Iterative solver for linear systems arising in interior point methods for semidefinite programming

Jacek Gondzio

Email: J.Gondzio@ed.ac.uk
URL: http://www.maths.ed.ac.uk/~gondzio/

joint work with

Stefania Bellavia and Margherita Porcelli

Objective: Improve IPMs for SDP

- Remove memory bottleneck
- Accelerate (if possible)

Redesign IPMs for SDP:

- Replace *exact* Newton Method with *inexact* Newton Method
- Work in *matrix-free* and *limited-memory* regime
- Preconditioning

Applications

- Max-Cut
- Matrix completion

Such techniques work in LP/QP/Least Squares:

Dembo, Eisenstat and Steihaug,
Inexact Newton Methods, *SINUM* 19 (1982) 400–408.

Bellavia, Inexact IPM, *JOTA* 96 (1998) 109–121.

Gondzio, Convergence analysis of an inexact feasible IPM for convex QP, *SIOPT* 23 (2013) 1510–1527.

Gondzio, Matrix-free IPM, *COAP*, 51 (2012) 457–480.

Bellavia, Gondzio and Morini,
A matrix-free preconditioner for sparse symmetric positive definite systems and least-squares problems, *SISC* 35 (2013) A192–A211.

SDP in standard form

- Primal form

$$\begin{array}{ll} \min & C \bullet X \\ \text{s.t.} & X \succeq 0 \\ & A_i \bullet X = b_i \quad i = 1, \ldots, m, \end{array}$$

where $A_i \in S\mathbb{R}^{n\times n}, C \in S\mathbb{R}^{n\times n}$, $b \in \mathbb{R}^m$ and $X \in S\mathbb{R}^{n\times n}$.

- Dual form

$$\begin{array}{ll} \max & b^T y \\ \text{s.t.} & S \succeq 0 \\ & S = C - \sum_{i=1}^m y_i A_i, \end{array}$$

where $y \in \mathbb{R}^m$ and $S \in S\mathbb{R}^{n\times n}$.

The operation $A \bullet B = trace(A^T B)$.

The "sparse" SDP problem

- Special interest in S sparse.
  S is the linear combination: $S = C - \sum_{i=1}^{m} y_i A_i$
  hence its sparsity pattern is a union of those of C and A_i's.
  Vanderberghe, Andersen [FnTO, 2015].

- Applications:
  in semidefinite relaxations of the graph-partitioning problem (e.g. max-cut problem), eigenvalue optimization problems associated with graphs, box-constrained quadratic optimization problem, matrix-completion.

- Inspired by the Dual Potential Reduction method
  by **Benson, Ye, Zhang** [SIOPT 2000, OMS 1999].

Dual Path-Following Interior-Point Algorithm

- Dual barrier problem parametrized by $\mu > 0$

$$\begin{array}{ll} \max & b^T y - \mu \ln(det(S)), \\ \text{s.t.} & \sum_{i=1}^m y_i A_i + S = C \end{array}$$

- Let $X = \mu S^{-1} \succeq 0$, then the first-order optimality conditions for this problem are given by:

$$F_\mu(X, y, S) = \begin{pmatrix} \sum_{i=1}^m y_i A_i + S - C \\ A_i \bullet X - b_i \quad i = 1, \ldots, m \\ X - \mu S^{-1} \end{pmatrix} = 0.$$

Primal-dual complementarity condition: $XS = \mu I$

Dual Path-Following IPM (cont'd)

Choose a dual strictly feasible pair (S, y) and a scalar $\mu > 0$.

Outer **Interior-Point** iterations:
Update (reduce) $\mu := \sigma\mu$ until it is sufficiently small.

Inner **Newton** iterations:
Perform (damped) steps in Newton direction $(\Delta X, \Delta S, y)$ for the problem

$$F_\mu(X, y, S) = 0$$

until the following proximity criteria is satisfied:

$$\|S^{-1/2}\Delta S\, S^{-1/2}\|_F \leq \tau < 1$$

(maintaining S positive definite).

