1 Recap

Consider $f(x) = \frac{1}{2}x^\top Ax - b^\top x$, where $A \succ 0$. Minimizing f is equivalent to solving the linear system $Ax = b$.

The conjugate gradient (CG) method is given by

$$x_k = \arg\min_{x \in x_0 + \mathcal{K}_k} f(x), \qquad k = 1, 2, \ldots,$$

where $\mathcal{K}_k := \mathrm{Lin}\{A(x_0 - x^*), \ldots, A^k(x_0 - x^*)\}$ is the *Krylov subspace* of order k.

Lemma 1. *For any $k \geq 1$, we have $\mathcal{K}_k = \mathrm{Lin}\{\nabla f(x_0), \ldots, \nabla f(x_{k-1})\}$.*

Lemma 2. *For any $0 \leq i < k$, we have $\langle \nabla f(x_k), \nabla f(x_i) \rangle = 0$.*

Corollary 1. *CG finds $x^* = \arg\min_{x \in \mathbb{R}^d} f(x)$ in at most d iterations.*

Corollary 2. $\forall p \in \mathcal{K}_k$, $\langle \nabla f(x_k), p \rangle = 0$.

$f(x) = \frac{1}{2}x^TAx - b^Tx$,

$\nabla f(x) = Ax - b$. $\quad \nabla f(x^*) = 0 \quad Ax^* = b$

$= A(x - x^*)$

2 Efficient implementation of CG

Define $\delta_i := x_{i+1} - x_i$.

Lemma 3. *For all $k \geq 1$, $\mathcal{K}_k = \mathrm{Lin}\{\delta_0, \delta_1, \ldots, \delta_{k-1}\}$.*

pf: By Induction. $k=1$. $x_1 = \arg\min_{x \in x_0 + \mathcal{K}_1} f(x) \Rightarrow \mathcal{K}_1 = \mathrm{Lin}\{\delta_0\}$.

Suppose $\mathcal{K}_k = \mathrm{Lin}\{\delta_0, \ldots, \delta_{k-1}\}$. We show that $\mathcal{K}_{k+1} = \mathrm{Lin}\{\delta_0, \ldots, \delta_k\}$.

① $\nabla f(x_k) = 0$. As lemma 1's proof, $\mathcal{K}_{k+1} = \mathcal{K}_k$, $x_{k+1} = x_k = x^*$.

$$\underset{\mathcal{K}_k = \mathcal{K}_{k-1}}{\mathrm{Lin}\{\delta_0, \delta_1, \ldots, \delta_{k-1}\}} = \mathrm{Lin}\{\delta_0, \delta_1, \ldots, \delta_{k-1}, \underset{x_{k+1} - x_k}{0}\} = \mathrm{Lin}\{\delta_0, \ldots, \delta_k\}.$$

② $\nabla f(x_k) \neq 0$. Follow lemma 1's pf,

$$x_{k+1} = x_0 + \sum_{i=1}^{k} \beta_{i,k+1} A^i(x_0 - x^*) + \beta_{k+1,k+1} A^{k+1}(x_0 - x^*), \text{ and } \beta_{k+1,k+1} \neq 0.$$

$$\delta_k = x_{k+1} - x_k = \underbrace{x_0 - x_k}_{\in \mathcal{K}_k} + \underbrace{\sum_{i=1}^{k} \beta_{i,k+1} A^i(x_0 - x^*)}_{\in \mathcal{K}_k} + \beta_{k+1,k+1} A^{k+1}(x_0 - x^*)$$

Hence $\mathrm{Lin}\{\delta_0, \ldots, \delta_{k-1}, \delta_k\} \overset{\text{ass.}}{=} \mathrm{Lin}\{\mathcal{K}_k \cup \delta_k\}$

$$= \mathrm{Lin}\{\mathcal{K}_k \cup A^{k+1}(x_0 - x^*)\} = \mathcal{K}_{k+1}.$$

Lemma 4 (Lemma 1.3.3 in Nesterov's book). *For any $k, i \geq 0$, $k \neq i$, the vectors δ_i, δ_k are conjugate w.r.t. A, i.e., $\langle A\delta_k, \delta_i \rangle = 0$.*

pf: Assume $k > i$ w.l.o.g.

