Matrices and Statistics with Applications Solution of large sparse Least Square problems

Maya Neytcheva

SeSE, September 2020

Least square problems I

Given $A(m,n)$ with full column rank, $b(n,1)$, consistent with A. We want to solve

$$Ax = b$$

in the Least Squares sense, thus, $x = (A^TA)^{-1}A^Tb$.

We do not want to form A^TA because
– it is usually badly conditioned
– it is in general full even if A is sparse.

A^TA is symmetric positive definite and we have a method for such systems.

The Conjugate Gradient (CG) method

The CG method:

Initialize: $\mathbf{r}^{(0)} = A\mathbf{x}^{(0)} - \mathbf{b}$, $\mathbf{g}^{(0)} = \mathbf{r}^{(0)}$
For $k = 0, 1, \cdots$, until convergence

$$\tau_k = \frac{(\mathbf{r}^{(k)},\mathbf{r}^{(k)})}{(A\mathbf{g}^k,\mathbf{g}^{(k)})}$$
$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \tau_k \mathbf{g}^k$$
$$\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} + \tau_k A\mathbf{g}^k$$
$$\beta_k = \frac{(\mathbf{r}^{(k+1)},\mathbf{r}^{(k+1)})}{(\mathbf{r}^{(k)},\mathbf{r}^{(k)})}$$
$$\mathbf{g}^{k+1} = \mathbf{r}^{(k+1)} + \beta_k \mathbf{g}^k$$

end

$\mathbf{r}^{(k)}$ – iteratively computed residuals
$\mathbf{g}^k$ – search directions

CG: the algorithm

```
x = x0
r = A*x-b
delta0 = (r,r)
g = -r
Repeat: h   = A*g
        tau = delta0/(g,h)
        x   = x + tau*g
        r   = r + tau*h
        delta1 = (r,r)
        if delta1 <= eps, stop
        beta = delta1/delta0
        g    = -r + beta*g
```

Computational complexity of one CG iteration: $O(N)$, where $A(N,N)$, sparse.

Optimality properties of the CG method

Opt1: Mutually orthogonal search directions: $(\mathbf{g}^{k+1}, A\mathbf{g}^j) = 0, j = 0, \cdots, k$

Opt2: There holds $\mathbf{r}^{(k+1)} \perp K_m(A, \mathbf{r}^{(0)}$, i.e., $(\mathbf{r}^{(k+1)}, A\mathbf{r}^{(k)}) = 0, j = 0, \cdots, k$

Opt3: Optimization property: $\|\mathbf{r}^{(k)}\|$ smallest possible at any step, since CG minimizes the functional $f(\mathbf{x}) = 1/2(\mathbf{x}, A\mathbf{x}) - (\mathbf{x}, \mathbf{b})$

Opt4: $(\mathbf{e}^{(k+1), A\mathbf{g}^j)} = (\mathbf{g}^{k+1}, A\mathbf{g}^j) = (\mathbf{r}^{(k+1)}, \mathbf{r}^{(k)}) = 0, j = 0, \cdots, k$

Opt5: Finite termination property: there are no breakdowns of the CG algorithm. Reasoning: if $\mathbf{g}^j = \mathbf{0}$ then τ_k is not defined. the vectors $\mathbf{g}^j$ are computed from the formula $\mathbf{g}^k = \mathbf{r}^{(k)} + \beta_k \mathbf{g}^{k-1}$. Then $0 = (\mathbf{r}^{(k)}, \mathbf{g}^j) = -(\mathbf{r}^{(k)}, \mathbf{r}^{(k)}) + \beta_k \underbrace{(\mathbf{r}^{(k)}, \mathbf{g}^{k-1})}_{0}, \Rightarrow \mathbf{r}^{(k)}\mathbf{0}$, i.e., the solution is already found.
As soon as $\mathbf{x}^{(k)} \neq \mathbf{x}_{exact}$, then $\mathbf{r}^{(k)} \neq \mathbf{0}$ and then $\mathbf{g}^{k+1}) \neq \mathbf{0}$.
However, we can generate at most n mutually orthogonal vectors in R^n, thus, CG has a finite termination property.