Todd, Acta Numerica, 2001,
Nesterov and Nemirovski, SIAM Publications, 1994.

The Newton Step

Let $A^T := [vec(A_1), vec(A_2), \ldots, vec(A_m)] \in \mathbb{R}^{n^2 \times m}$,
$\Delta x = vec(\Delta X)$, $\Delta s = vec(\Delta S)$.

- The Newton equation:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ I & 0 & \mu(S^{-1} \otimes S^{-1}) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = - \begin{bmatrix} 0 \\ 0 \\ vec(X - \mu S^{-1}) \end{bmatrix}$$

- The reduced form:

$$\underbrace{\boldsymbol{A(S^{-1} \otimes S^{-1})A^T}}_{\boldsymbol{M}} \Delta y = \frac{1}{\mu} A \left(vec(X) - \mu\, vec(S^{-1})\right);$$

$$\Delta s = -A^T \Delta y.$$

$M \in \mathbb{R}^{m \times m}$ is spd (Schur complement).

$\otimes$ denotes the Kronecker product

The Schur complement

$$\underbrace{A(S^{-1} \otimes S^{-1})A^T}_{M} \Delta y = \frac{1}{\mu} A\,(vec(X) - \mu\, vec(S^{-1}))$$

- The matrix $M \in \mathbb{R}^{m \times m}$ is generally dense.
- If we knew a primal feasible point (i.e. $A\,vec(X) = b$) and solved every linear system exactly

$$M \Delta y = \frac{1}{\mu} b - A\,vec(S^{-1})$$

then we could maintain the primal feasibility.

We will not do that!

New interior-point method for sparse SDP

- Context: S is sparse and the Schur complement M is too large to be stored or it is too difficult to solve systems with M directly.

- Aims: define a **matrix-free** procedure that exploits sparsity of S and allows for inexact computation of the step.

- Exploit a nice property: avoid updating the primal variable X. If needed it can be computed at the end of the process

$$X = \mu S^{-1}(I + (\sum_{i=1}^{m} y_i A_i) S^{-1}).$$

Inexactness

- **Inexact Interior-Point**: Fix μ and solve $F_\mu(X, y, S) = 0$ only to a low accuracy, except for the last IP iteration.
- **Inexact Newton**: use a CG-like method to approximately solve the Schur complement system, use $\eta \in (0, 1)$:

$$M\Delta y = \frac{1}{\mu}b - A\,vec(S^{-1}) + \boldsymbol{r}, \quad \|\boldsymbol{r}\| \leq \eta \left\| \frac{1}{\mu}b - A\,vec(S^{-1}) \right\|.$$

A new iterate $X^+ = X + \Delta X$ violates the primal feasibility:

$$A\,vec(X + \Delta X) = b + \mu\boldsymbol{r},$$

but the feasibility can be restored at the end of the inner Newton iteration. Eliminate X from the computation of the next rhs:

$$\frac{1}{\mu}(\underbrace{A\,vec(X^+) - \mu r}_{b}) - \mu\,vec((S^+)^{-1}),$$

Matrix-free context

Assume S is sparse and a sparse Cholesky factorization $S = R^T R$ has been computed.

- The structure of S does not change hence reordering can be carried out once at the beginning of the process.
- $M = A(S^{-1} \otimes S^{-1})A^T$ is needed only to perform matrix-vector multiplications. Each column of M can be computed once at a time and then discarded.
- The computation of each column of M can be performed involving back-solves with R. The number of required back-solves depends on the structure of A_i matrices.

Matrix-vector products

Assume that the constraint matrices A_i have rank p.
(In max-cut problem: $p = 1$; in matrix-completion $p = 2$.)

The evaluation of a column of M needs p back-solves with S.

A matrix-vector product with M needs mp back-solves with S.

