Mixed sparse-dense linear least squares and preconditioned iterative methods

Miroslav Tůma

Faculty of Mathematics and Physics

Charles University

mirektuma@karlin.mff.cuni.cz

Joint work with **Jennifer Scott**, RAL and University of Reading

Based on a preprint submitted to SISC

More 2018, Roztoky,

August 3, 2018

Outline

Introduction: the Problem

$$\min_x \|Ax - b\|_2, \ A \in R^{m,n}, \ m \geq n$$

Introduction: the Problem

$$\min_x \|Ax - b\|_2, \ A \in R^{m,n}, \ m \geq n$$

- Small and dense full-rank problems
  - The solver choice often easier
  - Often points out to direct methods based on (complete) factorizations (Cholesky, QR etc. applied to A)

$$A^T Ax = A^T b \Rightarrow x = (A^T A)^{-1} A^T b, \ A = (Q_1 \ Q_2) \begin{pmatrix} R_1 \\ 0 \end{pmatrix} \Rightarrow x = R_1^{-1} Q_1^T b$$

Introduction: the Problem

$$\min_x \|Ax - b\|_2, \ A \in R^{m,n}, \ m \geq n$$

- Small and dense full-rank problems
  - The solver choice often easier
  - Often points out to direct methods based on (complete) factorizations (Cholesky, QR etc. applied to A)

$$A^T A x = A^T b \Rightarrow x = (A^T A)^{-1} A^T b, \ A = (Q_1 \, Q_2) \begin{pmatrix} R_1 \\ 0 \end{pmatrix} \Rightarrow x = R_1^{-1} Q_1^T b$$

- Large and sparse problems
  - There exist nice implementations of direct methods as LUSOL (Saunders, ver 7 - 2008), sparse QR factorization (SPQR in SuiteSparse)

Preconditioned iterative solvers: traps

1. The least squares problems are often much less structured than believed.
2. $\Rightarrow$ much harder to be solved by iterative approaches, much harder to find preconditioning
   - This makes a problem for both complete factorizations of direct methods and preconditioners.
   - But the latter suffer more.
   - Incomplete factorizations for $A^T A$ (the simplest idea) are often the ways to approximate factorization and get a preconditioner.

Introduction: the Problem

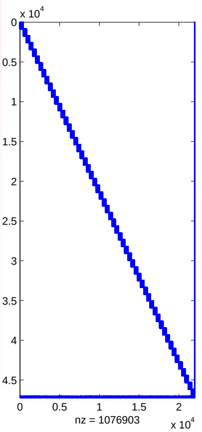
Preconditioned iterative solvers: traps

1. The least squares problems are often much less structured than believed.
2. $\Rightarrow$ much harder to be solved by iterative approaches, much harder to find preconditioning
   - This makes a problem for both complete factorizations of direct methods and preconditioners.
   - But the latter suffer more.
   - Incomplete factorizations for $A^T A$ (the simplest idea) are often the ways to approximate factorization and get a preconditioner.
3. What if a sparse problem has a few additional dense rows?

Introduction: the Problem

Example of a mixed sparse-dense matrix

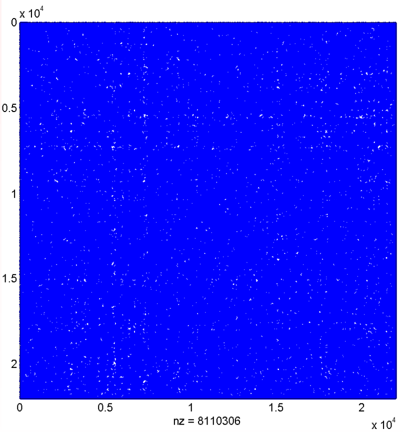
- Original matrix



Introduction: the Problem

Example of a mixed sparse-dense matrix

- Normal equations



Introduction: the Problem

$$\min_x \|Ax - b\|_2, \; A \in R^{m,n}, \; m \geq n$$

- Iterative approaches + approximations may add more flexibility than the direct methods (based on decompositions):

Introduction: the Problem

$$\min_x \|Ax - b\|_2, \ A \in R^{m,n}, \ m \geq n$$

- Iterative approaches + approximations may add more flexibility than the direct methods (based on decompositions):
- Hopefully also for treating the dense rows.

