

# A Newton-CG Augmented Lagrangian Method for Large Scale Semidefinite Programming

Defeng Sun

Department of Mathematics

National University of Singapore

March 12, 2009

Joint work with Kim Chuan Toh and Xin-Yuan Zhao

Let  $\mathcal{S}^n$  be the set of all real symmetric matrices and  $\mathcal{S}_+^n$  be the cone of all positive semidefinite matrices in  $\mathcal{S}^n$ .

We use  $X \succeq 0$  to indicate  $X \in \mathcal{S}_+^n$ .

A matrix  $X \in \mathcal{S}_+^n$  is called a **correlation matrix** if its diagonal elements are all ones.

Trace product:

$$\langle P, Q \rangle = \sum_{i,j} P_{ij} Q_{ij} = \text{Trace}(Q^T P).$$

Given data:  $C, A_1, \dots, A_m \in \mathcal{S}^n, b \in \mathfrak{R}^m$ .

The semidefinite programming (SDP) problem in the primal form:

$$\begin{aligned} \text{(P)} \quad & \max \langle C, X \rangle \\ & \text{s.t. } \mathcal{A}(X) = b, \quad X \succeq 0, \end{aligned}$$

where  $\mathcal{A} : \mathcal{S}^n \rightarrow \mathfrak{R}^m$  is the linear map s.t.

$$\mathcal{A}(X) = \left[ \langle A_1, X \rangle, \dots, \langle A_m, X \rangle \right]^T.$$

Assume (P) is feasible.

Problem dimension:

$n =$  dimension of  $X$ ;

$m =$  number of linear constraints.

We are interested in SDPs with large  $m \geq 10,000$ , but moderate  $n \leq 5,000$ .

Examples of SDPs:

The nearest correlation matrix (NCM) problem:

Given an estimated correlation matrix  $C$ , we want to find a valid correlation matrix  $X$  that is nearest to the data:

$$\min \left\{ \sum_{ij} |X_{ij} - C_{ij}| \quad : \text{diag}(X) = \mathbf{1}, X \succeq 0 \right\}$$

↓

$$\sum_{ij} v_{ij}^+ + v_{ij}^- \quad : X_{ij} - C_{ij} = v_{ij}^+ - v_{ij}^-$$

$$v_{ij}^+ \geq 0, \quad v_{ij}^- \geq 0.$$

In (NCM),  $m = n + n(n + 1)/2$ , which is about a half million when  $n = 1,000$ .

[The introduction of linear inequality constraints does no make much difference to our subsequent analysis]

The fact that the estimated matrix  $C$  is not a valid correlation matrix is due to several situations:

- expert opinions in reinsurance
- stress testing regulated by Basel II

Partial market data<sup>a</sup>

$$C = \begin{bmatrix} 1.0000 & 0.9872 & 0.9485 & 0.9216 & -0.0485 & -0.0424 \\ 0.9872 & 1.0000 & 0.9551 & 0.9272 & -0.0754 & -0.0612 \\ 0.9485 & 0.9551 & 1.0000 & 0.9583 & -0.0688 & -0.0536 \\ 0.9216 & 0.9272 & 0.9583 & 1.0000 & -0.1354 & -0.1229 \\ -0.0485 & -0.0754 & -0.0688 & -0.1354 & 1.0000 & 0.9869 \\ -0.0424 & -0.0612 & -0.0536 & -0.1229 & 0.9869 & 1.0000 \end{bmatrix}$$

The eigenvalues of  $C$  are: 0.0087, 0.0162, 0.0347, 0.1000, 1.9669, and 3.8736.

