mathbf{S}^n$

Canonicalization based on PSD cone:

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(SX) - \sum_{i=1}^n \log Z_{ii} \\ \text{subject to} & \begin{bmatrix} X & Z \\ Z^T & \mathbf{diag}(Z) \end{bmatrix} \succeq 0, \\ & Z \text{ lower triangular}, \end{array}$$

- canonicalization *not* done

Canonicalization based on logdet-cone:

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(SX) + t \\ \text{subject to} & v = 1 \\ & (t, v, X) \in K_{\text{logdet}} \end{array}$$

- canonicalization done

Canonicalization using log-determinant cone

$$\text{minimize } \mathbf{Tr}(SX) - \log\det X$$

with variable $X \in \mathbf{S}^n$

Canonicalization based on PSD cone:

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(SX) - \sum_{i=1}^n \log Z_{ii} \\ \text{subject to} & \begin{bmatrix} X & Z \\ Z^T & \mathbf{diag}(Z) \end{bmatrix} \succeq 0, \\ & Z \text{ lower triangular,} \end{array}$$

- canonicalization *not* done

Canonicalization based on logdet-cone:

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(SX) + t \\ \text{subject to} & v = 1 \\ & (t, v, X) \in K_{\text{logdet}} \end{array}$$

- canonicalization done

- to use spectral matrix cones within first-order solvers we must be able to **project onto them**

Outline

Conic convex optimization

Spectral matrix cones

Projecting onto spectral matrix cones

Numerical experiments

How to project onto spectral matrix cones?

- for any spectral function $F : \mathbf{S}^n \to \mathbf{R}$ there exists a *symmetric* convex function $f : \mathbf{R}^n \to \mathbf{R}$ such that
$$F(X) = f(\boldsymbol{\lambda}(X))$$
where $\boldsymbol{\lambda}(X) = (\lambda_1(X), \lambda_2(X), \ldots, \lambda_n(X))$ is the vector of eigenvalues in increasing order
- we use this result to reduce the problem of projecting a matrix to the problem of projecting its eigenvalues
- our main results are conceptually similar to many works employing a *general transfer principle*

How to project onto spectral matrix cones?

Let $F : \mathbf{S}^n \to \mathbf{R}$ be a spectral function and $f : \mathbf{R}^n \to \mathbf{R}$ the corresponding symmetric function.

- spectral *matrix* cone:

$$K_F = \mathbf{cl}\{(t, v, X) \in \mathbf{R} \times \mathbf{R}_{++} \times \mathbf{S}^n \mid vF(X/v) \le t\}$$

- spectral *vector* cone:

$$K_f = \mathbf{cl}\{(t, v, x) \in \mathbf{R} \times \mathbf{R}_{++} \times \mathbf{R}^n \mid vf(x/v) \le t\}$$

- relationship:

$$(t, v, X) \in K_F \text{ if and only if } (t, v, \lambda(X)) \in K_f$$

How to project onto spectral matrix cones?

- **Main result:** to compute the projection of $(\bar{t}, \bar{v}, \bar{X})$ onto K_F:
  1. compute spectral decomposition $\bar{X} = U\, \mathbf{diag}(\bar{\lambda}) U^T$
  2. compute the projection of $(\bar{t}, \bar{v}, \bar{\lambda})$ onto K_f, denoted by (t, v, λ)
  3. the projection of $(\bar{t}, \bar{v}, \bar{X})$ onto K_F is $(t, v, U\, \mathbf{diag}(\lambda) U^T)$

How to project onto spectral matrix cones?

- **Main result:** to compute the projection of $(\bar{t}, \bar{v}, \bar{X})$ onto K_F:
  1. compute spectral decomposition $\bar{X} = U\,\mathbf{diag}(\bar{\lambda})U^T$
  2. compute the projection of $(\bar{t}, \bar{v}, \bar{\lambda})$ onto K_f, denoted by (t, v, λ)
  3. the projection of $(\bar{t}, \bar{v}, \bar{X})$ onto K_F is $(t, v, U\,\mathbf{diag}(\lambda)U^T)$

- in the paper we discuss how to project onto spectral vector cones efficiently
  - the projection onto some spectral vector cones can be done by a sort (very cheap)
  - other projections require an iterative solver (still cheap)

Outline

Conic convex optimization

Spectral matrix cones

Projecting onto spectral matrix cones

Numerical experiments

Numerical results

- experiments based on the *splitting conic solver* (SCS) [O'Donoghue 2016, 2021]

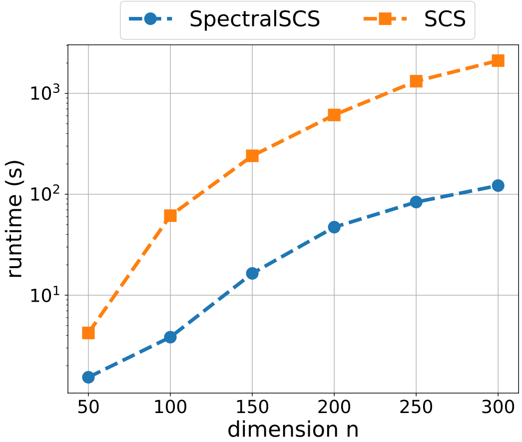
- in every iteration SCS solves a linear system and projects onto a cone
  - Standard SCS uses the PSD cone for canonicalization of matrix functions
  - SpectralSCS uses spectral matrix cones

