4. Iterative methods and regularization

Consider the "simple" problem of estimating $x \in \mathbb{R}^n$ that satisfies

$$Ax = b \in \mathbb{R}^m, \quad A \in \mathbb{R}^{n \times m}.$$

Iterative methods, instead of producing an estimate $\hat{x}$ in one batch operation, produce a sequence $x_0, x_1, x_2, \ldots$ $x_j \in \mathbb{R}^n$ that hopefully converges to a reasonable estimate of x.

We consider here one particular iterative method, the Conjugate Gradient (CG) method. Let us assume that

(1) $A \in \mathbb{R}^{n \times n}$ is symmetric : $A^T = A$

(2) A is positive definite : For every $x \neq 0$, $x^T A x > 0$,

or, briefly, A is a SPD matrix.

Before going further, let us see what SPD means: If A is symmetric, it has an SVD of the type

$$A = U D U^T,$$

$U \in \mathbb{R}^{n \times n}$ orthogonal, $D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$. The SVD is also the eigenvalue decomposition of A: if $U = [u_1, u_2, \ldots, u_n]$,

$$A u_j = d_j u_j$$

In particular,

$$d_j = u_j^T (d_j u_j) = u_j^T A u_j > 0,$$

i.e., all eigenvalues are positive.

The matrix A defines an A-norm,

$$\|x\|_A^2 = x^T A x$$

Since $\{u_1, u_2, \ldots, u_n\}$ is an orthogonal basis, we can write

$$x = \sum_{j=1}^{n} \tilde{x}_j u_j \quad , \quad \tilde{x}_j = u_j^T x$$

Then

$$x^T A x = \left(\sum_{j=1}^{n} \tilde{x}_j u_j \right)^T \left(\sum_{k=1}^{n} \tilde{x}_k d_j u_j \right)$$

$$= \sum_{j=1}^{n} d_j \tilde{x}_j^2 = \|x\|_A^2 ,$$

so $\|\cdot\|_A$ is a weighted Euclidean norm in the coordinate frame $\{u_1, \ldots, u_n\}$.

Let $x_* \in \mathbb{R}^n$ denote the <u>true</u> solution of $Ax = b$, i.e., $x_* = A^{-1}b$. This, in the case of perfect, errorless measurement, we would like to find.

For any $x \in \mathbb{R}^n$, define

$$e = x - x_* \qquad \text{(error)}$$

$$r = Ae = Ax - Ax_* = Ax - b, \qquad \text{(residual)}$$

To measure the size of the error, define

$$\varphi(x) = \|e\|_A^2 = e^T A e = e^T r$$

$$= r^T A^{-1} r$$

Without knowing x_*, the functional $\varphi(x)$ cannot be evaluated. Notice that

$$\varphi(x) \geq 0, \quad \varphi(x) = 0 \iff x = x_*$$

Thus, minimizing φ would lead to the true solution x_*. It turns out that while φ cannot be evaluated, it can be still minimized iteratively.

CG algorithm is the most classical of the Krylov subspace methods. Given an initial guess $x_1 \in \mathbb{R}^n$ (often $x_1 = 0$), define the initial residual

$$r_1 = b - Ax_1$$

and assume that $r_1 \neq 0$. The Krylov subspace of order k is

$$\mathcal{K}_k(r_1, A) = sp\{r_1, Ar_1, A^2 r_1, \dots, A^{k-1} r_1\}$$

In Krylov subspace methods, the idea is to look for an approximation in $\mathcal{K}_k(r_1, A)$. The rationale is that $\mathcal{K}_k(r_1, A)$ can be generated by applying the matrix A on a vector, i.e., by calculating matrix-vector-products only. In fact, one does not have to know the matrix A, it is enough to know how it acts on a vector.

Let $x_1 \in \mathbb{R}^n$ be the initial guess, and

$$r_1 = b - Ax_1,$$

and assume that $r_1 \neq 1$. Define the first <u>search direction</u>

$$s_1 = r_1.$$

We seek to minimize the function

$$\mathbb{R} \to \mathbb{R}, \quad \alpha \mapsto \varphi(x_1 + \alpha s_1).$$

By definition,

$$\varphi(x_1 + \alpha s_1) = (x_1 + \alpha s_1 - x_*)^T A (x_1 + \alpha s_1 - x_*)$$

$$= \alpha^2 s_1^T A s_1 + 2\alpha s_1^T \underbrace{A(x_1 - x_*)}_{Ax_1 - b = -r_1} + (x_1 - x_*)^T A (x_1 - x_*)$$

$$= \alpha^2 s_1^T A s_1 - 2\alpha s_1^T r_1 + \varphi(x_1),$$

and the minimum is attained at the zero of the derivative,

$$2\alpha s_1^T A s_1 - 2 s_1^T r_1 = 0$$

or

$$\alpha = \alpha_1 = \frac{s_1^T r_1}{s_1^T A s_1}.$$

Define the new iterate x_2 by

$$x_2 = x_1 + \alpha_1 s_1$$

The new residual is defined as

$$r_2 = b - Ax_2 = r_1 - \alpha_1 A s_1$$

For later reference, observe that

$$s_1^T r_2 = s_1^T r_1 - \alpha_1 s_1^T A s_1 = 0,$$

i.e., the new residual is perpendicular to the old search direction.