Rate of convergence of the CG method

Theorem: Let A is symmetric and positive definite.
Suppose that for some set S, containing all eigenvalues of A, for some polynomial $\widetilde{P}(\lambda) \in \Pi_k^1$ and some constant M there holds $\max_{\lambda \in S} \left|\widetilde{P}(\lambda)\right| \leq M$. Then,

$$\|\mathbf{x}_{exact} - \mathbf{x}^{(k)}\|_A \leq M \|\mathbf{x}_{exact} - \mathbf{x}^{(0)}\|_A.$$

$$\|\mathbf{e^k}\|_A \leq 2\left[\frac{\varkappa(A)+1}{\varkappa(A)-1}\right]^k \|\mathbf{e^0}\|_A$$

$\varkappa(A)$ - the condition number of A, $\| \times \|_A^2 = (x, Ax)$

Rate of convergence (cont)

Repeat:

$$\|\mathbf{e^k}\|_A \leq 2\left[\frac{\varkappa(A)+1}{\varkappa(A)-1}\right]^k \|\mathbf{e^0}\|_A$$

Seek now the smallest k, such that

$$\|\mathbf{e}^k\|_A \leq \varepsilon \|\mathbf{e}^0\|_A$$

$$\text{we want } \left(\tfrac{\varkappa+1}{\varkappa-1}\right)^k > \tfrac{2}{\varepsilon}$$

$$\Rightarrow \; k \; ln\left(\tfrac{\varkappa+1}{\varkappa-1}\right) > ln(\tfrac{2}{\varepsilon})$$

$$\Rightarrow \; k > ln(\tfrac{2}{\varepsilon})/ln\left(\tfrac{\varkappa+1}{\varkappa-1}\right)$$

$$\Rightarrow \; k > \tfrac{1}{2}\sqrt{\varkappa}\; ln(\tfrac{2}{\varepsilon})$$

CG - A Krylov subspace iteration method

Definition of a Krylov subspace, based on a vector $\boldsymbol{v} \in R^n$ and a matrix $B \in R^{n\times n}$,

$$\mathcal{K}_k(B, \boldsymbol{v}) = span\{\boldsymbol{v}, B\boldsymbol{v}, B^2\boldsymbol{v}, \cdots, B^{k-1}\boldsymbol{v}\}.$$

Through the iterations, CG constructs a Krylov subspace, based on A and b.
Remarkably, the solution x lies in that space!

CGLS - Conjugate Gradient for Least Square problems,
i.e., CG for the normal equation

Remember, we do not want to form A^TA!!

CGLS: solve the normal equation for $A \in R^{n\times m}$

History:
CG has appeared in a paper by Hestenes and Stiefel (1952). In that paper and in a followup paper by Stiefel (1952), a version of CG for solving the normal equation has peen presented.
First result for using a preconditioned CG for solving Least Square problems appears in a paper by Peter Läuchli (1959).

CGLS

Recall the definition of a Krylov subspace, based on a vector $\boldsymbol{v} \in R^n$ and a matrix $B \in R^{n\times n}$,

$$\mathcal{K}_k(B, \boldsymbol{v}) = span\{\boldsymbol{v}, B\boldsymbol{v}, B^2\boldsymbol{v}, \cdots, B^{k-1}\boldsymbol{v}\}.$$

The standard CG method minimizes the following functional

$$f(\mathsf{x}) = \frac{1}{2}(\mathsf{x}, A\mathsf{x}) - (\mathsf{x}, \mathbf{b}).$$