Introduction: the Problem

$$\min_x \|Ax - b\|_2, \; A \in R^{m,n}, \; m \geq n$$

- Iterative approaches + approximations may add more flexibility than the direct methods (based on decompositions):
- Hopefully also for treating the dense rows.
  - Stretching dense rows (splitting them into a bunch of “shorter” rows) ⇒ interesting bunch o problems related to the matrix conditioning, scaling, sparsity patterns

Introduction: the Problem

$$\min_x \|Ax - b\|_2,\ A \in R^{m,n},\ m \geq n$$

- Iterative approaches + approximations may add more flexibility than the direct methods (based on decompositions):
- Hopefully also for treating the dense rows.
  - Stretching dense rows (splitting them into a bunch of "shorter" rows) ⇒ interesting bunch o problems related to the matrix conditioning, scaling, sparsity patterns
  - Canonical decomposition of the problem (Dulmage-Mendelsohn) into a block triangular form ⇒ embedding dense rows into the blocks, structural versus "graph-bsed" rank-deficiency

Introduction: the Problem

$$\min_x \|Ax - b\|_2, \ A \in R^{m,n}, \ m \geq n$$

- Iterative approaches + approximations may add more flexibility than the direct methods (based on decompositions):
- Hopefully also for treating the dense rows.
  - Stretching dense rows (splitting them into a bunch of "shorter" rows) ⇒ interesting bunch o problems related to the matrix conditioning, scaling, sparsity patterns
  - Canonical decomposition of the problem (Dulmage-Mendelsohn) into a block triangular form ⇒ embedding dense rows into the blocks, structural versus "graph-bsed" rank-deficiency

treated here – simple overdetermined case, full column rank

Introduction: the Problem

Example of a mixed sparse-dense matrix

- Trouble caused by the dense rows can be observed from more different angles

Introduction: the Problem

Example of a mixed sparse-dense matrix

- Trouble caused by the dense rows can be observed from more different angles
  - The matrix to be factorized (completely/incompletely) is dense as we saw above

Introduction: the Problem

Example of a mixed sparse-dense matrix

- Trouble caused by the dense rows can be observed from more different angles
  - The matrix to be factorized (completely/incompletely) is dense as we saw above
  - The outer-product of the dense rows makes from the useful information a noise

Introduction: the Problem

Example of a mixed sparse-dense matrix

- Trouble caused by the dense rows can be observed from more different angles
  - The matrix to be factorized (completely/incompletely) is dense as we saw above
  - The outer-product of the dense rows makes from the useful information a noise
- Troublemakers to be treated in a special way may be not only dense rows.

Example of a mixed sparse-dense matrix

- Trouble caused by the dense rows can be observed from more different angles
  - The matrix to be factorized (completely/incompletely) is dense as we saw above
  - The outer-product of the dense rows makes from the useful information a noise
- Troublemakers to be treated in a special way may be not only dense rows.
- "Bad" columns as?

$$A = \begin{pmatrix} \tilde{A} & a \end{pmatrix}, \; A^T A = \begin{pmatrix} \tilde{A}^T \tilde{A} & \tilde{A}^T a \\ a^T \tilde{A} & a^t a \end{pmatrix}$$

Outline

Mixed sparse-dense least squares

Terminology: split the system

$$A = \begin{pmatrix} A_s \\ A_d \end{pmatrix},\ A_s \in R^{m_s \times n},\ A_d \in R^{m_d \times n},\ b = \begin{pmatrix} b_s \\ b_d \end{pmatrix},\ b_s \in R^{m_s},\ b_d \in R^{m_d}, \quad (1)$$

with $m = m_s + m_d$, $m_s \geq n$, and $m_d \geq 1$ (in general, $m_s \gg m_d$).

$$\min_x \left\| \begin{pmatrix} A_s \\ A_d \end{pmatrix} x - \begin{pmatrix} b_s \\ b_d \end{pmatrix} \right\|_2 . \quad (2)$$

Set $C = A^T A$, $C_s = A_s^T A_s$ (reduced normal matrix), $C_d = A_d^T A_d$

A lot of previous work on direct methods' approaches and related problems:

Björck, Duff, 1980; Heath, 1982; Björck, 1984; George, Ng, 1984-1987; Adlers, Björck, 2000; Avron, Ng, Toledo, 2009 and many others ...

