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

Defeng Sun

Department of Mathematics

National University of Singapore

October 2, 2008

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 $X \succeq 0$.

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

Trace product:

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

Given data: $C, A_1, \cdots, A_m \in \mathcal{S}^n$, $b \in \Re^m$.

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

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

where $\mathcal{A} : \mathcal{S}^n \to \Re^m$ is the linear map s.t.

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

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 $n \leq 2,000$.

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:

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

$$\Downarrow$$

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

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

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[a]

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

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

[a] RiskMetrics (`www.riskmetrics.com/stddownload_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: -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 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:

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

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:

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

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

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

Recall that

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

where $\mathcal{A} : \mathcal{S}^n \to \Re^m$ is a linear map.

The dual problem of (P) is

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

where $\mathcal{A}^* : \Re^m \to \mathcal{S}^n$ 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 70.

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 [Kojima, Toh]
- Shifted barrier method [Kocvara-Stingl]
- 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

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

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

$$\begin{array}{ll} \min & \dfrac{1}{2}\|Z - X\|^2 \\ \text{s.t.} & Z \in \mathcal{S}^n_+ . \end{array}$$

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

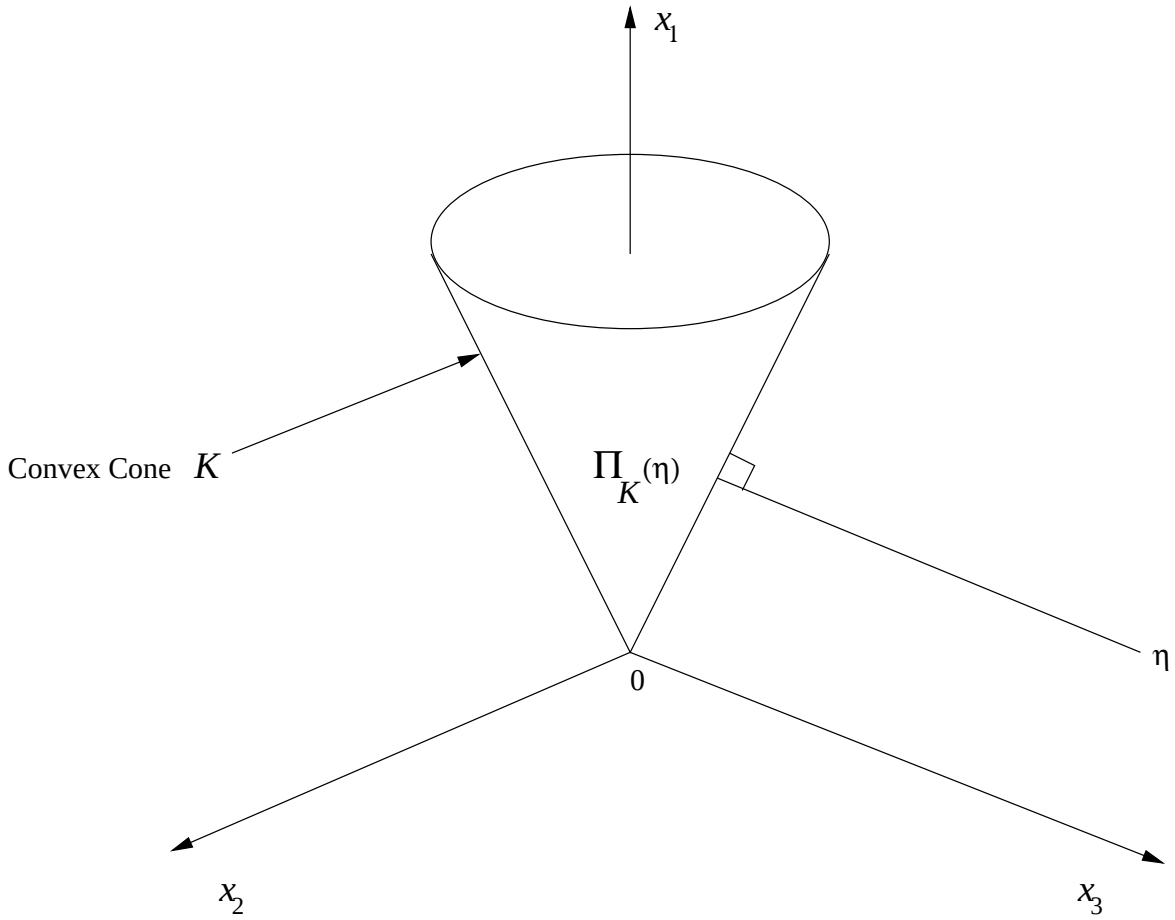

Figure 0.1: Metric projection onto closed convex sets

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

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

where Λ 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 methods needs $4n^3$]

• For my Dell Laptop, it needs about 5 or 6 seconds for $n = 1,000$, about 45 seconds for $n = 2,000$, and less than 155 seconds for $n = 3,000$.

• 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

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

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

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

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[a]:

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

[a] K. Löwner *Über monotone matrixfunktionen.* Mathematische Zeitschrift 38 (1934) 177–216.

Note that (D) can be written as

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

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

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

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

$$\begin{cases} y^{k+1} \approx \arg\min\limits_{y\in\Re^m} L_{\sigma_k}(y, X^k), \\ X^{k+1} = \Pi_{\mathcal{S}^n_+}(X^k - \sigma_k(\mathcal{A}^* y^{k+1} - C)), \ \ k = 0, 1, 2, \ldots \\ \sigma_{k+1} = \rho\sigma_k \ \text{or} \ \sigma_{k+1} = \sigma_k, \end{cases}$$

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

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

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

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

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

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

The (1,1) and (2,2) blocks in Ω 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

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

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

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

Practical Newton-CG ALM

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

  Stop when `relative-residual` ≤ 0.01.

- Stop the inner iteration when $\|\nabla_y L(y^k, X^k)\| \leq 0.2\|X^{k+1} - X^k\|$.

- Typically ALM 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:

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 7.3

	parallel IPM 64 nodes 2.4GHz PCs	boundary point method	NCG-ALM
θ:theta62 $m = 13390$ $n = 300$	459s	223 95s	20 32s
θ:theta82 $m = 23872$ $n = 400$	2403s	236 228s	21 73s

	boundary point method	NCG-ALM
θ:G43	2000	16
$m = 9991$	7.5h	15m
$n = 1000$	1.2e-5	
NCM:400H1	2000	22
$m = 80.6K$	1944s	539s
$n = 400$	3.1e-6	

	boundary point method	NCG-ALM
Rn8m100P3 $m = 100K$ $n = 800$	135 17m	11 27m
QAP:lipa40a m=1.28×10^6 $n = 1600$		22 19h

	boundary point method	NCG-ALM
θ_+:lzc.2048 $m = 2.14 \times 10^6$ $n = 2048$		11 3.6h
θ:2dc.512 $m = 54896$ $n = 512$		27 2400s 2.2e-5

Summary:

- We have tested NCG-ALM on about 400 SDPs from θ, θ_+, NCM, QAP, binary QP.
- When the SDPs are primal-dual nondegenerate, NCG-ALM can efficiently solve large SDPs to rather high accuarcy.
- For SDPs with degeneracies, relative primal infeasibilities can range from 10^{-6} to 10^{-3}, while relative dual infeasibilities are $< 10^{-6}$.