Computational Issues for a Low Rank Correction Preconditioner

Trond Steihaug
Department of Informatics, University of Bergen, Norway

Joint work with
Venansius Baryamureeba
Director, Institute of Computer Science, Makerere University, Uganda

Optimization 2001
July 23–25, 2001
Aveiro, Portugal

Outline

- Sequence of weighted linear least squares (WLLS) problems
- Four examples leading to a sequence of WLLS problems
- A preconditioner
- Computational issues
- Concluding Remarks

Weighted linear least squares problems

$$\min_x \|Ax - b\|_G$$

- $A \in \mathbb{R}^{m\times n}$, A has full rank, $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$.
- $G \in \mathbb{R}^{n\times n}$, diagonal and positive definite.
- $\|z\|_G^2 = z^T G z$, $z \in \mathbb{R}^n$.

Normal Equations ($h = A^T G b$):

$$A^T G A x = h.$$

Want to solve a *sequence* of problems of this type where
A is **constant** and the weight G and right hand side h **change**.

Linear Regression Model

$$y = Ax + \varepsilon,$$

- $A \in \mathbb{R}^{m\times n}$ is data or design matrix
  - $m > n$, $\mathrm{rank}(A) = n$
- $y \in \mathbb{R}^m$ is vector of observations
- $x \in \mathbb{R}^n$ is vector of unknown parameters
- $\varepsilon \in \mathbb{R}^m$ is vector of measurement errors
- $r(x) = y - Ax$ is the residual vector

Robust Regression

$$\min_x f(x) \equiv \min_x \sum_{j=1}^{m} \rho(r_j(x)/\sigma),$$

ρ is a given function

- Less sensitive to large residuals, measured with a scale factor β.
- σ is a scale factor connected to data or defined by the method

A Newton Method Approach

For $i \geq 0$

$$\begin{aligned} \text{Solve for } \Delta x^{(i)} : &\quad A^T G^{(i)} A \Delta x^{(i)} = \sigma A^T v^{(i)} \\ \text{Update } : &\quad x^{(i+1)} = x^{(i)} + \alpha^{(i)} \Delta x^{(i)}. \end{aligned}$$

- v is an m-vector with elements $\rho'(r_j(x^{(i)})/\sigma)$.
- $G^{(i)} \in \mathbb{R}^{m \times m}$ (diagonal matrix) $G^{(i)}_{jj} = \rho''(r_j(x^{(i)})/\sigma)$
- $\alpha \in (0, 1]$ is steplength.

Observation: Solving for $\Delta x^{(i)}$ is solving the normal equations

$$A^T G A x = h$$

where only the weight G and h depends on the iteration index i.

Iteratively Reweighted Least Squares (IRLS)

For $i \geq 0$

$$\begin{aligned} \text{Solve for } \Delta x^{(i)} : &\quad A^T G^{(i)} A \Delta x^{(i)} = \sigma A^T v^{(i)} \\ \text{Update } : &\quad x^{(i+1)} = x^{(i)} + \Delta x^{(i)}. \end{aligned}$$

- v is an m-vector with elements $\rho'(r_j(x^{(i)})/\sigma)$.
- $G^{(i)} \in \mathbb{R}^{m \times m}$ (diagonal matrix) $G^{(i)}_{jj} = \frac{\rho'(r_j(x^{(i)})/\sigma)}{r_j(x^{(i)})/\sigma}$.

The IRLS linear system differs from the Newton equation in the weight matrix $G^{(k)}$ and no linesearch.

Observation: Solving for $\Delta x^{(i)}$ is solving the normal equations

$$A^T G A x = h$$

where only the weight G and h depends on the iteration index i.

The ρ Functions

Function	$\rho(z) =$	$\rho''(z) =$	$\rho'(z)/z =$
Huber	$\begin{cases} \frac{z^2}{2} \text{ if } \lvert z\rvert \leq \beta \\ \beta\lvert z\rvert - \frac{\beta^2}{2} \text{ if } \lvert z\rvert > \beta \end{cases}$	$\begin{cases} 1 \text{ if } \lvert z\rvert \leq \beta \\ 0 \text{ if } \lvert z\rvert > \beta \end{cases}$	$\begin{cases} 1 \text{ if } \lvert z\rvert \leq \beta \\ \frac{\beta}{\lvert z\rvert} \text{ if } \lvert z\rvert > \beta \end{cases}$
Talwar	$\begin{cases} \frac{z^2}{2} \text{ if } \lvert z\rvert \leq \beta \\ \frac{\beta^2}{2} \text{ if } \lvert z\rvert > \beta \end{cases}$	$\begin{cases} 1 \text{ if } \lvert z\rvert \leq \beta \\ 0 \text{ if } \lvert z\rvert > \beta \end{cases}$	$\begin{cases} 1 \text{ if } \lvert z\rvert \leq \beta \\ 0 \text{ if } \lvert z\rvert > \beta \end{cases}$
Logistic	$\beta^2 \log(\cosh(\frac{z}{\beta}))$	$1 - \tanh^2(\frac{z}{\beta})$	$(\frac{z}{\beta})^{-1} \tanh(\frac{z}{\beta})$
Fair	$\beta^2 \left(\frac{\lvert z\rvert}{\beta} - \log\left(1 + \frac{\lvert z\rvert}{\beta}\right)\right)$	$\beta^2(\beta + \lvert z\rvert)^{-2}$	$\beta(\beta + \lvert z\rvert)^{-1}$

