Calibrating Least Squares Covariance Matrix Problems with Equality and Inequality Constraints

Defeng Sun

Department of Mathematics

National University of Singapore

This talk in based on joint work with Yan Gao at NUS

♡ Happy 60th birthday, my teacher ♡

Dedicated to Professor Bingsheng He

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 $\mathcal{S}^n$.

A symmetric matrix $X \in \mathcal{S}^n$ is called a covariance matrix if $X \in \mathcal{S}^n_+$, i.e., $X \succeq 0$.

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

In finance and statistics, covariance matrices are in many situations found to be inconsistent, i.e., $X \not\succeq 0$.

These include, but are not limited to,

- expert opinions in reinsurance
- stress testing regulated by Basel II
- structured statistical estimations, and etc.

Partial market data[a]

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

The eigenvalues of G 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 G to

[change $G(1,6) = G(6,1)$ 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 G are: -0.0216, 0.0305, 0.0441, 0.1078, 1.9609, and 3.8783.

We are interested in the following least squares covariance matrix (LSCM) problem

$$\begin{array}{ll} \min & \frac{1}{2}\|X - C\|^2 \\ \text{s.t.} & \langle A_i, X\rangle = b_i, \quad i = 1, \ldots, p\,, \\ & \langle A_i, X\rangle \geq b_i, \quad i = p+1, \ldots, m\,, \\ & X \in \mathcal{S}^n_+\,, \end{array}$$

where $\|\cdot\|$ is the Frobenius norm induced by the standard trace inner product $\langle \cdot, \cdot \rangle$ in $\mathcal{S}^n$, C and A_i, $i = 1, \ldots, m$ are given matrices in $\mathcal{S}^n$, and $b \in \Re^m$.

Mathematically, the LSCM problem can be equivalently written as

$$\begin{array}{ll} \min & t \\ \text{s.t.} & \langle A_i, X\rangle = b_i, \quad i = 1, \ldots, p\,, \\ & \langle A_i, X\rangle \geq b_i, \quad i = p+1, \ldots, m\,, \\ & t+1 \geq \sqrt{\,(t-1)^2 + 2\|X-C\|^2\,}\,, \\ & X \in \mathcal{S}^n_+\,. \end{array}$$

So we have is a linear optimization problem with linear equality/inequality, the second order cone, and the positive semidefinite cone constraints.

We may use publicly available softwares, based on interior point methods (IPMs), such as SeDuMi and SDPT3 to solve the LSCM problem, directly.

- This is indeed feasible on a Pentium IV PC as long as n is small (say 80 at most) and m is not too large (say $5,000$).

- IPMs need at each iteration to formulate and solve a linear system with a dense Schur complement matrix of the size $(m+1+\bar{n}) \times (m+1+\bar{n})$, where $\bar{n} := \frac{1}{2}n(n+1)$.

- So, how about large n and m? First order methods?

The LSCM problem is a special case of the

best approximation problem

$$\begin{array}{ll} \min & \frac{1}{2}\|x-c\|^2 \\ \text{s.t.} & \mathcal{A}x \in b+Q\,, \\ & x \in K\,, \end{array}$$

where $\mathcal{X}$ is a real Hilbert space equipped with a scalar product $\langle \cdot, \cdot \rangle$ and its induced norm $\| \cdot \|$, $\mathcal{A} : \mathcal{X} \to \Re^m$ is a bounded linear operator, $Q = \{0\}^p \times \Re^q_+$ is a polyhedral convex cone, $1 \leq p \leq m$, $q = m - p$, and K is a closed convex cone in $\mathcal{X}$.

The Karush-Kuhn-Tucker conditions for the **best approximation problem** problem are:

$$\left\{\begin{array}{l} x - c - \mathcal{A}^* y + z = 0 \\ Q^+ \ni y \perp \mathcal{A}x - b \in Q \\ K^o \ni z \perp x \in K \end{array}\right. ,$$

where $Q^+ = \Re^p \times \Re^q_+$ is the dual cone of Q and K^o is the polar of K.