Mixed sparse-dense least squares

Direct solution of the sparse-dense least squares

- Woodbury formula (Guttman, 1946; Woodbury, 1949, 1950): Dense rows plugged in a posteriori

$$C^{-1} = (C_s + C_d)^{-1} = C_s^{-1} - C_s^{-1} A_d^T (I_{m_d} + A_d C_s^{-1} A_d^T)^{-1} A_d C_s^{-1}.$$

The least squares solution:

$$x = x_s + C_s^{-1} A_d^T (I_{m_d} + A_d C_s^{-1} A_d^T)^{-1} (b_d - A_d x_s) \text{ with } x_s = (A_s A_s^T)^{-1} A_s^T b_s$$

Mixed sparse-dense least squares

Direct solution of the sparse-dense least squares

- Woodbury formula (Guttman, 1946; Woodbury, 1949, 1950): Dense rows plugged in a posteriori

$$C^{-1} = (C_s + C_d)^{-1} = C_s^{-1} - C_s^{-1} A_d^T (I_{m_d} + A_d C_s^{-1} A_d^T)^{-1} A_d C_s^{-1}.$$

  The least squares solution:

$$x = x_s + C_s^{-1} A_d^T (I_{m_d} + A_d C_s^{-1} A_d^T)^{-1} (b_d - A_d x_s) \ \text{with} \ x_s = (A_s A_s^T)^{-1} A_s^T b_s$$

- Sautter trick (Sautter, 1978; see Björck, 1996)

$$(C_s + C_d)^{-1} A_d^T = C_s^{-1} A_d^T (I_{m_d} + A_d C_s^{-1} A_d^T)^{-1}$$

Mixed sparse-dense least squares

Direct solution of the sparse-dense least squares

- Woodbury formula (Guttman, 1946; Woodbury, 1949, 1950): Dense rows plugged in a posteriori

$$C^{-1} = (C_s + C_d)^{-1} = C_s^{-1} - C_s^{-1} A_d^T (I_{m_d} + A_d C_s^{-1} A_d^T)^{-1} A_d C_s^{-1}.$$

The least squares solution:

$$x = x_s + C_s^{-1} A_d^T (I_{m_d} + A_d C_s^{-1} A_d^T)^{-1} (b_d - A_d x_s) \text{ with } x_s = (A_s A_s^T)^{-1} A_s^T b_s$$

- Sautter trick (Sautter, 1978; see Björck, 1996)

$$(C_s + C_d)^{-1} A_d^T = C_s^{-1} A_d^T (I_{m_d} + A_d C_s^{-1} A_d^T)^{-1}$$

- This can be used to express direct solution more efficiently.

Outline

Mixed sparse-dense least squares

Approximate solution of the sparse-dense least squares

- What if the inverses are only approximate, e.g., from an incomplete factorization?

Mixed sparse-dense least squares

Approximate solution of the sparse-dense least squares

- What if the inverses are only approximate, e.g., from an incomplete factorization?

Theorem

Assume that ξ is an approximate solution to $\min_{x_s} \|A_s x_s - b_s\|_2$ (or whatever). Define $r_s = b_s - A_s\xi$ and $r_d = b_d - A_d\xi$. Then the exact least squares solution of the whole split problem is equal to $x = \xi + \Gamma$, where

$$\Gamma = C_s^{-1} A_s^T r_s + C_s^{-1} A_d^T (I_{m_d} + A_d C_s^{-1} A_d^T)^{-1} (r_d - A_d C_s^{-1} A_s^T r_s) \quad (3)$$

Mixed sparse-dense least squares

Approximate solution of the sparse-dense least squares

- What if the inverses are only approximate, e.g., from an incomplete factorization?

