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, $x = (A^T A)^{-1} A^T b$.

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

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

The Conjugate Gradient (CG) method

Initialize: $r^{(0)} = Ax^{(0)} - b$, $g^{(0)} = r^{(0)}$

For $k = 0, 1, \ldots$, until convergence

\[
\tau_k = \begin{pmatrix} r^{(k)} & r^{(k)} \\ Ag^{(k)} & g^{(k)} \end{pmatrix} \\
\begin{pmatrix} r^{(k+1)} & r^{(k+1)} \\ Ag^{(k+1)} & g^{(k+1)} \end{pmatrix} = r^{(k+1)} + \beta^k g^k
\]

end

$r^{(k)}$ – iteratively computed residuals
$g^k$ – search directions
CG: the algorithm

\[ x = x_0 \]
\[ r = A \times x - b \]
\[ \delta_0 = (r, r) \]
\[ g = -r \]
Repeat: \[ h = A \times g \]
\[ \tau = \delta_0 / (g, h) \]
\[ x = x + \tau \times g \]
\[ \delta_1 = (r, r) \]
\[ g = -r + \beta \times g \]
Computational complexity of one CG iteration: \( O(N) \), where \( A(N, N) \), sparse.

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 \( P(\lambda) \in \Pi_k^+ \), and some constant \( M \) there holds
\[ \max_{\lambda \in S} |P(\lambda)| \leq M. \]
Then,
\[ \|x_{\text{exact}} - x^{(k)}\|_A \leq M\|x_{\text{exact}} - x^{(0)}\|_A. \]
\[ \|e^k\|_A \leq 2 \left[ \frac{\lambda(A) + 1}{\lambda(A) - 1} \right]^k \|e^0\|_A \]
\[ \lambda(A) \text{ - the condition number of } A, \|x\|_A^2 = (x, Ax) \]

Optimality properties of the CG method

**Opt1:** Mutually orthogonal search directions: \( (g^{k+1}, Ag^j) = 0, j = 0, \cdots, k \)
**Opt2:** There holds \( r^{(k+1)} \perp K_m(A, r^{(0)}), i.e., (r^{(k+1)}, Ar^{(k)}) = 0, j = 0, \cdots, k \)
**Opt3:** Optimization property: \( \|r^{(k)}\| \) smallest possible at any step, since CG minimizes the functional \( f(x) = 1/2(x, Ax) - (x, b) \)
**Opt4:** \( (e^{(k+1)}, Ag^j) = (g^{k+1}, Ag^j) = (r^{(k+1)}, r^{(k)}) = 0, j = 0, \cdots, k \)
**Opt5:** Finite termination property: there are no breakdowns of the CG algorithm.
Reasoning: if \( g^j = 0 \) then \( \tau_j \) is not defined. The vectors \( g^j \) are computed from the formula \( g^j = r^{(k)} + \beta_j g^{k-1} \). Then \( 0 = (r^{(k)}, g^j) = -(r^{(k)}, r^{(k)}) + \beta_j (r^{(k)}, g^{k-1}, 0) \Rightarrow r^{(k)}0, i.e., the solution is already found.
As soon as \( x^{(k)} \neq x_{\text{exact}}, \) then \( r^{(k)} \neq 0 \) and then \( g^{k+1} \neq 0 \). However, we can generate at most \( n \) mutually orthogonal vectors in \( R^n \), thus, CG has a finite termination property.

Rate of convergence (cont)

Repeat:
\[ \|e^k\|_A \leq \varepsilon \|e^0\|_A \]
Seek now the smallest \( k \), such that
\[ \|e^k\|_A \leq \varepsilon \|e^0\|_A \]
we want \( \left( \frac{\lambda + 1}{\lambda - 1} \right)^k > \frac{2}{\varepsilon} \)
\[ \Rightarrow k \ln \left( \frac{\lambda + 1}{\lambda - 1} \right) > \ln \left( \frac{2}{\varepsilon} \right) \]
\[ \Rightarrow k > \ln \left( \frac{2}{\varepsilon} \right) / \ln \left( \frac{\lambda + 1}{\lambda - 1} \right) \]
\[ \Rightarrow k > \frac{1}{2} \sqrt{\lambda} \ln \left( \frac{2}{\varepsilon} \right) \]
Definition of a Krylov subspace, based on a vector $v \in \mathbb{R}^n$ and a matrix $B \in \mathbb{R}^{n \times n}$,

$$K_k(B, v) = \text{span}\{v, Bv, B^2v, \ldots, B^{k-1}v\}.$$  

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

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

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 been 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 $v \in \mathbb{R}^n$ and a matrix $B \in \mathbb{R}^{n \times n}$,

$$K_k(B, v) = \text{span}\{v, Bv, B^2v, \ldots, B^{k-1}v\}.$$  

The standard CG method minimizes the following functional

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

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

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

where $\hat{x}^k = (x)^0 + K_k(A^T A, (s)^0), \ s^0 = A^T (b - Ax^0)$. 