Equivalently,

$$\left\{\begin{array}{l} (x+z) - c + \mathcal{A}^* y = 0 \\ Q^+ \ni y \perp \mathcal{A}x - b \in Q \\ x - \Pi_K(x+z) = 0 \end{array}\right. ,$$

where $\Pi_K(x)$ is the unique optimal solution to

$$\begin{array}{ll} \min & \frac{1}{2}\|u - x\|^2 \\ \text{s.t.} & u \in K\,. \end{array}$$

Consequently, by first eliminating $(x+z)$ and then x, we get

$$Q^+ \ni y \perp \mathcal{A}\Pi_K(c+\mathcal{A}^*y) - b \in Q\,,$$

which is equivalent to

$$F(y) := y - \Pi_{Q^+}[y - (\mathcal{A}\Pi_K(c+\mathcal{A}^*y) - b)] = 0, \quad y \in \Re^m\,.$$

The above is nothing but the first order optimality condition to the convex dual problem

$$\begin{array}{ll} \max & -\theta(y) := -\left[\frac{1}{2}\|\Pi_K(c+\mathcal{A}^*y)\|^2 - \langle b, y\rangle - \frac{1}{2}\|c\|^2\right] \\ \text{s.t.} & y \in Q^+ \,. \end{array}$$

Then F can be written as

$$F(y) = y - \Pi_{Q^+}(y - \nabla\theta(y))\,.$$

Now, we only need to solve

$$F(y) = 0, \quad y \in \Re^m\,.$$

- But, F involves two metric projection operators.

- Even if F is differentiable at y, it is too costly to compute $F'(y)$.

Let consider $K = \mathcal{S}^n_+$.