Minimum-volume ellipsoid containing a set of points

$$
\begin{array}{ll}
\text{minimize} & -\log\det W \\
\text{subject to} & v_i^T W v_i \leq 1, \;\; i = 1, \ldots, p.
\end{array}
$$

- variable $W \in \mathbf{S}^n$ defines a centered ellipsoid $\{v \mid v^T W v \leq 1\}$
- the ellipsoid should cover the given points $v_1, \ldots, v_p \in \mathbf{R}^n$
- SpectralSCS uses the log-determinant cone

Minimum-volume ellipsoid



- on average SpectralSCS is 20 times faster than SCS.

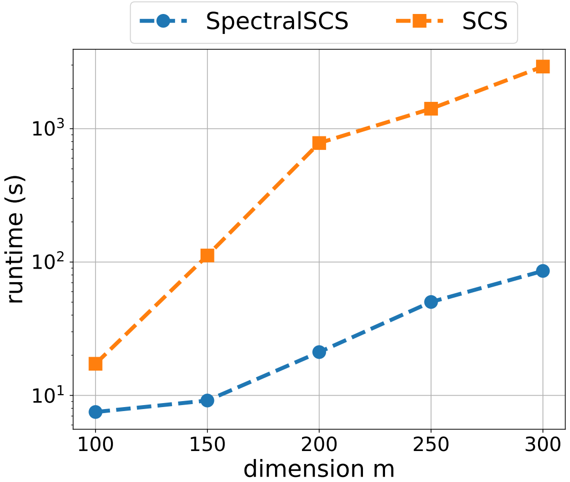
Robust principal component analysis

$$
\begin{array}{ll}
\text{minimize} & \|X\|_* \\
\text{subject to} & \|S\|_1 \leq \mu \\
& X + S = M,
\end{array}
$$

- variables $X \in \mathbf{R}^{m \times n}$, $S \in \mathbf{R}^{m \times n}$ and problem data $\mu \in \mathbf{R}$, $M \in \mathbf{R}^{m \times n}$
- recovers a low rank matrix from measurements M corrupted by sparse noise S
- nuclear norm $\|X\|_* = \sum_{i=1}^{\min\{n,m\}} \sigma_i(X)$ encourages low rank
- SpectralSCS uses the so-called *nuclear norm cone*:

$$
K_{\text{nuc}} = \{(t, X) \mid \|X\|_* \leq t\}
$$

Robust principal component analysis



- on average SpectralSCS is 22 times faster than SCS.

How fast is the spectral vector cone projection?

- to project onto a spectral matrix cone we compute its spectral decomposition and then project the eigenvalues onto a spectral vector cone

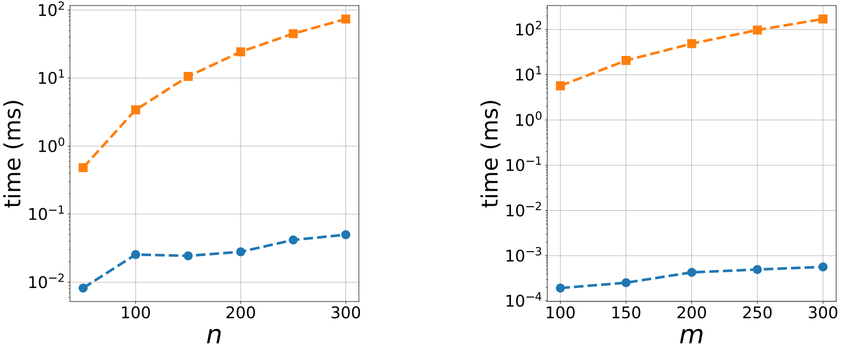
How fast is the spectral vector cone projection?

- to project onto a spectral matrix cone we compute its spectral decomposition and then project the eigenvalues onto a spectral vector cone
- important that the spectral vector cone projection does not incur an overhead

How fast is the spectral vector cone projection?

- to project onto a spectral matrix cone we compute its spectral decomposition and then project the eigenvalues onto a spectral vector cone
- important that the spectral vector cone projection does not incur an overhead
- which part is most expensive?

How fast is the spectral vector cone projection?



- **Left:** Experimental design. **Right:** Robust PCA.
- blue line = time for spectral vector cone projection
- orange line = time for spectral matrix cone projection

Summary

- convex optimization problems are canonicalized to conic form
- first-order solvers project onto the cone in every iteration
- we extend SCS with projections onto spectral matrix cones
- this extension makes SCS significantly faster for several types of SDPs

Thanks!