We continue in the same fashion: pick a new search direction s_2 and minimize

$$\alpha \longmapsto \varphi(x_2 + \alpha s_2).$$

In general, at the k:th iteration, set

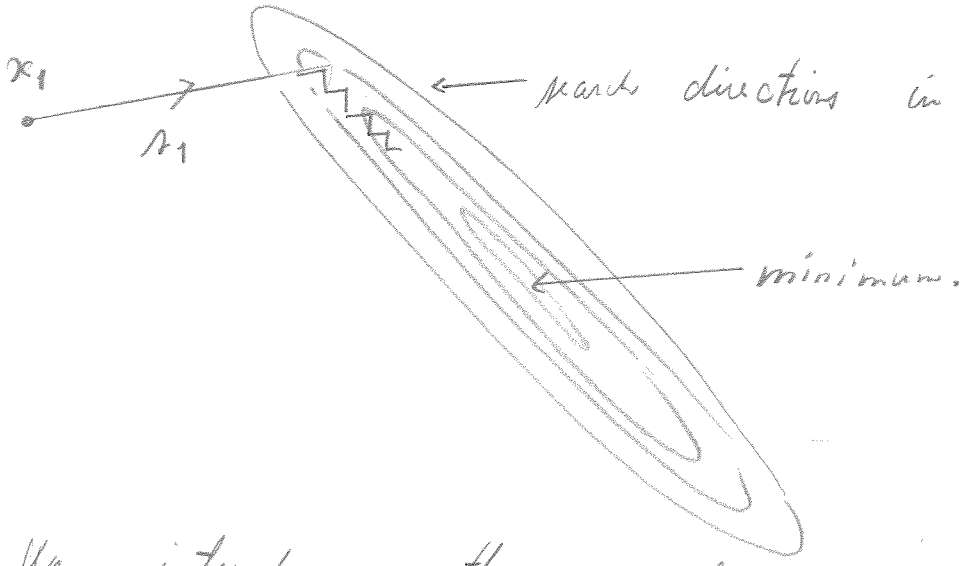
$$\alpha_k = \arg\min \varphi(x_k + \alpha s_k)$$

$$= \frac{s_k^T r_k}{s_k^T A s_k}, \quad k = 1, 2, \ldots$$

But how do we choose the search directions? The choice $s_k = r_k$ looks natural, but it is not very smart. The reason is that

$$r_{k+1} \perp s_k$$

by the choice of α_k, and it means that the iterative scheme may become very slow, see the figure below:



We introduce the concept of <u>A-conjugate</u> directions:
The linearly independent vectors $\{s_1, s_2, \dots, s_k\}$ are <u>A-conjugate</u>, if

$$s_j^T A s_l = 0 \quad , \text{if} \quad j \neq l$$

The idea is to select the search directions with the following two criteria:

(1) $\{s_1, s_2, \dots s_k\}$ are A-conjugate

(2) $sp\{s_1, s_2, \dots s_k\} = \mathcal{K}_k(r_1, A)$

Before explaining how to accomplish this choice, let us look at some consequences of this choice. First, define the matrix

$$S_k = [s_1, s_2, \dots, s_k] \in \mathbb{R}^{n \times k}$$

Thus,

$$(S_k^T A S_k)_{ij} = s_i^T A s_j = \begin{cases} \lambda_i > 0, & \text{if } i = j \\ 0, & \text{if } i \neq 0 \end{cases}$$

that is,

$$S_k^T A S_k = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_k \end{bmatrix} := \Lambda \in \mathbb{R}^{k \times k}$$

Assume now that we seek to minimize the mapping $\varphi(x)$ in $x_1 + \operatorname{sp}\{s_1, s_2, \ldots, s_k\}$, i.e., we seek the vector $\alpha \in \mathbb{R}^k$ that minimizes

$$\alpha \mapsto \varphi(x_1 + S_k \alpha).$$

By definition,

$$\varphi(x_1 + S_k \alpha) = (x_1 + S_1 \alpha - x_*)^T A (x_1 + S_1 \alpha - x_*)$$

$$= \alpha^T \underbrace{(S_1^T A S_1)}_{=\Lambda} \alpha + 2 \alpha^T S_1^T \underbrace{A (x_1 - x_*)}_{=r_1} + \varphi(x_1)$$

$$= \sum_{j=1}^{k} (\alpha_j^2 \lambda_j - 2 \alpha_j s_j^T r_1) + \varphi(x_1),$$

so we see that the minimizer is

$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{bmatrix}, \qquad \alpha_j = \frac{s_j^T r_1}{s_j^T A s_j}$$

But this is the same result as what we obtained before using one-dimensional line searches, since

$$s_k^T r_k = s_k^T (b - Ax_k) = s_k^T (b - A(x_{k-1} + \alpha_{k-1} s_{k-1}))$$

$$= \alpha_{k-1} \underbrace{s_k^T A s_{k-1}}_{=0} + s_k^T r_{k-1}$$

$$= \dots = s_k^T r_1 .$$

<u>Conclusion</u>: Using A-conjugate search directions, the minimization can be done <u>iteratively</u>, <u>one dimension</u> <u>at a time</u>.