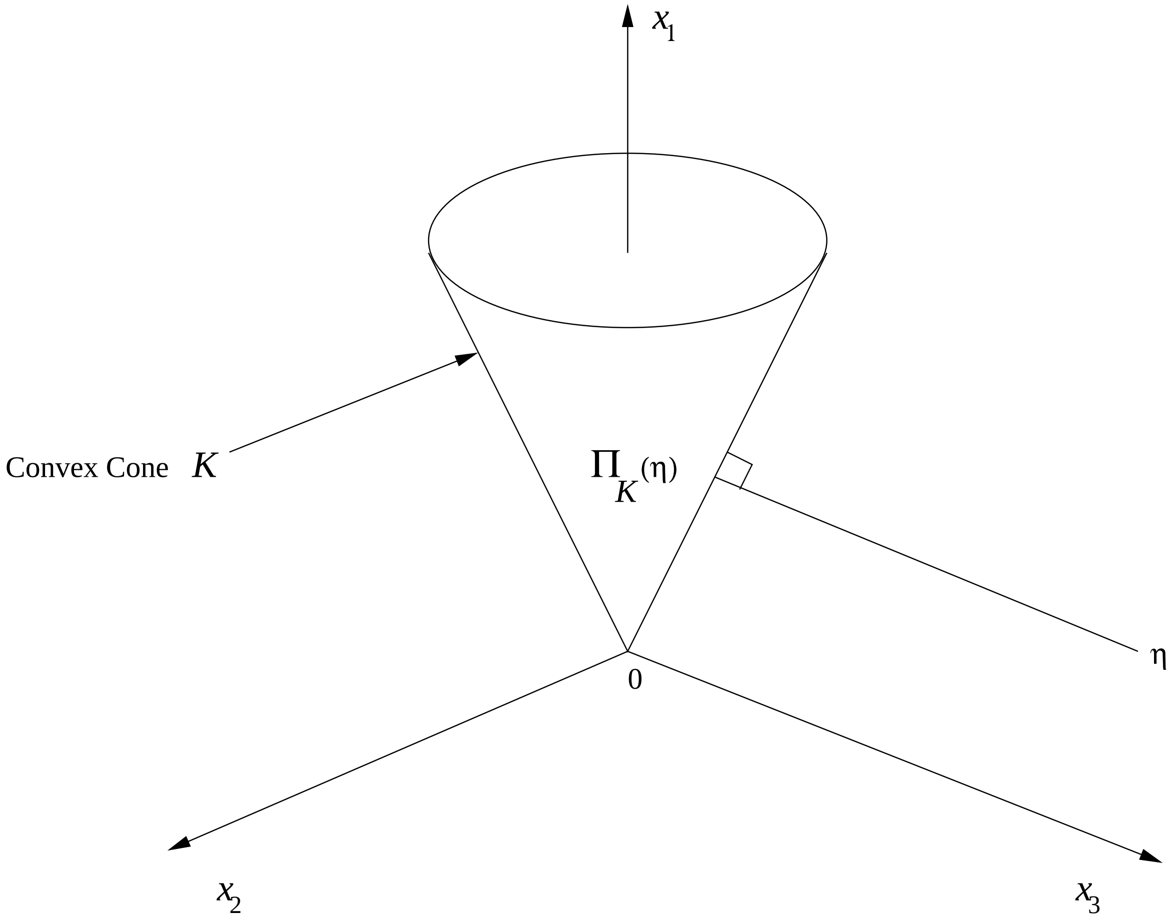

Figure 0.1: Metric projection onto closed convex sets

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.

- 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

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

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

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

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]:

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

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

Let $\phi : \Re \times \Re \to \Re$ be defined by

$$\phi(\varepsilon, t) = \left[t + \sqrt{\varepsilon^2 + t^2} \right]/2, \quad (\varepsilon, t) \in \Re \times \Re\,. \tag{1}$$

For any $\varepsilon \in \Re$, let

$$\Phi(\varepsilon, X) := P \begin{bmatrix} \phi(\varepsilon, \lambda_1) & & \\ & \ddots & \\ & & \phi(\varepsilon, \lambda_n) \end{bmatrix} P^T\,. \tag{2}$$

Then, by matrix analysis, we have

$$\Phi(\varepsilon, X) = \left[\, X + \sqrt{\varepsilon^2 I + X^2}\, \right]/2,$$

where we use I to represent the identity matrix of appropriate dimension. Note that when $\varepsilon = 0$, $\Phi(0, X) = \Pi_{\mathcal{S}^n_+}(X)$.

By the above famous result of Löwner, we know that when $\varepsilon \neq 0$ or $\beta = \emptyset$,

$$\Phi'_X(\varepsilon, X)(H) = P[\Omega(\varepsilon, \lambda) \circ (P^T H P)]P^T \quad \forall\, H \in \mathcal{S}^n\,,$$

where " $\circ$ " denotes the Hadamard product, $\lambda = (\lambda_1, \ldots, \lambda_n)^T$, and the symmetric matrix $\Omega(\varepsilon, \lambda)$ is given by

$$\big[\,\Omega(\varepsilon,\lambda)\,\big]_{ij} = \begin{cases} \dfrac{\phi(\varepsilon,\lambda_i) - \phi(\varepsilon,\lambda_j)}{\lambda_i - \lambda_j} \in [0,1] & \text{if } \lambda_i \neq \lambda_j\,, \\ \phi'_{\lambda_i}(\varepsilon,\lambda_i) \in [0,1] & \text{if } \lambda_i = \lambda_j\,. \end{cases}$$

When $\varepsilon \neq 0$ or $\beta = \emptyset$, the partial derivative of $\Phi(\cdot,\cdot)$ with respect to ε can be computed by

$$\Phi'_\varepsilon(\varepsilon, X) = P\mathrm{diag}(\phi'_\varepsilon(\varepsilon,\lambda_1), \cdots, \phi'_\varepsilon(\varepsilon,\lambda_n))P^T\,.$$

Let $\psi : \Re \times \Re^m \to \Re^m$ be defined by

$$\psi_i(\varepsilon, z) = \begin{cases} z_i & \text{if } i = 1, \ldots, p, \\ \phi(\varepsilon, z_i) & \text{if } i = p+1, \ldots, m, \end{cases} \quad (\varepsilon, z) \in \Re \times \Re^m .$$

The function ψ is obviously continuously differentiable around any $(\varepsilon, z) \in \Re \times \Re^m$ as long as $\varepsilon \neq 0$ and is strongly semismooth everywhere.