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

Remember, we do not want to form $A^T A$!!

CGLS

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

$$K_k(B, v) = \text{span}\{v, Bv, B^2v, \ldots, B^{k-1}v\}.$$  

The standard CG method minimizes the following functional

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

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

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

where $\hat{x}^k = (x)^0 + K_k(A^T A, (s)^0), \ s^0 = A^T (b - Ax^0)$.
\[ E_\mu(x^k) = (\hat{x} - x^k)^T (A^T A)^\mu (\hat{x} + x^k) \]

Values of \( \mu \) of practical interest:
- \( \mu = 0 \) minimizes \( \|\hat{x} - x^k\|^2 \)
- \( \mu = 1 \) minimizes \( \|\hat{r} - r^k\|^2 = \|\hat{r}\|^2 - \|r^k\|^2 \)
  (due to the orthogonality relation \( \hat{r} \perp \hat{r} - r^k \))
- \( \mu = 2 \) minimizes \( \|A^T (\hat{r} - r^k)\|^2 \)
- \( \mu = 0 \) - feasible only for consistent systems.
- \( \mu = 1 \) - CGLS

Properties of CGSL:
- \( E_\mu(x^k) \) decreases monotonically.
- For \( \mu = 1, 2 \), \( E_\nu(x^k) \) decreases monotonically for all \( \nu \leq \mu \).
- For \( \mu = 1 \) also \( r^k \) decreases monotonically.
- The rate of convergence is estimated as follows:
  \[ E_\mu(x^k) < 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k E_\mu(x^0), \]
  where \( \kappa = \kappa(A^T A) \).
- For \( \mu = 1 \), both \( \|\hat{r} - r^k\| \) and \( \|\hat{x} - x^k\| \) decrease monotonically, however \( \|A^T r^k\| \) does oscillate (not due to roundoff errors).

Algorithm CGLS

Unpreconditioned CG
\[ x = x_0, \ r = b - A^* x \]
\[ \text{delta0} = (r,r) \]
\[ g = -r \]
Repeat: \( h = A^* g \)
\[ \tau = \text{delta0}/(g,h) \]
\[ x = x + \tau \cdot g \]
\[ r = r - \tau \cdot h \]
\[ \text{delta1} = (r,r) \]
if \( \text{delta1} < \epsilon \) stop
\[ \beta = \text{delta1}/\text{delta0} \]
\[ g = r + \beta \cdot g \]

Unreconditioned CGLS
\[ x = x_0, \ r = b - A^* x \]
\[ \text{delta0} = (s,s) \]
\[ g = s = A^T * r \]
Repeat: \( h = A^* s \)
\[ \tau = \text{delta0}/(h,h) \]
\[ x = x + \tau \cdot s \]
\[ r = r - \tau \cdot h \]
\[ \text{delta1} = (s,s) \]
if \( \text{delta1} < \epsilon \) stop
\[ \beta = \text{delta1}/\text{delta0} \]
\[ g = s + \beta \cdot g \]

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

With \( s = A^T (b - Ax) \), by construction, \( x \) minimizes
\[ s^T (A^T A)^{-1} s \]
over the space \( K_k(A^T A, A^T b) \).
Thus, \( s^k \in T_k, \ T_k = \{A^T(b - Ax) \mid x \in K_k(A^T A, A^T b)\} \) and any vector from \( T_k \) can be expressed as
\[ s^k = (I - A^T A P_{k-1} A^T A) A^T b = R_k(A^T A) A^T b, \]
where \( P_{k-1} \) is a polynomial of degree \( k - 1 \) and \( R_k \) is a residual polynomial of degree less than or equal \( k \) and is normalized at zero, thus \( R_k(0) = 1 \).
Consider the singular value decomposition of $A$, $A = U\Sigma V^T$. Then

$$b = \sum_{i=1}^{m} b_i u_i, \quad A^T b = \sum_{i=1}^{n} b_i \sigma_i v_i$$

and

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

Any polynomial from $\Pi_k$ will give an upper bound. For the choice $R_n(\sigma^2) = \left(1 - \frac{\sigma^2}{\sigma_1^2}\right) \left(1 - \frac{\sigma^2}{\sigma_2^2}\right) \cdots \left(1 - \frac{\sigma^2}{\sigma_n^2}\right)$ we get $\|s_n\|_{(A^T A)^{-1}} = 0$, 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.

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:

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

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

Preconditioning

CG:

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

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

Algorithm: Preconditioned CGLS
Algorithm: Preconditioned CGLS

x = x0,

r = b - A*x;

g = s = A^T*r

delta0 = (s,s)

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

Preconditioned CGLS

x = x0,

r = b - A*x;

g = s = C^{-1} A^T*r

delta0 = (s,s)

Repeat: t = C^{-1}s; h = A*s

tau = delta0/(h,h)

x = x + tau*t

r = r - tau*h

s = C^{-1}A^T*r

delta1 = (s,s)

if delta1 <= eps, stop

beta = delta1/delta0

g = s + beta*g