Theorem

Assume that ξ *is an approximate solution to* $\min_{x_s} \|A_s x_s - b_s\|_2$ *(or whatever). Define* $r_s = b_s - A_s\xi$ *and* $r_d = b_d - A_d\xi$*. Then the exact least squares solution of the whole split problem is equal to* $x = \xi + \Gamma$*, where*

$$\Gamma = C_s^{-1} A_s^T r_s + C_s^{-1} A_d^T (I_{m_d} + A_d C_s^{-1} A_d^T)^{-1} (r_d - A_d C_s^{-1} A_s^T r_s) \quad (3)$$

- We do not need to solve the sparse least squares exactly
- The formulation deals with residuals that represent a basic quantity inside iterative methods

Mixed sparse-dense least squares

Approximate solution and scaling transformation

- Computed factor can be applied as a scaling transformation similarly as in the direct sparse-dense solvers

Mixed sparse-dense least squares

Approximate solution and scaling transformation

- Computed factor can be applied as a scaling transformation similarly as in the direct sparse-dense solvers
- Also, the transformation is a crucial practical step. See, again, in Björck, 1996

$$C_s = L_s L_s^T \tag{4}$$

Mixed sparse-dense least squares

Approximate solution and scaling transformation

- Computed factor can be applied as a scaling transformation similarly as in the direct sparse-dense solvers
- Also, the transformation is a crucial practical step. See, again, in Björck, 1996

$$C_s = L_s L_s^T \tag{4}$$

- Getting an equivalent problem

$$\min_z \left\| \begin{pmatrix} B_s \\ B_d \end{pmatrix} z - \begin{pmatrix} b_s \\ b_d \end{pmatrix} \right\|_2, \tag{5}$$

$$B_s = A_s L_s^{-T}, \;\; B_d = A_d L_s^{-T}, \;\; z = L_s^T x$$

Mixed sparse-dense least squares

Transformation + exact decomposition $C_s = L_s L_s^T$ leads to

Lemma

If $C_s = L_s L_s^T$ (exactly) the least squares solution of the transformed split problem can be written as $z = \xi_1 + \Gamma_1$, where ξ_1 is an approximate solution to the scaled problem $\min_z \|B_s z - b_s\|_2$ (or whatever), $\rho_s = b_s - B_s\xi_1$ and $\rho_d = b_d - B_d\xi_1$ and

$$\Gamma_1 = B_s^T \rho_s + B_d^T (I_{m_d} + B_d B_d^T)^{-1} (\rho_d - B_d B_s^T \rho_s). \qquad (6)$$

Solving the Least Squares

Choice of ξ

- Approximate solution for A_s overdetermined

$$\min_{\xi} \|B_s \xi - b_s\|_2$$

  - $$\xi_1 \approx (B_s^T B_s)^{-1} B_s^T b_s = L_s^{-1} A_s^T A_s L_s^{-T} L_s^{-1} A_s^T b_s = L_s^{-1} A_s^T b_s$$

Solving the Least Squares

Choice of ξ

- Approximate solution for A_s overdetermined

$$\min_{\xi} \|B_s\xi - b_s\|_2$$

  - $$\xi_1 \approx (B_s^T B_s)^{-1} B_s^T b_s = L_s^{-1} A_s^T A_s L_s^{-T} L_s^{-1} A_s^T b_s = L_s^{-1} A_s^T b_s$$

- If B_d represents a significant part of the problem and its effect dominates:

$$\min_{\xi} \|B_d\xi - b_d\|_2$$

  - $$\xi \approx B_d^{\dagger} b_d$$

Solving the Least Squares

Scaling + exact $C_s = L_s L_s^T$ + exact sparse subproblem

Lemma

If $C_s = L_s L_s^T$ (exactly), the least squares solution of problem (5) can be written as $z = \xi_1 + \Gamma_1$, where ξ_1 minimizes $\|B_s z - b_s\|_2$ (exactly), $\rho_s = b_s - B_s \xi_1$ and $\rho_d = b_d - B_d \xi_1$ and

$$\Gamma_1 = B_d^T (I_{m_d} + B_d B_d^T)^{-1} \rho_d. \tag{7}$$

Solving the Least Squares

Scaling + exact $C_s = L_s L_s^T$ + exact sparse subproblem

Lemma

If $C_s = L_s L_s^T$ (exactly), the least squares solution of problem (5) can be written as $z = \xi_1 + \Gamma_1$, where ξ_1 minimizes $\|B_s z - b_s\|_2$ (exactly), $\rho_s = b_s - B_s \xi_1$ and $\rho_d = b_d - B_d \xi_1$ and

$$\Gamma_1 = B_d^T (I_{m_d} + B_d B_d^T)^{-1} \rho_d. \tag{7}$$

Various ways to evaluate Γ_1

- Dense least squares minimum in norm
- Dense LQ factorization.