Now, we are ready to define a smoothing function for $F(\cdot)$ itself. Let

$$\Upsilon(\varepsilon, y) := y - \psi\left(\varepsilon, y - (\mathcal{A}\Phi(\varepsilon, C + \mathcal{A}^* y) - b)\right),$$

where $(\varepsilon, y) \in \Re \times \Re^m$.

By the definitions of Υ, ψ, and Φ, we know that for any $y \in \Re^m$, $F(y) = \Upsilon(0, y)$.

In general, let $G : \Re \times \Re^m \to \Re^m$ be a locally Lipschitz continuous function satisfying

$$G(\varepsilon, y') \to F(y) \quad \text{as} \quad (\varepsilon, y') \to (0, y)\,.$$

Furthermore, G is required to be continuously differentiable around any (ε, y) unless $\varepsilon = 0$. The existence of such a function G can be easily proven via convolution.

Define $E : \Re \times \Re^m \to \Re \times \Re^m$ by

$$E(\varepsilon, y) := \left[\begin{array}{c} \varepsilon \\ G(\varepsilon, y) \end{array} \right], \quad (\varepsilon, y) \in \Re \times \Re^m\,.$$

Then solving the nonsmooth equation

$$F(y) = 0$$

is equivalent to solving the following smoothing-nonsmooth equation

$$E(\varepsilon, y) = 0\,.$$

Define the merit function $\varphi : \Re \times \Re^m \longrightarrow \Re_+$ by

$$\varphi(\varepsilon, y) := \|E(\varepsilon, y)\|^2 \,, \quad (\varepsilon, y) \in \Re \times \Re^m \,.$$

Choose $r \in (0, 1)$. Let

$$\zeta(\varepsilon, y) := r \min\{1, \varphi(\varepsilon, y)\} \,, \quad (\varepsilon, y) \in \Re \times \Re^m \,.$$

(An inexact smoothing Newton method)

Step 0. Let $\hat{\varepsilon} \in (0, \infty)$ and $\eta \in (0,1)$ be such that

$$\delta := \sqrt{2}\max\{r\hat{\varepsilon}, \eta\} < 1\,.$$

Select constants $\rho \in (0,1)$, $\sigma \in (0, 1/2)$, $\tau \in (0,1)$, and $\hat{\tau} \in [1, \infty)$. Let $\varepsilon^0 := \hat{\varepsilon}$ and $y^0 \in \Re^m$ be an arbitrary point. $k := 0$.

Step 1. Compute $\zeta_k := r\min\{1, \varphi(\varepsilon^k, y^k)\}$ and $\eta_k := \min\{\tau, \hat{\tau}\|E(\varepsilon^k, y^k)\|\}\,.$

Step 2. Solve the following equation

$$E(\varepsilon^k, y^k) + E'(\varepsilon^k, y^k) \begin{bmatrix} \Delta\varepsilon^k \\ \Delta y^k \end{bmatrix} = \begin{bmatrix} \zeta_k \hat{\varepsilon} \\ 0 \end{bmatrix}$$

approximately such that

$$\|R_k\| \leq \min\{\eta_k \|G(\varepsilon^k, y^k) + G'_\varepsilon(\varepsilon^k, y^k)\Delta\varepsilon^k\|, \, \eta\|E(\varepsilon^k, y^k)\|\},$$

where

$$\Delta\varepsilon^k := -\varepsilon^k + \zeta_k \hat{\varepsilon}$$

and

$$R_k := G(\varepsilon^k, y^k) + G'(\varepsilon^k, y^k) \begin{bmatrix} \Delta\varepsilon^k \\ \Delta y^k \end{bmatrix}.$$

Step 3. Let l_k be the smallest nonnegative integer l satisfying

$$\varphi(\varepsilon^k + \rho^l \Delta\varepsilon^k, y^k + \rho^l \Delta y^k) \le [1 - 2\sigma(1-\delta)\rho^l]\varphi(\varepsilon^k, y^k)\,.$$

Define:

$$(\varepsilon^{k+1}, y^{k+1}) := (\varepsilon^k + \rho^{l_k}\Delta\varepsilon^k, y^k + \rho^{l_k}\Delta y^k)\,.$$

Step 4. Replace k by $k+1$ and go to **Step 1**.