---

<sup>a</sup>RiskMetrics ([www.riskmetrics.com/stdownload\\_edu.html](http://www.riskmetrics.com/stdownload_edu.html))

Let's change  $C$  to

[change  $C(1, 6) = C(6, 1)$  from  $-0.0424$  to  $-0.1000$ ]

$$\begin{bmatrix} 1.0000 & 0.9872 & 0.9485 & 0.9216 & -0.0485 & -\mathbf{0.1000} \\ 0.9872 & 1.0000 & 0.9551 & 0.9272 & -0.0754 & -0.0612 \\ 0.9485 & 0.9551 & 1.0000 & 0.9583 & -0.0688 & -0.0536 \\ 0.9216 & 0.9272 & 0.9583 & 1.0000 & -0.1354 & -0.1229 \\ -0.0485 & -0.0754 & -0.0688 & -0.1354 & 1.0000 & 0.9869 \\ -\mathbf{0.1000} & -0.0612 & -0.0536 & -0.1229 & 0.9869 & 1.0000 \end{bmatrix}$$

The eigenvalues of  $C$  are:  $-\mathbf{0.0216}$ ,  $0.0305$ ,  $0.0441$ ,  $0.1078$ ,  $1.9609$ , and  $3.8783$ .



## The maximum stable set problem of a graph:

For a graph  $G = (V, \mathcal{E})$ ,

**a stable set  $S$  is a subset of  $V$  such that no vertices in  $S$  are adjacent.**

The problem is to find a stable set with maximum cardinality.

The standard SDP relaxations of the maximum stable set problem are:

$$\theta(G) := \max \left\{ \langle E, X \rangle \quad : \quad X_{ij} = 0 \quad \forall (i, j) \in \mathcal{E}, \right. \\ \left. \langle I, X \rangle = 1, \quad X \succeq 0 \right\}$$

and

$$\theta_+(G) := n(n+1)/2 \text{ additional constraints } X \geq 0$$

$\theta(G)$ : number of constraints  $m = |\mathcal{E}| + 1$ .

$\theta_+(G)$ : number of constraints  $m = |\mathcal{E}| + 1 + n(n+1)/2$ .

## Estimating the Covariance Matrix with Sparsity:

$$\min \left\{ \frac{1}{2} \|\Sigma - S\|_F^2 + \lambda \sum_{i \neq j} w_{ij} |\sigma_{ij}| : \Sigma \succeq 0 \right\},$$

where  $S$  is the sample variance matrix and  $w_{ij} > 0$  are given weights.

Without the positive semidefinite constraint on  $\Sigma$ , the estimator is simply a soft thresholding version of  $S$ .

Recall that

$$\begin{aligned} \text{(P)} \quad & \max \langle C, X \rangle \\ & \text{s.t. } \mathcal{A}(X) = b, \quad X \succeq 0, \end{aligned}$$

where  $\mathcal{A} : \mathcal{S}^n \rightarrow \mathfrak{R}^m$  is a linear map.

The dual problem of (P) is

$$\text{(D)} \quad \min \left\{ b^T y \mid \mathcal{A}^* y - C \succeq 0 \right\},$$

where  $\mathcal{A}^* : \mathfrak{R}^m \rightarrow \mathcal{S}^n$  is the adjoint of  $\mathcal{A}$ .

One obvious choice for solving (P) and (D) is the **interior point method** (IPM). Indeed, much progress of research of SDP is largely credited to the discovery of polynomial time IPMs.

But, we do have one difficulty:

At each iteration, the primal-dual IPMs need to formulate and solve a linear system with a dense Schur complement matrix of size  $m$  by  $m$ . This limits the problems to be of size  **$m$  smaller than 5,000**. For the NCM problem,  $n$  must be less than 100.

This means we have to look for other methods because in our cases  $m \geq 10,000$ .

## Related approaches:

- First-order methods (low accuracy): convergence?
- Inexact IPM  $\leftarrow$  compute direction via iterative solvers [Toh, Kojima]
- Shifted barrier method [Kocvara-Stingl]: theoretical? How much different from barrier methods?
- Augmented Lagrangian method for primal SDPs from relaxation of lift-and-project scheme [Burer-Vandenbussche]
- Boundary-point method: based on augmented Lagrangian method for (D) [Rendl et al. ]