Notice also that now

$$x_{k+1} = x_1 + S_k \alpha ,$$

so

$$r_{k+1} = b - Ax_{k+1} = r_1 - AS_k\alpha$$

In particular,

$$S_k^T r_{k+1} = S_k^T r_1 - S_k^T A S_k \alpha = 0$$

by the choice of α, that is,

$$r_{k+1} \perp sp\{s_1, s_2, s_3, \dots, s_k\} .$$

How do we select the search directions?

The choice of s_1 was

$$s_1 = r_1$$

that is,

$$sp\{s_1\} = \mathcal{K}_1(r_1, A).$$

Assume now that we have chosen $s_1, s_2, \ldots, s_k$ A-conjugate and

$$sp\{s_1, \ldots, s_k\} = \mathcal{K}_k(r_1, A),$$

and let

$$r_{k+1} = b - Ax_{k+1}.$$

We assume that $r_{k+1} \neq 0$, otherwise the search for x_* has converged. Search s_{k+1} as

$$s_{k+1} = r_{k+1} + \beta s_k$$

To guarantee A-conjugacy, we must have

$$s_k^T A s_{k+1} = 0 = s_k^T A r_{k+1} + \beta s_k^T A s_k,$$

implying that

$$\beta = \beta_k = - \frac{s_k^T A r_{k+1}}{s_k^T A s_k}.$$

Notice that for $j < k$,

$$s_j^T A s_{k+1} = s_j^T A r_{k+1} + \beta \underbrace{s_j^T A s_k}_{=0}$$

$$= (A s_j)^T r_{k+1}$$

and

$$A s_j \in A\left(sp\{s_1, \dots, s_{k-1}\}\right) = A\, \mathcal{K}_{k-1}(r_1, A)$$

$$\subset \mathcal{K}_k(r_1, A) = sp\{s_1, \dots, s_k\} \perp r_{k+1},$$

so

$$s_j^T A s_{j+1} = 0.$$

So we have indeed found a new direction s_{k+1} which is A-conjugate to $s_1, \dots, s_k$,

and

$$sp\{s_1, \dots, s_{k+1}\} = sp\{s_1, \dots, s_k, r_{k+1}\}$$

$$= sp\left(\mathcal{K}_k(r_1, A) \cup \{r_{k+1}\}\right)$$

$$= \mathcal{K}_{k+1}(r_1, A).$$

We are ready to collect the results in an algorithm

Before that, few remarks:

(1) When calculating α_k, observe that

$$s_k^T r_k = (r_k + \beta_{k-1} s_{k-1})^T r_k = \|r_k\|^2,$$

since $r_k \perp s_{k-1}$, so $\alpha_k = \dfrac{\|r_k\|^2}{s_k^T A s_k}$

(2) When calculating β_k, notice that

$$\|r_{k+1}\|^2 = r_{k+1}^T r_{k+1} = r_{k+1}^T (b - A x_{k+1})$$

$$= r_{k+1}^T (\underbrace{b - A x_k}_{= r_k} - \alpha_k A s_k)$$

$$= r_{k+1}^T r_k - \alpha_k r_{k+1}^T A s_k$$

$$= -\alpha_k r_{k+1}^T A s_k,$$

since $r_k \in \mathcal{K}_k(r_1, A) = sp\{s_1, \dots, s_k\} \perp r_{k+1}$.

By substituting α_k and comparing to the formula of β_k, we find

$$\|r_{k+1}\|^2 = -\frac{\|r_k\|^2}{s_k^T A s_k} r_{k+1}^T A s_k = \|r_k\|^2 \beta_k,$$

so $\beta_k = \dfrac{\|r_{k+1}\|^2}{\|r_k\|^2}$

CG - algorithm for solving $Ax = b$.

Choose x_1

$k = 1, \quad r_1 = b - Ax_1, \quad s_1 = r_1$

Iterate until stopping criterion satisfied

$$\alpha_k = \frac{\|r_k\|^2}{s_k^T A s_k}$$

$$x_{k+1} = x_k + \alpha_k s_k$$

$$r_{k+1} = r_k - \alpha_k A s_k$$

$$\beta_k = \frac{\|r_{k+1}\|^2}{\|r_k\|^2}$$

$$s_{k+1} = r_{k+1} + \beta_k s_k$$

$$k \leftarrow k+1$$

end.