If $\delta(R)$ denotes the density of the Cholesky factor of S, then the computational effort of a matrix-vector product with M is

$$2mp \times O(m^2\, \delta(R)) \;\approx\; O(m^3\, p\, \delta(R)).$$

On the other hand, if M is stored, then the cost of a matrix-vector product drops to $O(m^2)$.

- We pay high price for saving memory.

However, back-solves with S can be performed in parallel.

The Limited Memory Preconditioner

Consider a partition of M

$$M = A(S^{-1} \otimes S^{-1})A^T = \begin{bmatrix} M_{11} & M_{21}^T \\ M_{21} & M_{22} \end{bmatrix},$$

where $M_{11} \in \mathcal{R}^{k\times k}$, $M_{21} \in \mathcal{R}^{(m-k)\times k}$, $M_{22} \in \mathcal{R}^{(m-k)\times(m-k)}$.

Compute the **partial Cholesky factorization**

$$M = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix} \begin{bmatrix} D_L & \\ & \boldsymbol{Z} \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ & I \end{bmatrix},$$

where

$$\boldsymbol{Z} = M_{22} - M_{21}M_{11}^{-1}M_{21}^T,$$

is the Schur complement of M_{11} in M.

Order diagonal elements of D_L and $D_Z = diag(Z)$:

$$\underbrace{d_1 \geq d_2 \geq \cdots \geq d_k}_{D_L} \geq \underbrace{d_{k+1} \geq d_{k+2} \geq \cdots \geq d_m}_{D_Z}.$$

The "Partial" Cholesky Preconditioner

Form only the first k columns of M: $\begin{bmatrix} M_{11} \\ M_{21} \end{bmatrix}$ and compute the partial Cholesky factorization

$$M = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix} \begin{bmatrix} D_L & \\ & \boldsymbol{Z} \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ & I \end{bmatrix}.$$

Do **not** compute $\boldsymbol{Z}$. Update only its diagonal.

Precondition M with

$$P = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix} \begin{bmatrix} D_L & \\ & \boldsymbol{D_Z} \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ & I \end{bmatrix},$$

where $\boldsymbol{D_Z}$ is a diagonal of $\boldsymbol{Z}$.

[Gondzio, COAP, 2012], [Bellavia, Gondzio, Morini, SISC, 2013].

Reordering of M

A "greedy" heuristic acts on the largest eigenvalues of M.

- Permute rows and columns of M so that M_{11} contains the k largest elements of $diag(M)$.
- This choice is motivated by the fact that if we set $D_Z = I$, then

$$\lambda_{max}(P^{-1}M) \leq k + tr(Z) \leq k + tr(M_{22})$$

hence it is expected that
$\lambda_{max}(P^{-1}M)$ is significantly smaller than $\lambda_{max}(M)$.

Spectral properties

- k eigenvalues of $P^{-1}M$ are equal to 1.
- The remaining ones are the eigenvalues of ${D_Z}^{-1}Z$.

Preconditioner: storage and computations

Memory requirements

- $nnz(L) \leq O(km)$
- simple choice of k (depends only on the available memory).

Computational cost

- Computing k columns of M requires kp back-solves with S.
- Computing the Cholesky factorization of the first k columns of M is negligible:

$$O(k^2 m) + O(k^3).$$

Two examples of SDP problems

- **SDP relaxation of Max-Cut Problem**
  Find a subset V of the vertex set in a graph such that the total weight of edges between V and its complement V^C is maximized.

- **SDP relaxation of Matrix Completion Problem**
  Recover a (low rank) data matrix $B \in \mathbb{R}^{\hat{m} \times \hat{n}}$ knowing a "small" number of its entries $B_{i,j}$ for $(i, j) \in \Omega$.