- Tuning parameters β and σ.
- Theory presented is only valid for Logistic and Fair – but can be expanded.

L_p Approach

The L_p approach is based on

$$\min_x \sum_{j=1}^{m} |r_j(x)|^p\,,\ 1 \le p$$

For $i \ge 0$, we solve for $x^{(i+1)}$ by solving a sequence of problems:

$$A^T G^{(i)} A\, x^{(i+1)} = A^T G^{(i)} y,\ i \ge 0.$$

G is the diagonal matrix with diagonal elements $|r_j(x^{(i)})|^{p-2}$

Observation: Solving for $x^{(i+1)}$ is solving a weighted least squares problem

$$\min_z \|Az - h\|_G$$

where only the weight G depends on the iteration index i.

Linear Programming

Consider the primal linear program

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to:} & Ax = b \\ & x \ge 0 \end{array}$$

and its dual program

$$\begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to:} & A^T y + z = c \\ & z \ge 0 \end{array}$$

where $A \in \mathbb{R}^{m\times n}$ $(m < n)$ has full rank m, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$.

Perturbed KKT conditions

The basic framework of primal-dual interior-point methods can be viewed as applying Newton's method to the perturbed Karush-Kuhn-Tucker conditions

$$F_\mu(x,y,z) = \begin{pmatrix} Ax - b \\ A^T y + z - c \\ XZe - \mu e \end{pmatrix} = 0, \quad (x,z) > 0,$$

for a sequence of positive μ values that decreases to zero.

Here $X = \operatorname{diag}(x)$, and $Z = \operatorname{diag}(z)$ and e is a vector of all ones $e^T = (1,\dots,1)$.

Newton step on perturbed KKT

At a given iterate (x,y,z), where $x,z > 0$, and for a fixed value of μ, the linear equation that defines a Newton step is

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = - \begin{pmatrix} Ax - b \\ A^T y + z - c \\ XZe - \mu e \end{pmatrix}.$$

The next iterate is obtained by the update

$$(x,y,z) \leftarrow (x,y,z) + \alpha(\Delta x, \Delta y, \Delta z)$$

where the step length α is chosen to ensure that the new x and z remain positive.

The linear system as weighted least squares system

The most widely-used approach for solving is to use a block Gaussian elimination and solve the Schur complement

$$(AGA^T)\Delta y = AG(c - A^T y - \mu X^{-1} e) + (b - Ax)$$

where $G = Z^{-1}X$ is a positive definite diagonal matrix. After Δy is determined from solving the above linear system, Δx and Δz can then be easily computed.

Observation: The equations for Δy is weighted normal equations

$$AGA^T \Delta y = h$$

where only the weight G and h changes from one iteration to next.

Optimization 2001, Aveiro, Portugal

Sequence of Weighted Normal equations

Want to solve a *sequence* of problems of the following type

$$AG_iA^T y_i = h_i, \quad i = 0, 1, \ldots$$

- $A \in \mathbb{R}^{m \times n}$, $m \leq n$, rank$(A) = m$.
- $y_i \in \mathbb{R}^m$, $h_i \in \mathbb{R}^n$.
- $G_i \in \mathbb{R}^{n \times n}$, positive definite.

Solution methods:

- **Direct**: Form AGA^T and compute L (Cholesky) $L^T L = AGA^T$
- **Iterative**: PCGLS not forming AGA^T and using a preconditioner
- Try to combine !

Optimization 2001, Aveiro, Portugal

Issues

$$H \equiv G_{i_0}, \quad i_0 \geq 0$$
$$AHA^T = LL^T$$

Want to solve for $i > i_0$

$$AGA^T y = h$$

where $G \equiv G_i$ and $h \equiv h_i$

- How "long" can we use this factorization?
- Can we improve the factorization ?
- Can we combine with preconditioned iterative method with "improved" AHA^T preconditioner?

