Krylov subspace revisited

For a given A_{nxn} and $b \in \mathbb{R}^n$, r^{th} order Krylov subspace is the subspace spanned by the images of b under r powers of A i.e,

$$\kappa_r(A, b) = span\{b, Ab, A^2b, A^3b, ..., A^{r-1}b\}$$

Conjugate Direction Methods

Conjugate direction methods aim to achieve faster than steepest descent convergence without incurring the compute cost of inverting Hessian (like Newton's method) by instead using Hessian information to chose search directions with the property *conjugacy*

Recall two direction $a, b \in \mathbb{R}^n$ are said to be H-conjugate if,

$$a^T Hb = 0$$

As we have seen in previous lectures, generating iterates using n H-conjugate vectors $\{p_0, p_1, \ldots, p_{n-1}\}$ as search directions,

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

guarantees convergence to minima in at most n steps for an n-dimensional unconstrained quadratic minimization problem

$$\phi(x) = \frac{1}{2}x^T Hx + b^T x$$

where H is symmetric, positive definite, α_k is the minimizer of ϕ along p_k
Let's define residual r_k as,

$$r_k = \nabla\phi(x)|_{x=x_k}$$

Setting $\nabla\phi = 0$ results in the following linear system

$$Hx + b = 0$$

Hence Conjugate Gradient method to minimize quadratic objectives of the form ϕ is referred to as **linear conjugate gradient method**

Note : for quadratic objective ϕ, α_k has a closed form solution

$$\alpha_k = \frac{r_k^T r_k}{p_k^T H p_k}$$

whereas, for a general non-linear objective f, we resort to iterative methods to find α_k

The conjugate direction set $\{p_0, p_1, \ldots, p_{n-1}\}$ can be generated in many ways.

Linear Conjugate Gradient method

Conjugate Gradient (CG) method is a special conjugate direction method that generates the conjugate direction set in a compute and memory efficient manner. CG computes the new conjugate vector, i.e search direction, p_k using only p_{k-1} and residual r_k

$$p_k = -r_k + \beta_k p_{k-1}$$

$$\beta_k = \frac{p_{k-1}^T H r_k}{p_{k-1}^T H p_{k-1}}$$

Algorithm (CG)

Given x_0
Set $r_0 \leftarrow \nabla\phi|_{x=x_0}$, $p_0 \leftarrow -r_0$, $k \leftarrow 0$
while($r_k \neq 0$):

$$\alpha_k \leftarrow \frac{r_k^T r_k}{p_k^T H p_k}$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k$$

$$r_{k+1} \leftarrow r_k + \alpha_k H p_k$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k$$

$$k \leftarrow k + 1$$

end (while)

Theorem : Suppose that the k^{th} iterate generated by the conjugate gradient method is not the solution point x^* . The following four properties hold,

$$r_k^T r_i = 0 \quad for\ i \in \{0, 1, 2, \ldots, k-1\}$$

$$span\{r_0, r_1, \ldots, r_k\} = \kappa_k(H, r_0)$$

$$span\{p_0, p_1, \ldots, p_k\} = \kappa_k(H, r_0)$$

$$p_k^T H p_i = 0 \quad for\ i \in \{0, 1, 2, \ldots, k-1\}$$

Extending CG to nonlinear optimization

So far we have looked at CG algorithm and its properties for minimizing the quadratic ϕ Now we look at how to adapt CG for a general non-linear function f

Consider the minimization of the non-linear objective f along p_{k-1}. Let,

$$\alpha^*_{k-1} = argmin_\alpha f(x_{k-1} + \alpha p_{k-1})$$

As noted earlier, for a general non-linear objective f, there is no closed-form solution for $\alpha_{k-1} = \alpha^*_{k-1}$ Line search is therefore used to compute α_{k-1}

By definition,

$$p_k = -\nabla f_k + \beta_k p_{k-1}$$

p_k is a descent direction when, $\nabla f_k^T p_k < 0$

$$\nabla f_k^T p_k = -||\nabla f_k||^2 + \beta_k \nabla f_k^T p_{k-1}$$

When $\alpha_{k-1} = \alpha^*_{k-1}$,

$$\nabla f_k^T p_{k-1} = 0$$

and therefore,

$$\nabla f_k^T p_k = -||\nabla f_k||^2 < 0$$

When $\alpha_{k-1} \neq \alpha^*_{k-1}$,

β_k determines whether p_k is a descent direction. Error in α_{k-1} estimated by line search leads to errors in β_k.

Note that a large enough β_k could make p_k an ascent direction. This imposes constraints on acceptable values of step length α_{k-1}

Imposing strong Wolfe conditions on α_{k-1} ensures that p_{k-1} is always a descent direction. There are different variants of CG method that differ from each other primarily in the choice of the parameter β_k

THE FLETCHER–REEVES METHOD approximates β_k as,

$$\beta_k^{FR} = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$$

where,

$$r_k = \nabla f_k$$

THE POLAK–RIBIÈRE METHOD approximates β_k as,

$$\beta_k^{PR} = \frac{r_k^T (r_k - r_{k-1})}{r_{k-1}^T r_{k-1}}$$

where,

$$r_k = \nabla f_k$$