Theorem 1 (global convergence). Suppose that for any $(\varepsilon, y) \in \Re_{++} \times \Re^n$, $E'(\varepsilon, y)$ is nonsingular.

Then our inexact smoothing Newton method is well defined and generates an infinite sequence $\{(\varepsilon^k, y^k)\} \in \mathcal{N}$ with the property that any accumulation point $(\bar{\varepsilon}, \bar{y})$ of $\{(\varepsilon^k, y^k)\}$ is a solution of $E(\varepsilon, y) = 0$.

Theorem 2 (local convergence). Suppose that for any $(\varepsilon, y) \in \Re_{++} \times \Re^n$, $E'(\varepsilon, y)$ is nonsingular.

Let $(\bar{\varepsilon}, \bar{y})$ be an accumulation point of the generated infinite sequence $\{(\varepsilon^k, y^k)\}$. Suppose that E is strongly semismooth at $(\bar{\varepsilon}, \bar{y})$ and that all $V \in \partial_B E(d\bar{\varepsilon}, \bar{y})$ are nonsingular. Then the whole sequence $\{(\varepsilon^k, y^k)\}$ converges to $(\bar{\varepsilon}, \bar{y})$ quadratically, i.e.,

$$\Big\|(\varepsilon^{k+1} - \bar{\varepsilon}, y^{k+1} - \bar{y})\Big\| = O\Big(\|(\varepsilon^k - \bar{\varepsilon}, y^k - \bar{y})\|^2\Big) .$$

Now consider the LSCM problem. Let $\kappa \in (0, \infty)$ be a constant. Define $G : \Re \times \Re^m \to \Re^m$ by

$$G(\varepsilon, y) := \Upsilon(\varepsilon, y) + \kappa|\varepsilon|y\,, \quad (\varepsilon, y) \in \Re \times \Re^m\,.$$

Let $E : \Re \times \Re^m \to \Re \times \Re^m$ be defined by

$$E(\varepsilon, y) := \begin{bmatrix} \varepsilon \\ G(\varepsilon, y) \end{bmatrix} = \begin{bmatrix} \varepsilon \\ \Upsilon(\varepsilon, y) + \kappa|\varepsilon|y \end{bmatrix}.$$

Theorem $1'$ (global convergence). The inexact smoothing Newton method is well defined and generates an infinite sequence $\{(\varepsilon^k, y^k)\}$ with the properties that any accumulation point $(\bar{\varepsilon}, \bar{y})$ of $\{(\varepsilon^k, y^k)\}$ is a solution of $E(\varepsilon, y) = 0$ and $\lim_{k\to\infty} \varphi(\varepsilon^k, y^k) = 0$.

Additionally, if the generalized Slater condition holds, then $\{(\varepsilon^k, y^k)\}$ is bounded.

Theorem $2'$ (local convergence). Let $(\bar{\varepsilon}, \bar{y})$ be an accumulation point of generated the infinite sequence $\{(\varepsilon^k, y^k)\}$. Assume that the constraint nondegeneracy (LICQ) holds at $\overline{X} := \Pi_{\mathcal{S}^n_+}(C + \mathcal{A}^*\bar{y})$.