Outline

Solving the Least Squares

Algorithm

Preconditioned CGLS algorithm $(A_s,\ A_d,\ A_s^T A_s \approx \tilde{L}_s \tilde{L}_s^T;\ z = M^{-1}s))$

0. $r_s^{(0)} = b_s - A_s x^{(0)},\ r_d^{(0)} = b_d - A_d x^{(0)},\ w_s^{(0)} = A_s^T r_s^{(0)},\ w_d^{(0)} = A_d^T r_d^{(0)},$

$$z^{(0)} = M^{-1}(w_s^{(0)} + w_d^{(0)}),\ p^{(0)} = z^{(0)}$$

1. **for** $i = 1 : nmax$ **do**

2. $q_s^{(i-1)} = A_s p^{(i-1)},\ q_d^{(i-1)} = A_d p^{(i-1)}\ \ \alpha = \dfrac{(w_s^{(i-1)} + w_d^{(i-1)}, z^{(i-1)})}{(q_s^{(i-1)}, q_s^{(i-1)}) + (q_d^{(i-1)}, q_d^{(i-1)})}$

3. $x^{(i)} = x^{(i-1)} + \alpha p^{(i-1)},\ r_s^{(i)} = r_s^{(i-1)} - \alpha q_s^{(i-1)},\ r_d^{(i)} = r_d^{(i-1)} - \alpha q_d^{(i-1)}$

4. $z^{(i)} = M^{-1}(A_s^T r_s^{(i)} + A_d^T r_d^{(i)})\ \ \beta = \dfrac{(w_s^{(i)} + w_d^{(i)}, z^{(i)})}{(w_s^{(i-1)} + w_d^{(i-1)}, z^{(i-1)})}$

5. $p^{(i)} = z^{(i)} + \beta p^{(i-1)}$

6. **end do**

Note the difference between CGLS1, CGLS2

Solving the Least Squares

Algorithm

Preconditioning procedure$(r_s, r_d, w, C_s \approx \tilde{L}_s \tilde{L}_s^T, B_d = A_d^T \tilde{L}_s^{-T}$, *chosen mode (*$Cholesky$ *or LQ)) The Cholesky mode needs* $I_{m_d} + B_d B_d^T \approx \tilde{L}_d \tilde{L}_d^T$ */ the LQ mode needs* $\begin{pmatrix} B_d & I_{m_d} \end{pmatrix} \approx \begin{pmatrix} \tilde{L}_d & 0_{m_d} \end{pmatrix} \tilde{Q}_d^T$.

1. *Solve* $\tilde{L}_s \xi_1 = w$ *for* ξ_1
2. $\rho_d = r_d - B_d \xi_1$
3. **if** *mode* == *Cholesky* **then**
4. $\quad u = B_d^T (\tilde{L}_d \tilde{L}_d^T)^{-1} \rho_d$
5. **else if** *mode* == *LQ* **then**
6. $\quad \rho_s = r_s - A_s \tilde{L}_s^{-T} \xi_1$
7. $\quad u = \tilde{L}_s^{-1} A_s^T \rho_s + \tilde{Q}_d(1:n, 1:m_d) * \tilde{L}_d^{-1} * (\rho_d - B_d \tilde{L}_s^{-1} A_s^T \rho_s)$
8. **end if**
9. *Solve* $\tilde{L}_s^T z = (\xi_1 + u)$ *for* z

- Can avoid recomputing $A_s^T r_s$ inside the preconditioner

Outline

Solving the Least Squares

Experimental evaluation

- Stopping criterion

  C1: Stop if $\|r\|_2 < \delta_1$

  C2: Stop if

$$\frac{\|A^T r\|_2}{\|r\|_2} < \frac{\|A^T r_0\|_2}{\|r_0\|_2} * \delta_2,$$

  r residual, r_0 initial residual, $\delta_1 = 10^{-8}$ and $\delta_2 = 10^{-6}$.

- Intel(R) Core(TM) i5-4590 CPU running at 3.30 GHz, 12 GB of internal memory. Visual Fortran Intel(R) 64 XE compiler (version 14.0.3.202)

Solving the Least Squares

Experimental evaluation: II

- Most of the matrices from the University of Florida Sparse Matrix Collection
- A prescaled normalizing columns:
  - A replaced by by AD, where D is diagonal
  - $D_{ii}^2 = 1/\|Ae_i\|_2$
  - $\Rightarrow$ Entries of AD are all less than one in absolute value.
- A row of A to be dense if the number of entries in the row either exceeds 100 times the average number of entries in a row or is more than 4 times greater than the number of entries in any row in the sparse part A_s.
- Removing dense rows can leave A_s rank deficient: modifying A by removing any columns of A that correspond to null columns of A_s.