$$\langle A\delta_k, \delta_i \rangle = \langle A(x_{k+1} - x_k), \delta_i \rangle = \langle A(x_{k+1} - x^*) - A(x_k - x^*), \delta_i \rangle$$

$$= \langle A(x_{k+1} - x^*), \delta_i \rangle - \langle A(x_k - x^*), \delta_i \rangle$$

$$= \langle \nabla f(x_{k+1}), \delta_i \rangle - \langle \nabla f(x_k), \delta_i \rangle = 0 - 0 = 0.$$

$$\delta_i = x_{i+1} - x_i \in \mathcal{K}_{i+1} \subseteq \mathcal{K}_k \subseteq \mathcal{K}_{k+1}.$$

By Collary 2, $\begin{cases} \delta_i \in \mathcal{K}_k \Rightarrow \langle \nabla f(x_k), \delta_i \rangle = 0. \\ \delta_i \in \mathcal{K}_{k+1} \Rightarrow \langle \nabla f(x_{k+1}), \delta_i \rangle = 0. \end{cases}$

Explicit formula for CG, Derivation.

$\mathcal{K}_k = \operatorname{Lin}\{\delta_0, \cdots, \delta_{k-1}\}$.

We can express $x_{k+1} \in x_0 + \mathcal{K}_{k+1}$ as follows:

$$x_{k+1} = \underbrace{x_k}_{\in x_0 + \mathcal{K}_k} - \underbrace{h_k \nabla f(x_k)}_{\in \mathcal{K}_{k+1} / \mathcal{K}_k} + \underbrace{\sum_{j=0}^{k-1} \alpha_j \delta_j}_{\in \mathcal{K}_k}.$$

for some h_k, $\alpha_0 \sim \alpha_{k-1}$.

$$\Leftrightarrow \delta_k = -h_k \nabla f(x_k) + \sum_{j=0}^{k-1} \alpha_j \delta_j.$$

To explicitly implement, we need h_k, $\alpha_0 \sim \alpha_{k-1}$.

For $i = 0, \cdots, k-1$,

$$0 = \langle A\delta_i, \delta_k \rangle = -h_k \langle A\delta_i, \nabla f(x_k) \rangle + \sum_{j=0}^{k-1} \alpha_j \langle A\delta_i, \delta_j \rangle$$

$$= -h_k \langle A\delta_i, \nabla f(x_k) \rangle + \alpha_i \langle A\delta_i, \delta_i \rangle$$

Lemma 4.

$$\Rightarrow \alpha_i \langle A\delta_i, \delta_i \rangle = h_k \langle A\delta_i, \nabla f(x_k) \rangle$$

$$= h_k \langle A(x_{i+1} - x^*) - A(x_i - x^*), \nabla f(x_k) \rangle$$

$$= h_k \langle \nabla f(x_{i+1}) - \nabla f(x_i), \nabla f(x_k) \rangle$$

$i = 0, 1, \cdots, k-2.$ $\Rightarrow \langle \nabla f(x_{i+1}), \nabla f(x_k) \rangle = 0$, $\langle \nabla f(x_i), \nabla f(x_k) \rangle = 0$ $\Rightarrow$ $\alpha_i \langle A\delta_i, \delta_i \rangle = 0$

$\because A \succ 0$ $\therefore \alpha_i = 0.$

$i = k-1$ $\quad \alpha_{k-1} \langle A\delta_{k-1}, \delta_{k-1} \rangle = h_k \|\nabla f(x_k)\|_2^2$ $\quad \alpha_{k-1} = \dfrac{h_k \|\nabla f(x_k)\|_2^2}{\langle \nabla f(x_k) - \nabla f(x_{k-1}), \delta_{k-1} \rangle}$

$$\therefore x_{k+1} = x_k - h_k \nabla f(x_k) + \frac{h_k \|\nabla f(x_k)\|_2^2}{\langle \nabla f(x_k) - \nabla f(x_{k-1}), \delta_{k-1} \rangle} \delta_{k-1}$$

$$= x_k - h_k \underbrace{\left(\nabla f(x_k) - \frac{\|\nabla f(x_k)\|_2^2}{\langle \nabla f(x_k) - \nabla f(x_{k-1}), \delta_{k-1} \rangle} \delta_{k-1} \right)}_{:= p_k \in \mathbb{R}^d.}$$

p_k: search direction

h_k: step size.