Then the whole sequence $\{(\varepsilon^k, y^k)\}$ converges to $(\bar{\varepsilon}, \bar{y})$ quadratically, i.e.,

$$\left\|(\varepsilon^{k+1} - \bar{\varepsilon}, y^{k+1} - \bar{y})\right\| = O\left(\|(\varepsilon^k - \bar{\varepsilon}, y^k - \bar{y})\|^2\right).$$

Numerical results:

Consider

$$\begin{array}{ll} \min & \frac{1}{2}\|X - C\|^2 \\ \text{s.t.} & X_{ij} = e_{ij}, \ (i,j) \in \mathcal{B}_e \, , \\ & X_{ij} \geq l_{ij}, \ (i,j) \in \mathcal{B}_l \, , \\ & X_{ij} \leq u_{ij}, \ (i,j) \in \mathcal{B}_u \, , \\ & X \in \mathcal{S}^n_+ \, . \end{array}$$

Here $\mathcal{B}_e$, $\mathcal{B}_l$, and $\mathcal{B}_u$ are three index subsets of $\{(i,j) \,|\, 1 \leq i \leq j \leq n\}$ satisfying $\mathcal{B}_e \cap \mathcal{B}_l = \emptyset$, $\mathcal{B}_e \cap \mathcal{B}_u = \emptyset$, and $l_{ij} < u_{ij}$ for any $(i,j) \in \mathcal{B}_l \cap \mathcal{B}_u$. Denote the cardinalities of $\mathcal{B}_e$, $\mathcal{B}_l$, and $\mathcal{B}_u$ by p, q_l, and q_u, respectively. Let $m := p + q_l + q_u$. For any $(i,j) \in \{1, \ldots, n\} \times \{1, \ldots, n\}$, define $\mathcal{E}^{ij} \in \Re^{n \times n}$ by

$$(\mathcal{E}^{ij})_{st} := \begin{cases} 1 & \text{if } (s,t) = (i,j)\,, \\ 0 & \text{otherwise}\,, \end{cases} \qquad s,t = 1, \ldots, n\,.$$

In numerical implementation, we first obtain $\Delta\varepsilon^k = -\varepsilon^k + \zeta_k \hat{\varepsilon}$, and then apply the BiCGStab iterative solver of Van der Vorst to the resulted linear system

$$G'_y(\varepsilon^k, y^k)\Delta y^k = -G(\varepsilon^k, y^k) - G'_\varepsilon(\varepsilon^k, y^k)\Delta\varepsilon^k$$

to obtain Δy^k such that it satisfies

$$\|R_k\| \leq \min\{\eta_k \|G(\varepsilon^k, y^k) + G'_\varepsilon(\varepsilon^k, y^k)\Delta\varepsilon^k\|, \, \eta\|E(\varepsilon^k, y^k)\|\} \, ,$$

We have

$$
\begin{aligned}
& G'_y(\varepsilon, y)\Delta y \\
&= \Upsilon'_y(\varepsilon, y)\Delta y + \kappa\varepsilon\Delta y \\
&= \Delta y - \psi'_z(\varepsilon, z)(\Delta y - \mathcal{A}\Phi'_X(\varepsilon, X)(\mathcal{A}^*\Delta y)) + \kappa\varepsilon\Delta y \\
&= \Delta y - \psi'_z(\varepsilon, z)\Delta y + \psi'_z(\varepsilon, z)(\mathcal{A}\Phi'_X(\varepsilon, X)(\mathcal{A}^*\Delta y)) + \kappa\varepsilon\Delta y,
\end{aligned}
$$

where $z := y - (\mathcal{A}\Phi(\varepsilon, X) - b)$ and $X := C + \mathcal{A}^* y$. Then,

$$
\Phi'_X(\varepsilon, X)(\mathcal{A}^*\Delta y) = P[\Omega(\varepsilon, \lambda) \circ (P^T\mathcal{A}^*\Delta y P)]P^T.
$$

In order to compute the coefficient matrix $G'_y(\varepsilon, y)$ one needs $O(m^2n^3)$ flops. This implies that it is impractical to use direct methods to solve the Newton linear system even when n and m are not large, say $n = 100$ and $m = 1,000$.

Given the fact that the coefficient matrix $G'_y(\varepsilon, y)$ is nonsymmetric when the LSCM problem has inequality constraints, i.e., $q \neq 0$, it is natural to choose the BiCGStab as our iterative solver for solving the Newton linear system.