Solving the Least Squares

Problem of null columns after removal of the dense part

$$A = \begin{pmatrix} A_1 & A_2 \end{pmatrix} \equiv \begin{pmatrix} A_{s_1} & A_{s_2} \\ A_{d_1} & A_{d_2} \end{pmatrix}, \qquad (8)$$

- A_s has n_2 null columns with $n_2 \ll n$ (null A_{s2}).
- The solution can be expressed as a combination of partial solutions.

Theorem

Let $\xi \in R^{n_1}$ *and* $\Gamma \in R^{n_1 \times n_2}$ *be the solutions to* $\min_z \|A_1 z - b\|_2$ *and* $\min_W \|A_1 W - A_2\|_F$*, respectively. Then the solution* $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ *of the original problem split conformally is given by* $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \xi - \Gamma x_2 \\ x_2 \end{pmatrix}$ *with* $(A_2^T A_2 - A_2^T A_1 \Gamma) x_2 = A_2^T b - A_2^T A_1 \xi$.

Solving the Least Squares

Table: Statistics: (density= $nnz(C)/n^2$)

Identifier	m	n	$nnz(A)$	nnz(C)	$nnz(C)/n^2$
aircraft	7,517	3,754	20,267	1.4×10^6	0.200
lp_fit2p	13,525	3,000	50,284	4.5×10^6	1.000
scrs8-2r	27,691	14,364	58,439	6.2×10^6	0.143
sctap1-2b	33,858	15,390	99,454	2.6×10^6	0.050
scsd8-2r	60,550	8,650	190,210	2.0×10^6	0.100
scagr7-2r	62,423	35,213	123,239	2.2×10^7	0.036
sc205-2r	62,423	35,213	123,239	6.5×10^6	0.010
sctap1-2r	63,426	28,830	186,366	9.1×10^6	0.050
scfxm1-2r	65,943	37,980	221,388	8.3×10^5	0.014
world	67,147	34,506	198,883	3.1×10^5	0.001
neos1	133,743	131,581	599,590	1.7×10^8	0.027
neos2	134,128	132,568	685,087	2.3×10^8	0.033
stormg2-125	172,431	66,185	433,256	1.0×10^6	0.002
PDE1	270,595	271,792	990,587	1.6×10^{10}	0.670
neos	515,905	479,119	1,526,794	5.3×10^8	0.034
stormg2_1000	1,377,306	528,185	3,459,881	4.2×10^7	0.002
cont1_l	1,921,596	1,918,399	7,031,999	8.2×10^{11}	0.667

Solving the Least Squares

Identifier	Dense rows not exploited				Dense rows exploited				
	size_p	*T_p*	*Its*	*T_i*	m_d	*size_ps*	*T_p*	*Its*	*T_i*
aircraft	22,509	0.09	44	0.02	17	3,750	0.01	1	0.01
lp_fit2p	17,985	0.26	‡	‡	25	4,940	0.09	1	0.01
scrs8-2r	86,169	0.94	380	0.50	22	36,385	0.01	1	0.02
sctap1-2b	92,325	0.39	639	0.69	34	68,644	0.01	1	0.01
scsd8-2r	51,885	0.25	90	0.11	50	51,855	0.05	7	0.02
scagr7-2r	197,067	3,34	244	0.53	7	152,977	0.06	1	0.01
sc205-2r	211,257	1.56	72	0.19	8	104,022	0.08	1	0.01
sctap1-2r	172,965	1.47	673	1.90	34	127,712	0.03	1	0.01
scfxm1-2r	227,835	0.59	187	0.51	58	227,823	0.14	33	0.23
neos1	789,471	†	†	†	74	789,471	5.27	132	3.71
neos2	†	†	†	†	90	795,323	5.46	157	4.84
stormg2-125	395,595	0.27	‡	‡	121	7,978,135	0.22	16	0.29
PDE1	†	†	†	†	1	1,623,531	12.7	696	1.28
neos	†	†	†	†	20	2,874,699	4.93	232	15.0
stormg2_1000	3,157,095	19.1	‡	‡	121	3,125,987	19.1	18	2.92
cont1_l	†	†	†	†	1	11,510,370	4.82	1	0.33