Given a penalty parameter  $\sigma > 0$ , the *augmented Lagrangian* function for problem (D) is defined as

$$L_\sigma(y, X) = b^T y + \frac{1}{2\sigma} \left( \|\Pi_{\mathcal{S}_+^n}(X - \sigma(\mathcal{A}^*y - C))\|^2 - \|X\|^2 \right),$$

where  $(y, X) \in \mathfrak{R}^m \times \mathcal{S}^n$  and for any  $X \in \mathcal{S}^n$ ,  $\Pi_{\mathcal{S}_+^n}(X)$  is the unique optimal solution to

$$\begin{aligned} \min \quad & \frac{1}{2} \|Z - X\|^2 \\ \text{s.t.} \quad & Z \in \mathcal{S}_+^n. \end{aligned}$$

For  $K = \mathcal{S}_+^n$ ,

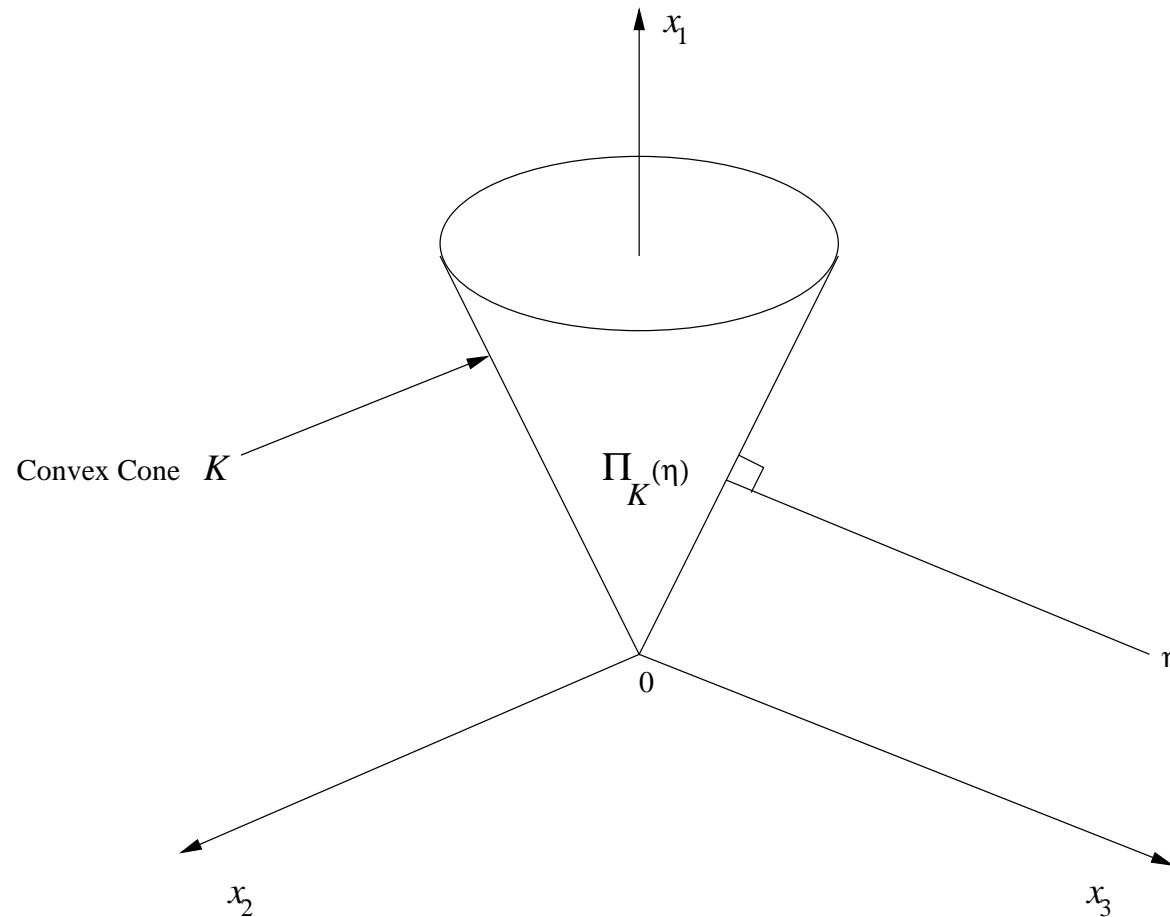


Figure 0.1: Metric projection onto closed convex sets



Let  $X \in \mathcal{S}^n$  have the following spectral decomposition

$$X = P\Lambda P^T,$$

where  $\Lambda$  is the diagonal matrix of eigenvalues of  $X$  and  $P$  is a corresponding orthogonal matrix of orthonormal eigenvectors. Then