A diagonal preconditioner:

Define the vector $d \in \Re^m$ as the diagonal part of the coefficient matrix $G'_y(\varepsilon, y)$. Then for each $l \in \{1, \ldots, m\}$ that

$$d_l = [G'_y(\varepsilon, y)]_{ll} = [G'_y(\varepsilon, y)I_l]_l$$
$$= 1 - \phi'_{z_l}(\varepsilon, z_l) + \phi'_{z_l}(\varepsilon, z_l)w_l + \kappa\varepsilon,$$

where $I_l \in \Re^m$ denotes the lth column of the identity matrix I

And w_l is defined by

$$\begin{aligned} w_l \; &:= [\,\mathcal{A}\Phi'_X(\varepsilon, X)(\mathcal{A}^* I_l)\,]_l \\ &= \langle\, I_l \,,\, \mathcal{A}\Phi'_X(\varepsilon, X)(\mathcal{A}^* I_l)\,\rangle \\ &= \langle\, \mathcal{A}^* I_l \,,\, \Phi'_X(\varepsilon, X)(\mathcal{A}^* I_l)\,\rangle \\ &= \langle\, \mathcal{A}^* I_l \,,\, P[\Omega(\varepsilon, \lambda) \circ (P^T \mathcal{A}^* I_l P)]P^T \rangle \\ &= \langle\, P^T A_l P \,,\, \Omega(\varepsilon, \lambda) \circ (P^T A_l P)\rangle. \end{aligned}$$

To compute the diagonal matrix $D := \mathrm{diag}(d)$ needs $O(mn^3)$ flops.

The cost for computing D can be reduced if most of the matrices A_l's are sparse.

For instance, for each $l \in \{1, \ldots, m\}$, let the matrix A_l take the form

$$A_l = \frac{1}{2}(\mathcal{E}^{i_l j_l} + \mathcal{E}^{j_l i_l}),$$

where

$$(i_l, j_l) \in \begin{cases} \mathcal{B}_e & \text{if } l = 1, \ldots, p, \\ \mathcal{B}_l & \text{if } l = p+1, \ldots, p+q_l, \\ \mathcal{B}_u & \text{if } l = p+q_l+1, \ldots, m. \end{cases}$$

By taking account of such a special structure of A_l, w_l can be further simplified as follows

$$\begin{aligned} w_l &= \langle\, P^T A_l P\,,\, \Omega(\varepsilon,\lambda)\circ (P^T A_l P)\rangle \\ &= \frac{1}{2}[\, a_{i_l}^2 \Omega(\varepsilon,\lambda)(a_{j_l}^2)^T + (a_{i_l}\circ a_{j_l})\Omega(\varepsilon,\lambda)(a_{i_l}\circ a_{j_l})^T], \end{aligned}$$

where $l = 1,\ldots,m$ and "$\circ$" denotes the Hadamard product of two vectors, a_i is the ith row of P, and $a_i^2 := a_i \circ a_i$, $i = 1,\ldots,n$.

Thus, in this case the diagonal matrix D can be computed with a reduced cost of $O(mn^2)$ flops.

However, when m is much larger than n, say $m = O(n^2)$, this cost is still too expensive.

In our numerical implementation, we use an estimated diagonal matrix of D as our diagonal preconditioner with cost of $O(n^3)$ flops.

Let $\tilde{w} \in \Re^m$ and $\tilde{d} \in \Re^m$ be defined by

$$\tilde{w}_l := a^2_{i_l} \Omega(\varepsilon, \lambda)(a^2_{j_l})^T \,, \quad l = 1, \ldots, m,$$

and

$$\tilde{d}_l := 1 - \phi'_{z_l}(\varepsilon, z_l) + \phi'_{z_l}(\varepsilon, z_l)\tilde{w}_l + \kappa\varepsilon \,, \quad l = 1, \ldots, m \,,$$

respectively. Then it holds

$$0 < \frac{d_l}{\tilde{d}_l} \leq 1 \,, \quad l = 1, \ldots, m \,.$$

Since

$$(a_{i_l} \circ a_{j_l})\Omega(\varepsilon, \lambda)(a_{i_l} \circ a_{j_l})^T$$

is usually very small compared with

$$a_{i_l}^2 \Omega(\varepsilon, \lambda)(a_{j_l}^2)^T,$$

we have $\frac{d_l}{\tilde{d}_l} \approx 1$.

