Preconditioning Dense Linear Systems from Large-Scale Semidefinite Programming Problems

Chih-Jen Lin

Industrial and Operations Engineering
University of Michigan

Mathematics and Computer Science Division
Argonne National Lab.

Fifth Copper Mountain Conference on Iterative Methods, April, 98

Outline

- Semidefinite programming problems (SDP) and interior point methods
- Large dense linear systems from interior point methods
- Incomplete Cholesky factorization for dense matrices
- Numerical experiments
- Conclusions and future research

Semidefinite Programming (SDP)

Linear Programming

$$\min c^T x$$
$$Ax = b$$
$$x \geq 0$$

Semidefinite Programming(SDP):

$$\min C \bullet X$$
$$A_i \bullet X = b_i, i = 1, \ldots, m$$
$$X \succeq 0$$

Dual of SDP:

$$\max \ \sum_{i=1}^m b_i y_i$$
$$\sum_{i=1}^m A_i y_i + S = C$$
$$S \succeq 0$$

1. A_i, C : $n \times n$ symmetric
2. $A \bullet B = \text{trace}(A^T B)$
   trace : sum of diagonal elements
   $\text{trace}(A^T B) = \sum (A^T B)_{ii}$
3. $X \succeq 0$: symmetric positive semidefinite ($v^T X v \geq 0, \forall v$)

An Example of SDP

$$\min \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} \bullet \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}$$

$$\begin{bmatrix} 4 & 5 \\ 5 & 6 \end{bmatrix} \bullet \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} = 1, \begin{bmatrix} 7 & 8 \\ 8 & 9 \end{bmatrix} \bullet \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} = 2$$

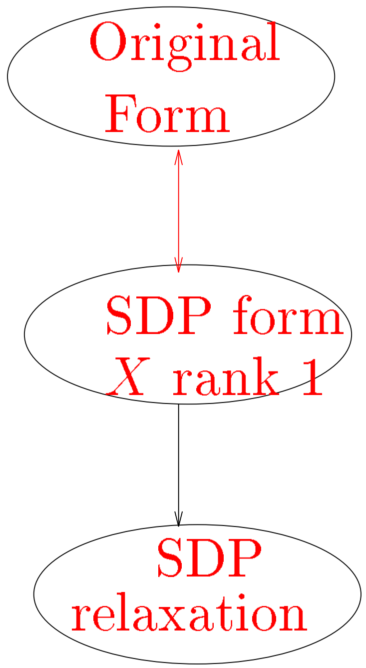
$$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} \text{ positive semidefinite}$$

$\Rightarrow$

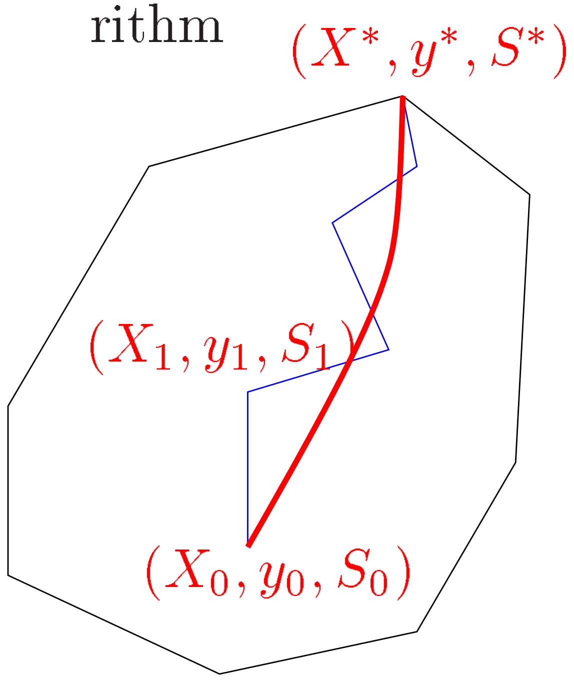
$$\begin{array}{ll} \min & x_{11} + 2x_{21} + 2x_{12} + 3x_{22} \\ \text{s.t} & 4x_{11} + 5x_{21} + 5x_{12} + 6x_{22} = 1 \\ & 7x_{11} + 8x_{21} + 8x_{12} + 9x_{22} = 2 \end{array} , \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} \text{ P.S.D.}$$

SDP and Interior Point Methods

- SDP relaxation to get lower bounds of NP-hard problems



- A polynomial time algorithm



Predictor and Corrector Directions

- Predictor Direction:

$$
\begin{aligned}
A_i \bullet \Delta X_k & & & = -(A_i \bullet X_k - b_i) \\
& \sum_{i=1}^{m} A_i \Delta y_i^k & + \quad \Delta S_k & = -(\sum_{i=1}^{m} A_i y_i^k + S_k - C) \\
\Delta X_k S_k & & + \quad X_k \Delta S_k & = t_k I - X_k S_k
\end{aligned}
$$

- Corrector direction:

$$
\begin{aligned}
A_i \bullet \Delta \bar{X}_k & & & = 0 \\
& \sum_{i=1}^{m} A_i \Delta \bar{y}_i^k & + \quad \Delta \bar{S}_k & = 0 \\
\Delta \bar{X}_k S_k & & + \quad X_k \Delta \bar{S}_k & = \bar{t}_k I - \bar{X}_k \bar{S}_k
\end{aligned}
$$

- The computational bottleneck. $\Delta X_k + \Delta \bar{X}_k$: the real direction.