$$X_+ := P_{\mathcal{S}_+^n}(X) = P\Lambda_+P^T.$$

- Note that computing  $X_+$  is equivalent to computing the full eigen-decomposition of  $X$ , which in turn needs  $9n^3$  flops [The divide and conquer method needs  $4n^3$ , which we use]
- For my Dell Laptop, it needs about 1.2 seconds for  $n = 1000$ , about 8.2 seconds for  $n = 2000$ , less than 28 seconds for  $n = 3000$ , and slightly over 60 seconds for  $n = 4000$ .
- For semidefinite optimization, at each step  $O(n^3)$  cost is not a problem.

Define

$$\alpha := \{i : \lambda_i > 0\}, \beta := \{i : \lambda_i = 0\}, \gamma := \{i : \lambda_i < 0\}.$$

Write

$$\Lambda = \begin{bmatrix} \Lambda_\alpha & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Lambda_\gamma \end{bmatrix} \quad \text{and} \quad P = [P_\alpha \ P_\beta \ P_\gamma].$$

Define  $\Omega \in \mathcal{S}^n$ :

$$\Omega_{ij} := \frac{\max\{\lambda_i, 0\} + \max\{\lambda_j, 0\}}{|\lambda_i| + |\lambda_j|}, \quad i, j = 1, \dots, n,$$

where  $0/0$  is defined to be 1.

$\Pi_{\mathcal{S}_+^n}$  is directionally differentiable with  $\Pi'_{\mathcal{S}_+^n}(X; H)$  being given by

$$P \begin{bmatrix} P_\alpha^T H P_\alpha & P_\alpha^T H P_\beta & \Omega_{\alpha\gamma} \circ P_\alpha^T H P_\gamma \\ P_\beta^T H P_\alpha & \Pi_{\mathcal{S}_+^{|\beta|}}(P_\beta^T H P_\beta) & 0 \\ P_\gamma^T H P_\alpha \circ \Omega_{\alpha\gamma}^T & 0 & 0 \end{bmatrix} P^T.$$

When  $|\beta| = 0$ ,  $\Pi_{\mathcal{S}_+^n}(\cdot)$  is continuously differentiable around  $X$  and the above formula reduces to the classical result of Löwner<sup>a</sup>:

$$\Pi'_{\mathcal{S}_+^n}(X)H = P \begin{bmatrix} P_\alpha^T H P_\alpha & \Omega_{\alpha\gamma} \circ P_\alpha^T H P_\gamma \\ P_\gamma^T H P_\alpha \circ \Omega_{\alpha\gamma}^T & 0 \end{bmatrix} P^T .$$

---

<sup>a</sup>K. LÖWNER *Über monotone matrixfunktionen.* Mathematische Zeitschrift 38 (1934) 177–216.

Note that (D) can be written as

$$(D') \quad \min \left\{ b^T y \mid \mathcal{A}^* y - C - Z = 0, \quad Z \succeq 0 \right\}.$$

The augmented Lagrangian function  $L_\sigma(y, X)$  can then be obtained in a simple way:

$$\begin{aligned} & L_\sigma(y, X) \\ &= \inf_{Z \succeq 0} \left\{ \underline{b^T y + \langle X, Z - (\mathcal{A}^* y - C) \rangle + \frac{\sigma}{2} \|Z - (\mathcal{A}^* y - C)\|^2} \right\} \\ &= b^T y + \frac{1}{2\sigma} \left( \|\Pi_{\mathcal{S}_+^n}(X - \sigma(\mathcal{A}^* y - C))\|^2 - \|X\|^2 \right). \end{aligned}$$

The augmented Lagrangian function is continuously differentiable. For any given  $X \in \mathcal{S}_+^n$ , we have

$$\nabla_y L_\sigma(y, X) = b - \mathcal{A}\Pi_{\mathcal{S}_+^n}(X - \sigma(\mathcal{A}^*y - C)).$$