Preliminary numerical results

- $\mu_0 = 1$, $CG_{max} = 100$, $tol_{CG} = 10^{-3}$, $\sigma = 0.1$
- Inexact IP: we stop the inner iterations and decrease μ by a factor σ whenever a full Newton step is taken and

$$\|S^{-1/2} \Delta S \, S^{-1/2}\|_F \leq 0.1$$

- We stop the outer process when $\mu < 10^{-5}$. In the last IP outer iteration the Newton process is carried out until

$$\|S^{-1/2} \Delta S \, S^{-1/2}\|_F < 10^{-5}$$

- $k = 0.3\, m$ for the partial Cholesky
- Matlab (R2015a) code run on a Xeon 6-core E5645, 2.4 Ghz, 12 GB RAM.

SDP formulation of Max-Cut Problem

Find a subset V of the vertex set in a graph such that the total weight of edges between V and its complement V^C is maximized.

SDP relaxation (primal-dual pair):

$$\begin{array}{lllll} \max & C \bullet X & \qquad & \min & e^T y \\ \text{s.t.} & diag(X) = e & & \text{s.t.} & Diag(y) + S = C \\ & X \succeq 0 & & & S \succeq 0 \end{array}$$

- The sparsity of C depends on the sparsity of the adjacency matrix of the graph.
- $A_i = e_i e_i^T$, $i = 1, \ldots, m$, where e_i is the vector with 1 for the ith component and 0 for all others (rank 1).
- $S = C - Diag(y)$ possesses the same sparsity structure as C (constant).

Max-Cut: thoroidal graphs

Find a subset V of the vertex set in a graph such that the total weight of edges between V and its complement V^C is maximized.

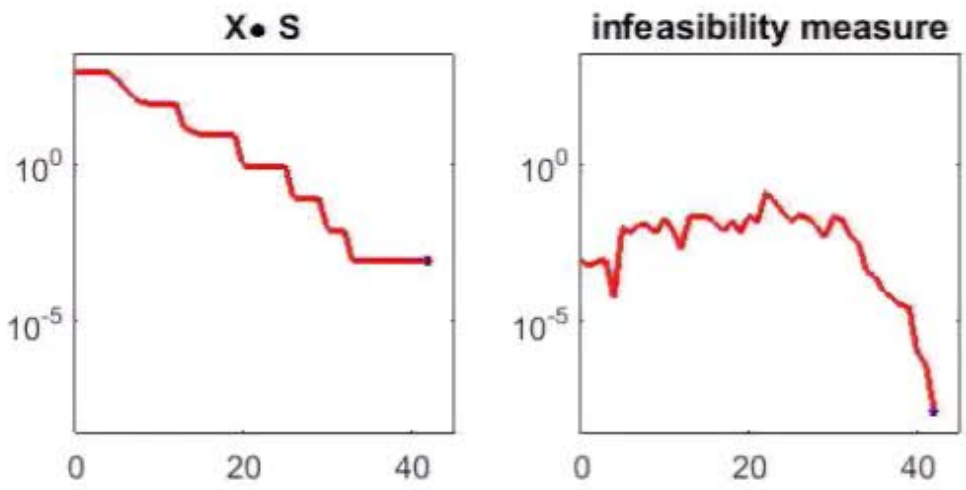
All m matrices $A_i, i = 1, 2, \ldots, m$ have rank 1.

Test name	m	$\delta(S)$	$\delta(R)$	IT_NEW	CG_AV	TIME_AV
G48	3000	1.7e-3	1.7e-2	18	8.5	2.5e1
G57	5000	1.0e-3	9.0e-3	39	41	2.3e2
G62	7000	7.1e-4	7.1e-3	44	47	5.4e2
G65	8000	6.2e-4	7.1e-3	41	48	7.3e2
G67	10000	5.0e-4	6.3e-3	40	48	1.2e3

IT_NEW: Overall number of inner Newton iterations;
CG_AV: Average number of CG iterations for each Newton iteration;
TIME_AV: Average time is seconds of one inner Newton iteration.

G67: Average time in seconds to perform one inner Newton iteration when storing M: 111 sec.

G48: convergence history



We stop the process when $\mu \leq 1.e - 7$.