Dense Linear Systems

There are different methods to solve both systems:

$$
\begin{aligned}
(\bar{A}(S_k^{-1} \otimes X_k)\bar{A}^T)\Delta y_k &= h_1 - \sum_{i=1}^{m} A_i \bullet ((h_3 - X_k h_2) S_k^{-1}) \\
\Delta S_k &= -\sum_{i=1}^{m} \Delta y_i^k A_i + h_2 \\
\Delta X_k &= (-h_3 - X_k \Delta S_k) S_k^{-1},
\end{aligned}
$$

h_1, h_2 and h_3 are the right-hand sides, $\otimes$ is the Kronecker product,
$\bar{A} = \begin{bmatrix} \mathrm{vec}(A_1)^T \\ \vdots \\ \mathrm{vec}(A_m)^T \end{bmatrix}$, $(\bar{A}(S_k^{-1} \otimes X_k)\bar{A}^T)_{i,j} = A_i \bullet (X_k A_j S_k^{-1})$ an m by m dense positive definite matrix

Preconditioned conjugate gradient method

- $m = 30,000$, 3.6GB for direct methods. $O(m^3)$ operations. A sequence of dense systems. Different from linear programming.
- $(\bar{A}F^{-1}G\bar{A}^T)p : m^2$, $\bar{A}(F^{-1}(G(\bar{A}^T p))) : 2m^2$ if $\text{nnz}(A_i) = O(1)$
- To store $A_i, X, S : O(n^2)$
- An example of CG iterations, $m = 8000$: (predictor step)

$$3, 4, 4, 5, 31, 12, 132, 133, 69, 29, 188, 219, 97, 130$$

- Diagonal preconditioner, more #CG in final iterations
- CG stopping criteria :

$$\frac{\|Ax - b\|_2}{1 + \|b\|_2} \leq 10^{-3}(\text{predictor}), 5 \times 10^{-3}(\text{corrector})$$

Dual feasibility

- Infeasible start algorithm: $0 < \alpha < 1$

$$A_i \bullet X_{k+1} - b = \alpha(A_i \bullet X_k - b_i), i = 1, \ldots, m,$$
$$A^T y^{k+1} + S_{k+1} - C = \alpha(A^T y^k + S_k - C),$$

$$\Delta S_k = -\sum_{i=1}^m \Delta y_i^k A_i + h_2, \quad \Delta X_k = (-h_3 - X_k \Delta S_k) S_k^{-1}.$$

- Primal-dual methods: duality gap $= 0 \Rightarrow$ Opt.

  Dual feasibility is good. Dual obj. value can be a lower bound

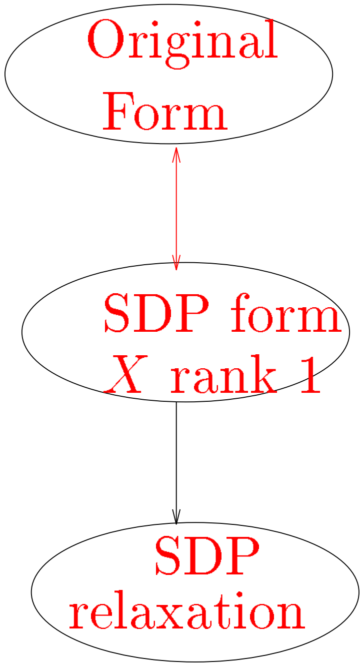
$$\frac{\|\bar{A}\text{vec}(X) - b\|_2}{1 + \|b\|_2} \approx 10^{-3}, \frac{\|\bar{A}^T y + \text{vec}(S) - \text{vec}(C)\|_2}{1 + \|\text{vec}(C)\|_2} < 10^{-15}$$