Let A be rectangular and denote $A^\dagger$ be its pseudoinverse. Denote $\widehat{\boldsymbol{x}} = A^\dagger\boldsymbol{b}$ - the pseudoinverse solution and the corresponding residual $\widehat{\boldsymbol{r}} = A\widehat{\boldsymbol{x}}$. Then, in the CG framework, $\widehat{\boldsymbol{x}}^k$ minimizes the following error functional:

$$E_\mu(\widehat{\boldsymbol{x}}^k) = (\widehat{\boldsymbol{x}} - \boldsymbol{x}^k)^T(A^TA)^\mu(\widehat{\boldsymbol{x}} + \boldsymbol{x}^k)$$

where $\widehat{\boldsymbol{x}}^k = (x)^0 + \mathcal{K}_k(A^TA, (s)^0)$, $\boldsymbol{s}^0 = A^T(\boldsymbol{b} - A\boldsymbol{x}^0)$.

CGLS I

$$E_\mu(\mathbf{x}^k) = (\widehat{\mathbf{x}} - \mathbf{x}^k)^T (A^T A)^{\mu} (\widehat{\mathbf{x}} + \mathbf{x}^k)$$

Values of μ of practical interest:

$\mu = 0$ minimizes $\|\widehat{\mathbf{x}} - \mathbf{x}^k\|_2^2$

$\mu = 1$ minimizes $\|\widehat{\mathbf{r}} - \mathbf{r}^k\|_2^2 = \|\widehat{\mathbf{r}}\|_2^2 - \|\mathbf{r}^k\|_2^2$
(due to the orthogonality relation $\widehat{\mathbf{r}} \perp \widehat{\mathbf{r}} - \mathbf{r}^k$)

$\mu = 2$ minimizes $\|A^T(\widehat{\mathbf{r}} - \mathbf{r}^k)\|_2^2$

$\mu = 0$ - feasible only for consistent systems.

$\mu = 1$ - CGLS

CGLS II

Properties of CGSL:

- $E_\mu(\mathbf{x}^k)$ decreases monotonically.
- For $\mu = 1, 2$, $E_\nu(\mathbf{x}^k)$ decreases monotonically for all $\nu \le \mu$.
- for $\mu = 1$ also $\mathbf{r}^k$ decreases monotonically.
- The rate of convergence is estimated as follows:

$$E_\mu(\mathbf{x}^k) < 2\left(\frac{\sqrt{\varkappa}-1}{\sqrt{\varkappa}+1}\right)^k E_\mu(\mathbf{x}^0),$$

  where $\varkappa = \varkappa(A^T A)$.
- For $\mu = 1$, both $\|\widehat{\mathbf{r}} - \mathbf{r}^k\|$ and $\|\widehat{\mathbf{x}} - \mathbf{x}^k\|$ decrease monotonically, however $\|A^T \mathbf{r}^k\|$ does oscillate (not due to roundoff errors).

Algorithm CGLS

Unpreconditioned CG

```
x = x0, r = b-A*x
delta0 = (r,r)
g = -r
Repeat: h = A*g
        tau = delta0/(g,h)
        x = x + tau*g
        r = r - tau*h

        delta1 = (r,r)
        if delta1 <= eps, stop
        beta = delta1/delta0
        g = r + beta*g
```

Unreconditioned CGLS

```
x = x0, r = b-A*x
delta0 = (s,s)
g = s = A^T*r
Repeat: h = A*s
        tau = delta0/(h,h)
        x = x + tau*s
        r = r - tau*h
        s = A^T*r
        delta1 = (s,s)
        if delta1 <= eps, stop
        beta = delta1/delta0
        g = s + beta*g
```