For given  $X^0 \in \mathcal{S}^n$ ,  $\sigma_0 > 0$ , and  $\rho > 1$ , the augmented Lagrangian method for solving problem (D) and its dual (P) generates sequences  $\{y^k\} \subset \Re^m$  and  $\{X^k\} \subset \mathcal{S}^n$  as follows

$$\left\{ \begin{array}{l} y^{k+1} \approx \arg \min_{y \in \mathfrak{R}^m} L_{\sigma_k}(y, X^k), \\ X^{k+1} = \Pi_{\mathcal{S}_+^n}(X^k - \sigma_k(\mathcal{A}^* y^{k+1} - C)), \quad k = 0, 1, 2, \dots \\ \sigma_{k+1} = \rho \sigma_k \text{ or } \sigma_{k+1} = \sigma_k, \end{array} \right.$$



## Why the augmented Lagrangian method?

Consider the nearest correlation matrix problem under the  $F$ -norm:

$$\begin{aligned} \min \quad & \frac{1}{2} \|X - G\|_F^2 \\ \text{s.t.} \quad & X_{ii} = 1, \quad i = 1, \dots, n, \\ & X \in \mathcal{S}_+^n, \end{aligned}$$

where  $G \in \mathcal{S}^n$  is given, but may not be positive semidefinite.

The Lagrangian **dual** takes the form

$$\max_{y \in \mathbb{R}^n} -\theta(y) := -\frac{1}{2} \|\Pi_{\mathcal{S}_+^n}(G + \mathcal{A}^*y)\|^2 + e^T y,$$

where  $\Pi_{\mathcal{S}_+^n}(\cdot)$  denotes the metric projection operator onto  $\mathcal{S}_+^n$  and  $\mathcal{A}^*$  is the **adjoint** of  $\mathcal{A}$ :

$$\mathcal{A}^*(y) = \text{Diag}(y) \quad \text{with} \quad \mathcal{A}(X) = \text{diag}(X),$$

$$\nabla\theta(y) = \mathcal{A}\Pi_{\mathcal{S}_+^n}(G + \mathcal{A}^*y) - e.$$

The above Lagrangian dual is a SPECIAL case of the inner problems in the aug. Lagrangian m. for SDPs.

The former can be solved very efficiently by a semismooth Newton-CG method [H.D. Qi and Sun, SIMAX 28 (2006) 360–385.]

— For randomly generated problems,

for  $n = 1000$ , it takes 9 Secs;

for  $n = 2000$ , it takes 78 Secs;

for  $n = 4000$ , it takes 595 Secs.

Rockafellar (1976) made a marvelous achievement on the augmented Lagrangian method for solving convex optimization problems.

The augmented Lagrangian method for convex problems is a **gradient ascent method** applied to the corresponding augmented Lagrangian dual problems

$$\max_{X \in \mathcal{S}^n} \psi_\sigma(X) := \inf_{y \in \mathcal{R}^m} L_\sigma(y, X) = L_\sigma(y(X), X).$$

But, recent studies reveal that under the constraint nondegenerate conditions for (P) and (D)[LICQs], the augmented Lagrangian method for solving SDPs is actually

an approximate semismooth Newton method.

This motivates us to take a closer look at the augmented lagrangian method.

A semismooth Newton-CG method for solving inner subproblem

We need to solve

$$\nabla_y L_{\sigma_k}(y, X^k) = b - \mathcal{A}\Pi_{\mathcal{S}_+^n}(U^k(y)) = 0.$$

where  $U^k(y) := X^k - \sigma_k(\mathcal{A}^*y - C)$ .