- Stopping criteria : relative dual value

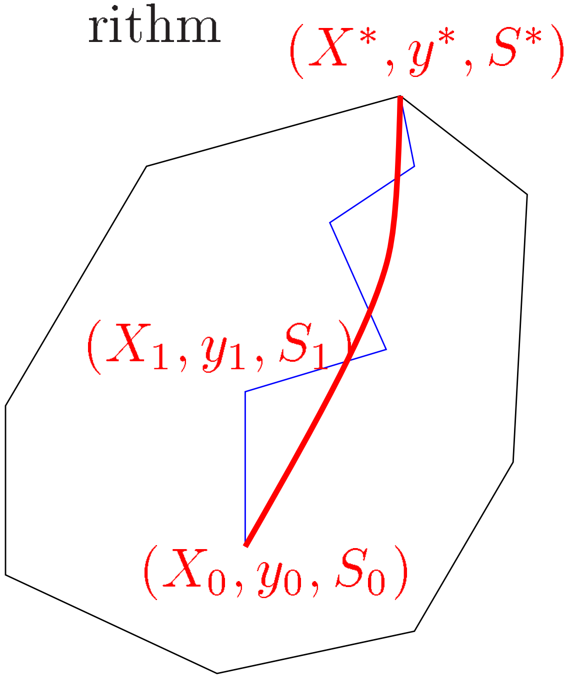
$$\frac{b^T y^{k+1} - b^T y^k}{1 + |b^T y^{k+1}|} \le 10^{-3}$$

SDP and Interior Point Methods

- SDP relaxation to get lower bounds of NP-hard problems



- A polynomial time algorithm



Preconditioning dense matrix

- Need a better but more expensive preconditioner
- Difficult to implement other preconditioner since we don't have $\bar{A}F^{-1}G\bar{A}^T$
- One preconditioner for predictor and corrector systems

Incomplete Cholesky Factorization (ICF)

Solve $Ax = b, LL^T$ as the preconditioner

$$A = LL^T + R$$

where

$$l_{i,j} = 0, \quad (i,j) \notin \mathcal{S}, \quad r_{i,j} = 0, \quad (i,j) \in \mathcal{S}.$$

Strategies

- Fixed fill : e.g. the same sparsity pattern of A
- Drop tolerance
- Variable fill

Questions

Is the sparsity pattern independent of numerical entries?

Are the storage requirements predictable?

ICF by selecting largest elements

- $A_1 = \begin{bmatrix} \alpha & v^T \\ v & B \end{bmatrix}$
- $A_2 = B - \frac{ww^T}{\alpha}$
- Practical implementation is different.
  L is obtained column by column
- Select largest elements of v to w
- Select largest elements in magnitude , Saad [1994], Jones and Plassmann [1995], Lin and Moré [1997] (sparse systems)
- m by m matrix, select $5m^{\frac{1}{3}}$ largest element
- L : m by m sparse matrix, #nnz $= O(m^{\frac{4}{3}}), (= O(n^2)$ for my application).
- Fix fill does not work
- Drop tolerance: memory unpredictable

Negative Pivots in ICF

- Two approaches : modify negative pivots to a positive number (local shift, $A + E$), or restart to factor $A + \alpha I$ (global shift), Manteuffel [1980]

- Local shift may generate a huge E. Running BCSSTK10 problem from Matrix Market by SLAP(Greenbaum and Seager), (P.D., cond $\approx 2 \times 10^8$, size $= 1000$)

```
33    -1410728.0881009
36    -44435433003.158
37    -3.6208110524319D+19
38    -1.4219949825513D+36
41    -3.2931075508642D+60
```

- Shift ICF, $D^{-1/2} A D^{-1/2} + 0.008I$ succeeds, is usually more robust

Scaling and shifting

- The actual procedure

  Choose $\mu = 1$.
  Compute $\hat{A} = D^{-1/2} A D^{-1/2}$, $D:$ scaling matrix,
  Set $\alpha_0 = 0$.
  For $k = 0, 1, \ldots,$
  Use ICF on $\hat{A}_k = \hat{A} + \alpha_k I$; exit if successful.
  Set $\alpha_{k+1} = \max(2\alpha_k, \mu)$

- Scaling matrices to improve the condition of $(\bar{A}(S_k^{-1} \otimes X_k)\bar{A}^T)$

- l_2 norm scaling vs. diagonal scaling

Large Quadratic Assignment Problems

- $\mathcal{P}$ plants assigned to $\mathcal{P}$ locations so as to minimize the flow-distance-cost. NP-hard. $\mathcal{P} \leq 20$ are not solved yet.
- Use ICF after #CG $\geq 0.1\mathcal{P}^2$

$\mathcal{P}$	m	diag.	ICF	diag.	ICF	
12	1730	905	626	818	539	66%
14	2746	739	648	652	561	86%
16	4098	935	863	848	776	92%
18	5834	948	712	861	625	73%
20	8002	1389	773	1282	666	52%
22	10650	4093	2256	3767	1930	51%
25	15627	940	895	702	657	94%

NUG problems

New Best Lower Bounds for QAP

Problem	Best lower bounds	New results
TAI25a	1006749	1010025
KRA30a	76003	78151

- Environment: SUN UltraSPARC1-140 with 128MB RAM
- QAP problems: from QAPLIB, Burkard, Karisch and Rendl[1996]
- SDP relaxations of QAP: Zhao, Karisch, Rendl and Wolkowicz [1996], Lin and Saigal[1997].

Conclusions and future research

- The new ICF is useful for solving dense systems from SDP relaxation of combinatorial problems.
- Negative pivots of ICF