SCSD8-2r_a: size of C_s

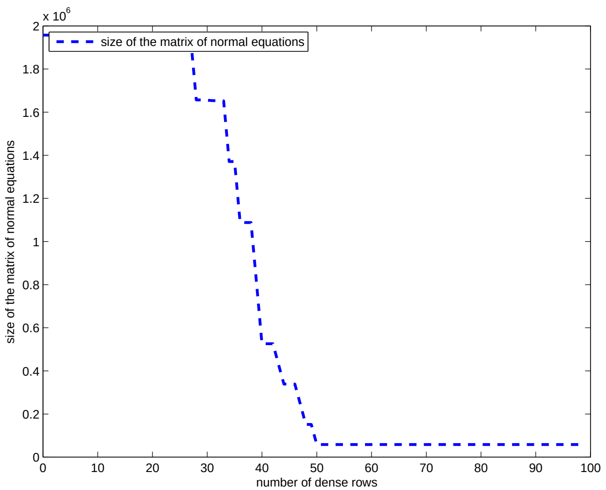
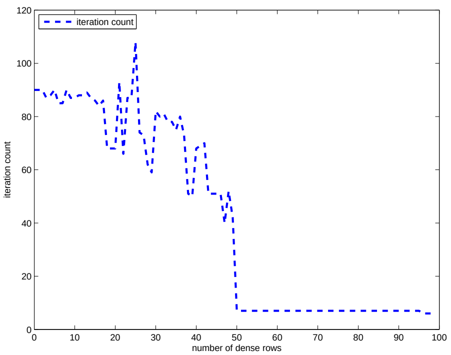


Figure: $|C_s|$.

Solving the Least Squares

SCSD8-2r_a: iteration counts + $size\_p/size(A^TA)$



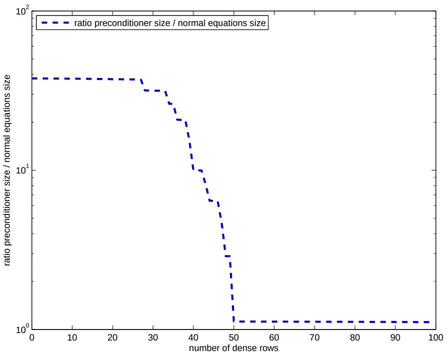


Figure: Problem $Meszaros/scsd8-2r$. Iteration counts (left), and ratio of the preconditioner size to the size of A^TA (right) as the number of dense rows that are removed from A is increased.

Solving the Least Squares

SCSD8-2r_a: timings

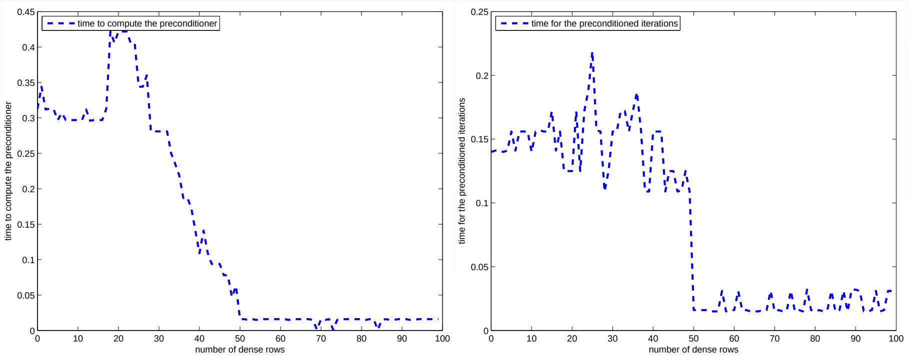


Figure: Problem $Meszaros/scsd8-2r$. Time to compute the preconditioner (left) and time for CGLS (right) as the number of dense rows that are removed from A is increased.