Let's set h_k.

h_k is given by exact line search.

$$x_{k+1} = x_k - h p_k \in x_0 + \mathcal{K}_{k+1} \Rightarrow h_k = \underset{h \in \mathbb{R}}{\arg\min}\, f(x_k - h p_k)$$

Explicit form of CG: In summary, CG can be implemented as

$$x_{k+1} = x_k - h_k p_k,$$

where

$$p_k = \nabla f(x_k) - \frac{\|\nabla f(x_k)\|_2^2}{\langle \nabla f(x_k) - \nabla f(x_{k-1}), \delta_{k-1} \rangle} \delta_{k-1},$$
$$\delta_{k-1} = x_k - x_{k-1},$$
$$h_k = \arg\min_{h \in \mathbb{R}} f(x_k - h p_k).$$

Easy to compute.

Note that the exact line search step involves optimizing a one-dimensional quadratic function and can be computed in closed form.

Q: How much storage is needed in CG? Computation per iteration?

Remark 1 (Conjugacy). The search directions $p_k = -\frac{1}{h_k}\delta_k$ are conjugate w.r.t. A:

$$\langle Ap_k, p_i \rangle = 0, \qquad \forall k \neq i$$

since $\langle A\delta_k, \delta_i \rangle = 0$ (Lemma 4).

Remark 2 (Relation to heavy-ball). From (1) we have

$$x_{k+1} = x_k - h_k \nabla f(x_k) + \alpha_{k-1}(x_k - x_{k-1}),$$

which resembles the heavy-ball method (gradient step + momentum step) but with time-varying h_k and α_k.

Remark 3. CG does not require knowing the smoothness and strong convexity parameters L and m.

Remark 4. CG for quadratic f has a very rich convergence theory beyond the asymptotic linear rate. For example:

- If A has r distinct eigenvalues, CG terminates in at most r iterations.
- More generally, CG converges fast when the eigenvalues of A have a clustering structure.
- Precondition CG: one may transform the problem so that A has a more favorable eigenvalue distribution.

We will not delve into these results; see Chapter 5.1 of Nocedal-Wright.

3 Extension to non-quadratic functions

We have written CG in a form that only involves the gradient of f, without explicit dependence on the quadratic structure of f. This allows extension to non-quadratic functions. (Such extensions are known as "nonlinear CG", since $\nabla f(x)$ is nonlinear in x.)

Algorithm 1 Nonlinear CG

- Initial search direction: $p_0 = \nabla f(x_0)$.
- For $k = 0, 1, \ldots$
  - Set
  $$x_{k+1} = x_k - h_k p_k,$$
  where h_k is computed by (exact or inexact) line search.
  - Compute the next search direction as
  $$p_{k+1} = \nabla f(x_{k+1}) - \beta_k p_k,$$
  with some specific choice of β_k (see below).

There are different ways of choosing β_k's:

- Dai-Yuan: $\beta_k = \frac{\|\nabla f(x_{k+1})\|_2^2}{\langle \nabla f(x_{k+1}) - \nabla f(x_k), p_k \rangle}$. (equivalent to the α_{k-1} that we derived for quadratic f)
- Fletcher-Rieves: $\beta_k = -\frac{\|\nabla f(x_{k+1})\|_2^2}{\|\nabla f(x_k)\|_2^2}$.
- Polak-Ribiere: $\beta_k = -\frac{\langle \nabla f(x_{k+1}), \nabla f(x_{k+1}) - \nabla f(x_k) \rangle}{\|\nabla f(x_k)\|_2^2}$.

All of above lead to the same results in the case of quadratic f. See Chapter 5.2 of Nocedal-Wright for more on nonlinear CG.

Nonlinear CG is attractive in practice: it does not require matrix storage and performs well empirically (e.g., faster than GD). Theoretical results are not as strong as AGD—this is a topic for further research.