Optimization 2001, Aveiro, Portugal

The Preconditioner

Want to solve $AGA^T y = h$. Write

$$AGA^T = AHA^T + A(G - H)A^T$$

$G,\ H \in \mathbb{R}^{n \times n}$ diagonal, positive definite.

Choose indexset $\mathcal{Q}$ and define diagonal matrix K:

$$\mathcal{Q} \subseteq \{j : G_{jj} \neq H_{jj}\}, \quad K_{jj} \quad = \quad \begin{cases} G_{jj} & \text{if } j \in \mathcal{Q} \\ H_{jj} & \text{otherwise} \end{cases}$$

Observe that AKA^T is a rank $q = |\mathcal{Q}|$ correction of AHA^T

$$AKA^T = AHA^T + \bar{A}\bar{D}\bar{A}^T$$

$\mathcal{Q} = \{j_1, j_2, \ldots, j_q\}, \quad \bar{A}_k = A_{j_k}, \quad \bar{D} = diag\{G_{j_k j_k} - H_{j_k j_k}\}, \quad 1 \leq k \leq q.$

Choice of Index Set $\mathcal{Q}$

Preconditioned Matrix

$$S = (AKA^T)^{-1}AGA^T$$

Theorem:

Bounds for eigenvalues of S depend on $\mathcal{Q}$:

$$\min_{j\notin\mathcal{Q}} \frac{G_{jj}}{H_{jj}} \;\leq\; \lambda_i(S) \;\leq\; \max_{j\notin\mathcal{Q}} \frac{G_{jj}}{H_{jj}}.$$

(If $G > H$ or $G < H$ then we need a scalar scale of G).

Optimization 2001, Aveiro, Portugal

Choice of Index Set $\mathcal{Q}$

Let

$$\gamma_1 = \min_j \left\{\frac{G_{jj}}{H_{jj}}\right\} \leq \gamma_2 \leq \cdots \leq \gamma_n = \max_j \left\{\frac{G_{jj}}{H_{jj}}\right\}.$$

Choose q and determine q_1 and q_2 so that

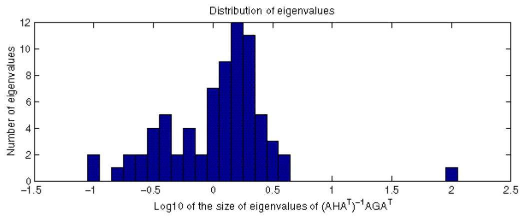
$$\min \frac{\gamma_{n-q_1}}{\gamma_{q_2+1}}$$

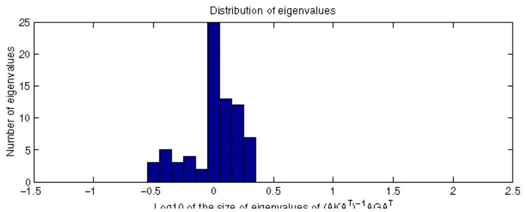
$\mathcal{Q} \equiv$ indices of the q_1 largest and q_2 smallest diagonal elements of $H^{-1}G$.

Corollary: Then

$$\gamma_{q_2+1} \;\leq\; \lambda_i((AKA^T)^{-1}AGA^T) \;\leq\; \gamma_{n-q_1}.$$

Optimization 2001, Aveiro, Portugal





BLEND: 2.IPD: $\mathcal{Q}$ the 20 largest and 10 smallest diagonal elements of $H^{-1}G$.

Optimization 2001, Aveiro, Portugal

Index Set $\mathcal{Q}$

Choose $\mathcal{Q}$ such that:

- AKA^T is better preconditioner than AHA^T.
- $AKA^Ts = v$ is easy to solve.
- $q = |\mathcal{Q}|$ so that the bound on the condition number is minimized.

Observe:

$\mathcal{Q} \neq \emptyset \Rightarrow$

$$AKA^T = AHA^T + \bar{A}\bar{D}\bar{A}^T$$

correction of AHA^T of rank q.

- Use Sherman–Morrison–Woodbury.
- Update Cholesky factors of AHA^T

Sherman–Morrison–Woodbury

Given $AHA^T = LL^T$,

$$AKA^T \quad = \quad LL^T + \bar{A}\bar{D}\bar{A}^T.$$

Want to solve $AKA^T z = r$.