Let $\widetilde{D} := \mathrm{diag}(\tilde{d})$. Computing the diagonal matrix $\widetilde{D}$ only requires $O(n^3)$ flops, which is independent of m. We use the diagonal matrix $\widetilde{D}$ instead of D as our diagonal preconditioner.

Example 0.1 *The matrix C is the 387×387 1-day correlation matrix (as of June 15, 2006) from the lagged datasets of RiskMetrics* (`www.riskmetrics.com/stddownload_edu.html`). *For the test purpose, we perturb C to*

$$C := (1-\alpha)C + \alpha R,$$

where $\alpha 0.1$ and R is a randomly generated symmetric matrix with entries in $[-1, 1]$. The index sets

$$\mathcal{B}_e := \{(i,i) \,|\, i = 1, \ldots, 387\}, \;\; \mathcal{B}_l \cup \mathcal{B}_u = \emptyset$$

and $e_{ii} = 1$ for $(i,i) \in \mathcal{B}_e$.

Example 0.2 *All the data are the same as in Example 0.1 except that* $e_{ii} \in [0, 1]$ for $(i, i) \in \mathcal{B}_e$ *are randomly generated. This example corresponds to the* W*-weighted nearest correlation problem when the weight matrix* W *is a randomly generated diagonal matrix.*

Example 0.3 *The matrix C is a randomly generated $n \times n$ symmetric matrix with entries in $[-1, 1]$. The index sets $\mathcal{B}_l$, $\mathcal{B}_u \subset \{(i,j) \,|\, 1 \le i < j \le n\}$ consist of the indices of $\min(\hat{n}_r, n-i)$ randomly generated elements at the ith row of X, $i = 1, \ldots, n$ with $\hat{n}_r$ taking the following values: a) $\hat{n}_r = 5$; and $\hat{n}_r = 10$. We take $l_{ij} = -0.1$ for $(i,j) \in \mathcal{B}_l$ and $u_{ij} = 0.1$ for $(i,j) \in \mathcal{B}_u$.*

We consider the following two cases: a) $e_{ii} = 1$, $(i,i) \in \mathcal{B}_e$ and b) $e_{ii} = \alpha + (1-\alpha)\omega$, $(i,i) \in \mathcal{B}_e$, where $\alpha = 0.1$ and ω is a randomly generated number in $[0, 1]$.

	Example 0.1			**Example 0.2**		
Method	*Iter*	*cputime*	*Res*	*Iter*	*cputime*	*Res*
`GM`	37	0:44	1.0e-5	5000*	1:23:00	3.3e-2
`BFGS`	16	0:33	7.4e-6	1203	34:35	9.8e-6
`Semismooth`	5	0:09	1.6e-7	12	0:21	4.0e-8
`IP-NCM`	11	1:18	6.8e-9	18	3:07	2.6e-8
`Smoothing`	5	0:10	3.2e-7	12	0:23	1.0e-7

Table 1: Numerical results for Examples 0.1 and 0.2

Example 0.3			**n=2000**		
Method	*Case*	$\vert\hat{n}_r$	*Iter*	*cputime*	*Res*
`PGM`	a)	5	213	6:23:57	9.5e-6
		10	229	7:02:37	9.7e-6
	b)	5	372	>10 `hrs`	4.0e-2
		10	371	>10 `hrs`	1.5e-2
`BFGS-SQP`	a)	5		−−	
		10		−−	
	b)	5		−−	
		10		−−	
`Smoothing`	a)	5	9	29:33	1.5e-8
		10	9	30:37	7.3e-7
	b)	5	10	29:49	4.3e-8
		10	13	46:39	1.5e-8