CGSL I

Note: $x, g \in R^n$, $r, h \in R^m$, $(A \in R^{n\times m})$

With $\mathbf{s} = A^T(\mathbf{b} - A\mathbf{x})$, by construction, $\mathbf{x}$ minimizes

$$\mathbf{s}^T (A^T A)^{-1} \mathbf{s}$$

over the space $\mathcal{K}_k(A^T A, A^T \mathbf{b})$.
Thus, $\mathbf{s}^k \in T_k$, $T_k = \{A^T(\mathbf{b} - A\mathbf{x}) \,|\, \mathbf{x} \in \mathcal{K}_k(A^T A, A^T \mathbf{b})\}$ and any vector from T_k can be expressed as

$$\mathbf{s}^k = (I - A^T A \mathcal{P}_{k-1}(A^T A))A^T \mathbf{b} = \mathcal{R}_k(A^T A)A^T \mathbf{b},$$

where $\mathcal{P}_{k-1}$ is a polynomial of degree $k-1$ and $\mathcal{R}_k$ is a residual polynomial of degree less than or equal k and is normalized at zero, thus $\mathcal{R}_k(0) = 1$.

CGSL II

$$\|s^k\|_{(A^TA)^{-1}} = \min_{\mathcal{R}\in\Pi_k} \|\mathcal{R}_k(A^TA)A^T\boldsymbol{b}^k\|_{(A^TA)^{-1}}$$

Consider the singular value decomposition of A, $A = U\Sigma V^T$. Then

$$\boldsymbol{b} = \sum_{i=1}^{m} b_i \boldsymbol{u}_i, \quad A^T\boldsymbol{b} = \sum_{i=1}^{n} b_i\sigma_i \boldsymbol{v}_i$$

and

$$\|s^k\|^2_{(A^TA)^{-1}} = \min_{\mathcal{R}\in\Pi_k} \sum_{i=1}^{n} b_i^2 \mathcal{R}_k^2(\sigma_i^2).$$

CGSL III

$$\|s^k\|^2_{(A^TA)^{-1}} = \min_{\mathcal{R}\in\Pi_k} \sum_{i=1}^{n} b_i^2 \mathcal{R}_k^2(\sigma_i^2).$$

Any polynomial from Π_k will give an upper bound. For the choice

$$\mathcal{R}_n(\sigma^2) = \left(1 - \frac{\sigma^2}{\sigma_1^2}\right)\left(1 - \frac{\sigma^2}{\sigma_2^2}\right)\cdots\left(1 - \frac{\sigma^2}{\sigma_n^2}\right)$$

we get $\|s_n\|_{(A^TA)^{-1}} = 0$, which shows the final termination property of CGLS.
If A has only q distinct singular values, then CGLS will converge in at most q iterations.

Preconditioning

CG:

$$Ax = b \to C^{-1}Ax = C^{-1}b$$

such that $\varkappa(C^{-1}A)$ is small, as close as possible to 1.
For CG the important role is played by the eigenvalues of $\varkappa(C^{-1}A)$.

Algorithm: Preconditioned CGLS

A good preconditioner for CGLS: the distinct singular values of the preconditioned matrix should be very few!
The normal equations for the preconditioned problem in factored form:

$$C^{-T}A^T(AC^{-1}\boldsymbol{y} - \boldsymbol{b}) = C^{-T}A^T(A\boldsymbol{x} - \boldsymbol{b}) = 0.$$

The convergence now depends on the condition number $\varkappa(AC^{-1})$.

Algorithm: Preconditioned CGLS

Unpreconditioned CGLS	Preconditioned CGLS
x = x0,	x = x0,
r = b-A*x;	r = b-A*x;
g = s = A^T*r	g = s =C^{-1} A^T*r
delta0 = (s,s)	delta0 = (s,s)
Repeat: h = A*s	Repeat: t=C^{-1}s; h = A*s
tau = delta0/(h,h)	tau = delta0/(h,h)
x = x + tau*s	x = x + tau*t
r = r - tau*h	r = r - tau*h
s = A^T*r	s = $C^{-1}A^T$*r
delta1 = (s,s)	delta1 = (s,s)
if delta1 <= eps, stop	if delta1 <= eps, stop
beta = delta1/delta0	beta = delta1/delta0
g = s + beta*g	g = s + beta*g