The mapping  $\nabla_y L(y, X^k)$  is not differentiable, but is **strongly semismooth**. At a current iterate  $y$ , we solve a generalized Newton equation:

$$\mathcal{H}_y := \sigma_k \mathcal{A}\Pi'_{\mathcal{S}_+^n}(U^k(y))\mathcal{A}^*, \quad \mathcal{H}_y \Delta y = -\nabla_y L(y, X^k).$$

Let  $U^k(y) = PDP^T$  with

$\lambda_1 \geq \dots \geq \lambda_r > 0 \geq \lambda_{r+1} \geq \dots \geq \lambda_n$ . We have

$$\Pi'_{\mathcal{S}_+^n}(U^k(y))H = P(\Omega \circ (P^T H P))P^T.$$

For  $\alpha = \{1, \dots, r\}$  and  $\gamma = \{r + 1, \dots, n\}$ , we have

$$\Omega = \begin{bmatrix} E_{\alpha\alpha} & \Omega_{\alpha\gamma} \\ \Omega_{\alpha\gamma}^T & 0 \end{bmatrix}.$$

The (1,1) and (2,2) blocks in  $\Omega$  allow for efficient computation of  $\mathcal{H}_y \Delta y$ .

The key issue is that  $\mathcal{H}_y$  may be good conditioned while for IPM, the Schur complement matrix  $\mathcal{M}$  at a point on the central path will become more and more ill conditioned when the parameter goes to zero.

Moreover,

The cost for computing  $\mathcal{H}_y \Delta y$  is

$$= 8 \min\{r, n - r\} n^2 + \text{cost}(\mathcal{A}(\cdot)) + \text{cost}(\mathcal{A}^*(\cdot))$$

and the cost for computing  $\mathcal{M} \Delta y$  is

$$= 4n^3 + \text{cost}(\mathcal{A}(\cdot)) + \text{cost}(\mathcal{A}^*(\cdot)).$$



## Practical Newton-CG augmented Lagrangian method [SDPNAL]

- Solve  $\mathcal{H}_y \Delta y = \text{rhs}$  by CG with a diagonal preconditioner.

Stop when **relative-residual**  $\leq 0.01$ .

- Stop the inner iteration when 
$$\|\nabla_y L(y^k, X^k)\| \leq 0.2 \|X^{k+1} - X^k\|.$$
- Typically SDPNAL needs 30-50 outer iterations, and each requires 5 – 30 Newton steps to solve the inner subproblem.

In contrast, IPM requires about 30-50 iterations each uses only 2 Newton steps.

For the boundary-point method of Rendl et al., one step of modified gradient method is used to solve the inner subproblem:

$$y^k = y^{k-1} - (\sigma_k \mathcal{A}\mathcal{A}^*)^{-1} \nabla_y L(y^{k-1}, X^k).$$

Numerical results:

$$\text{want: rel-err} = \max \left\{ \frac{\|R_p\|}{1+\|b\|}, \frac{\|R_d\|}{1+\|C\|}, \frac{\langle X, Z \rangle}{1+|\langle C, X \rangle|+|b^T y|} \right\} \leq 10^{-6}.$$

PC: Intel Xeon 3.2GHz with 4G RAM, MATLAB

	parallel IPM 64 nodes 2.4GHz PCs	boundary point method	SDPNAL
$\theta$ :theta62 $m = 13390$ $n = 300$	459s		20 14s
$\theta$ :theta82 $m = 23872$ $n = 400$	2403s		21 27s

	boundary point method	SDPNAL
$\theta$ :G43 $m = 9991$ $n = 1000$	2000 3900s $1.5e-5$	18 99s
Rn8m100P3 $m = 100K$ $n = 800$	135 256s	11 1129s

	boundary point method	SDPNAL
fap36 $m = 1.15 \times 10^6$ $n = 4110$	2000 47h $1.4 \times 10^{-4}$	17 65.5h
QAP:lipa40a $m = 1.28 \times 10^6$ $n = 1600$		22 21h

	boundary point method	SDPNAL
$\theta_+ : \text{lzc.2048}$ $m = 2.14 \times 10^6$ $n = 2048$		11 2.2h
$\theta : \text{2dc.512}$ $m = 54896$ $n = 512$		27 1936s $2.4e-5$

## Summary:

- We have tested SDPNAL on about 400 SDPs from  $\theta$ ,  $\theta_+$ , FAP, QAP, binary QP.
- When the SDPs are primal-dual nondegenerate, SDPNAL can efficiently solve large SDPs to rather high accuracy.
- For SDPs with degeneracies, relative primal infeasibilities can range from  $10^{-6}$  to  $10^{-3}$ , while relative dual infeasibilities are  $< 10^{-6}$ .