Sherman–Morrison-Woodbury formula yields

$$(AKA^T)^{-1} = L^{-T}(I - VF^{-1}V^T)L^{-1}$$

where $V = L^{-1}\bar{A},\ F = \bar{D}^{-1} + V^T V$.

Then

$$z = (AKA^T)^{-1}r = L^{-T}[d - (V(F^{-1}(V^T d)))]$$

for $d = L^{-1}r$.

Triangular–Factors–Update

Compute R and T such that

$$RTR^T = LDL^T + \bar{A}\bar{D}\bar{A}^T.$$

Set $V = \bar{A}$, $C = \bar{D}$ and $R = 0$.
for $i = 1, \ldots, m$ **do**
  if $V_{i\bullet} \neq 0$ **then**
    $p = (V_{i\bullet}C)^T$, $T_{ii} = D_{ii} + V_{i\bullet}\, p$
    $u = (1/T_{ii})\, p$, $C \leftarrow C - u\, p^T$
    for $j = i+1, \ldots, m$ **do**
      if $L_{ji} \neq 0$ **then**
        $V_{j\bullet} \leftarrow V_{j\bullet} - L_{ji}V_{i\bullet}$
        $R_{ji} = L_{ji} + V_{j\bullet}\, u$

Sparse version of algorithms for dense matrices by

- Bennett, 1965.
- Gill, Golub, Murray and Saunders, 1974.
- Bartels and Kaufman, 1989.

Problem				Maximal q	
Name	m	n	$s\%$	SMW	Update
czprob	737	3141	0.41	18	30
d2q06c	2171	5831	0.26	152	85
d6cube	404	6184	1.51	94	56
stocfor2	2157	3045	0.14	105	69
scsd8	397	2750	0.79	17	13
agg3	516	758	1.22	54	36
maros	835	1921	0.63	49	34

Break-even cost for a solve using Sherman–Morrison–Woodbury or Update (of a Cholesky factorization) in one PCGLS iteration compared to forming AGA^T, a Cholesky and solve. The break even is expressed in terms of q. IPD step 2 and symmetric minimum degree.

Maximal number of PCGLS iterations										
q	stocfor2		czprob		scsd8		d6cube		d2q06c	
	SMW	Δ	SMW	Δ	SMW	Δ	SMW	Δ	SMW	Δ
1	34	35	1	1	2	2	34	35	66	67
2	33	34	1	1	2	2	34	35	65	66
3	33	34	1	1	2	2	34	34	65	66
4	32	33	1	1	2	2	34	34	65	65
5	32	33	1	1	2	2	33	33	64	64
6	32	32	1	1	2	2	33	32	64	63
10	31	30	1	1	1	1	32	30	62	60
30	26	22		1			25	19	53	45
50	20	12					18	5	46	32

The break-even point for using either SMW or triangular update as as a function of q expressed in number of PCGLS iterations compared to one a 'direct step'. IPD iteration 2.

Interior Primal Dual Newton method

Problem	Cholesky		$q = 6$		$q = 20$		$q = 40$	
name	iter	time	iter	time	iter	time	iter	time
czprob	56	89.01	59	59.58	58	56.52	58	56.83
d2q06c	54	544.85	57	421.33	61	442.24	55	405.17
d6cube	42	317.58	45	239.16	47	226.72	46	229.93
fit1p	22	167.89	24	130.01	23	116.83	22	110.32
stocfor2	41	113.26	44	98.05	43	85.32	42	90.74
scsd8	16	21.41	19	17.49	19	17.31	19	17.33

Comparison of Interior Primal Dual Newton method solved with direct method (Cholesky) and alternating between a direct and PCGLS iterative method using SMW.

Optimization 2001, Aveiro, Portugal

Iterative solver: PCGLS

System to be solved: $AGA^Ty = h$ and $y_0 = 0$ is the starting vector

$v_0 = h$, Solve for $s_0 : (AKA^T)s_0 = v_0$, $p_0 = s_0$, $\gamma_0 = s_0^T v_0$

$k \leftarrow 0$

while not converged **and** $k \leq k_{\max}$ **do**

$q_k = A^T p_k$

$\alpha_k = \frac{\gamma_k}{q_k^T G q_k}$

$y_{k+1} = y_k + \alpha_k p_k$

$v_{k+1} = v_k - \alpha_k AG q_k$

Solve for s_{k+1}: $(AKA^T)s_{k+1} = v_{k+1}$

$\gamma_{k+1} = s_{k+1}^T v_{k+1}$

$\beta_k = \frac{\gamma_{k+1}}{\gamma_k}$

$p_{k+1} = s_{k+1} + \beta_k p_k$

$k \leftarrow k + 1$

end {**while**}