SDP reformulation of matrix completion prob

Recover a (low rank) data matrix $B \in \mathbb{R}^{\hat{m}\times\hat{n}}$ knowing a "small" number of its entries $B_{i,j}$ for $(i,j) \in \Omega$. ($|\Omega| = m \ll \hat{m}\hat{n}$).

- We generated $B \in \mathbb{R}^{\hat{n}\times\hat{n}}$ of rank p by sampling two $\hat{n} \times p$ factors B_L and B_R and setting $B = B_L B_R^T$.
- We sampled a subset of m entries uniformly at random with $m = 4p(2n - p)$.
- In the SDP reformulation all m matrices A_i have rank 2. Then each matrix-vector product with M requires $2m$ backsolves with S.
- The density of dual matrix S is given by m/n^2, i.e. the fraction of known entries of B.
- Dual feasible initial guess is available.

SDP formulation of Matrix Completion Prob

$$\begin{array}{ll}\min & \text{rank}(W) \\ \text{s.t.} & W_{ij} = B_{ij} \ (i,j) \in \Omega\end{array} \quad \underbrace{\Rightarrow}_{\text{SDP relaxation}} \quad \begin{array}{ll}\min & Tr(W_1) + Tr(W_2) \\ \text{s.t.} & \begin{bmatrix} W_1 & W \\ W^T & W_2 \end{bmatrix} \succeq 0 \\ & W_{ij} = B_{ij} \ (i,j) \in \Omega\end{array}$$

$W \in \mathbb{R}^{\hat{m}\times\hat{n}}$, $W_1 \in \mathbb{R}^{\hat{m}\times\hat{m}}$, $W_2 \in \mathbb{R}^{\hat{n}\times\hat{n}}$ are the unknowns, $B_{ij}, (i,j) \in \Omega$ are given and $|\Omega| = m \ll \hat{m}\hat{n}$,

- $C = \frac{1}{2} I_n$, $X = \begin{bmatrix} W_1 & W \\ W^T & W_2 \end{bmatrix} \in \mathbb{R}^{n\times n}$, with $n = (\hat{m} + \hat{n})$.
- $A_l = \frac{1}{2}\begin{bmatrix} 0 & \Theta^{ij} \\ (\Theta^{ij})^T & 0 \end{bmatrix}$, $l = 1, \ldots, m$, with for each $(i,j) \in \Omega$

  $\Theta^{ij} \in \mathbb{R}^{\hat{m}\times\hat{n}}$ is $(\Theta^{ij})_{st} = \begin{cases} 1 & \text{if } (s,t) = (i,j) \\ 0 & \text{else} \end{cases}$ $(rank(A_l) = 2)$.

[Candes, Recht, 2009]

SDP reformulation of matrix completion prob

$\hat{n}$	m	$\delta(S)$	$\delta(R)$	IT_NEW	CG_AV	TIME_AV	TIME_M_AV
50	784	1.6e-1	4.2e-1	32	39	2.4	0.2
100	2364	1.2e-1	3.7e-1	34	32	10.6	0.8

TIME_M_AV: Average time is seconds to perform one inner Newton iteration when M is stored.

- The matrix B is recovered as

$$\frac{\|B - \bar{B}\|_F}{\|B\|_F} \simeq 1.5e-5$$

where $\bar{B}$ is the returned approximation.

Conclusions and future work

Dual Path-Following Interior Point Method:

- suitable for applications where S is sparse;
- Inexact IP approach+Inexact-Newton approach: we loose primal feasibility, but it is recovered in the last outer iteration;
- Matrix-free implementation avoids storing dense matrix M;
- Partial Cholesky preconditioner works well.

Main computational cost: computation of matrix-vector products with M, especially in the last Newton iterations:

- exploit parallelism in the matrix-vector products with M;
- perform inexact matrix-vector products with M [Bouras, Fraysse', Giraud, 2000], [Simoncini, Szyld, SISC 2003].