Solving the Least Squares

stormg2_1000: size of C_s

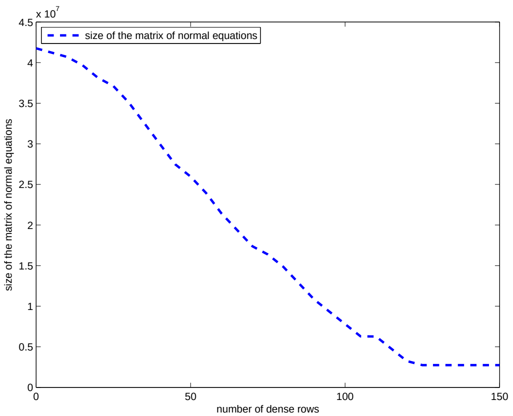


Figure: Problem $Mittelmann/stormg2\_1000$. Size of $A_s^T A_s$.

Solving the Least Squares

stormg2_1000: large problem: iteration counts + $size\_p/size(A^TA)$

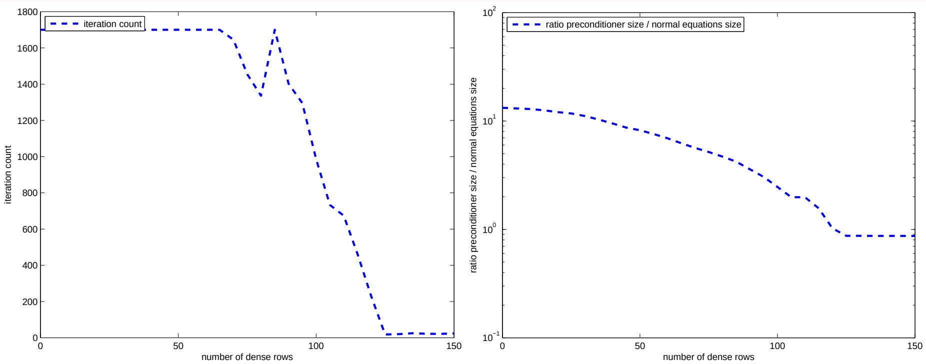


Figure: Problem $Mittelmann/stormg2\_1000$. Iteration counts (left), Ratio of the preconditioner size to the size of A^TA (right) as the number of dense rows that are removed from A is increase.

Solving the Least Squares

stormg2_1000: large problem: timings

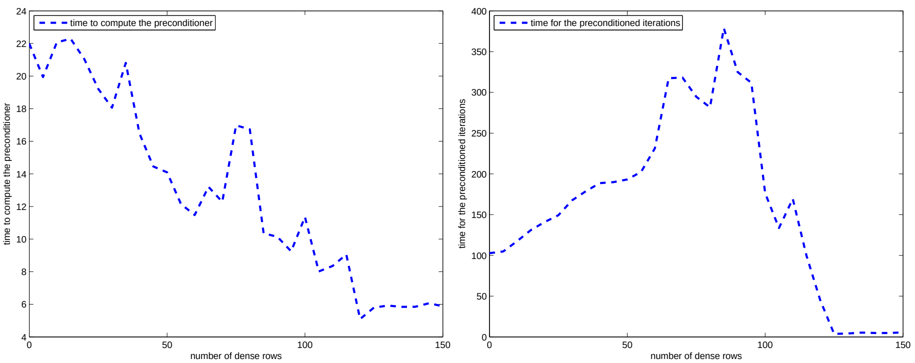


Figure: Problem $Mittelmann/stormg2\_1000$. Time to compute the preconditioner (left), time for the preconditioned iterations (right).

Solving the Least Squares

Conclusions

- Solving linear least squares problems where A has a number of dense rows.

Solving the Least Squares

Conclusions

- Solving linear least squares problems where A has a number of dense rows.
- A new approach that processes the dense rows separately within a conjugate gradient method

Conclusions

- Solving linear least squares problems where A has a number of dense rows.
- A new approach that processes the dense rows separately within a conjugate gradient method
- Not all the formulas above work in practice !!!! In our case, the best has been the simplest one.

Conclusions

- Solving linear least squares problems where A has a number of dense rows.
- A new approach that processes the dense rows separately within a conjugate gradient method
- Not all the formulas above work in practice !!!! In our case, the best has been the simplest one.
  - The dense rows must be treated separately
  - The dense rows must be considered (Avron, Ng and Toledo use an approach that takes them out from the consideration within a QR-based scheme - we faced significant troubles in our PCGLS based on Cholesky following this approach)

Last but not least

Thank you for your attention!