The conjugate gradient method

Nicholas Hu

The conjugate gradient method is an iterative method for solving the linear system $Ax = b$, where $A \in \mathbb{R}^{n \times n}$ is symmetric positive-definite.

Let $\langle x, y \rangle_A = \langle Ax, y \rangle$ be the inner product defined by A and $\|x\|_A = \sqrt{\langle x, x \rangle_A}$ be the induced norm. Given an initial guess $x^{(0)}$ for the solution x^*, the k^{th} iterate of the method is selected such that

$$x^{(k)} = \underset{x \in x^{(0)} + \mathcal{K}_k(A, r^{(0)})}{\arg\min} \|x^* - x\|_A \,,$$

where $\mathcal{K}_k(A, r^{(0)})$ is the Krylov subspace $\operatorname{span}\{A^j r^{(0)}\}_{j=0}^{k-1}$ and $r^{(0)} = b - Ax^{(0)}$. (In other words, the A-norm of the error is minimized over the k^{th} affine Krylov subspace generated by the initial residual and translated by the initial guess.) [1]

Let us abbreviate $\mathcal{K}_k(A, r^{(0)})$ as $\mathcal{K}_k$ and write $r^{(k)} = b - Ax^{(k)}$ for the residual of the k^{th} iterate. The iterate $x^{(k)}$ is therefore the A-orthogonal projection of x^* onto $x^{(0)} + \mathcal{K}_k$, defined by the Galerkin conditions $x^{(k)} - x^{(0)} \in \mathcal{K}_k$ and $x^* - x^{(k)} \perp_A \mathcal{K}_k$. In particular, the orthogonality condition implies that

$$\begin{aligned} r^{(k)} &\perp \mathcal{K}_k \,, \\ r^{(k)} &\perp_A \mathcal{K}_{k-1} \,, \end{aligned}$$

where the latter holds since $A\mathcal{K}_{k-1} \subseteq \mathcal{K}_k$.

If $\{p^{(j)}\}_{j<k}$ is a basis for $\mathcal{K}_k$ and $P_k = \begin{bmatrix} p^{(0)} & \cdots & p^{(k-1)} \end{bmatrix}$, we have

$$x^{(k)} = x^{(0)} + P_k (P_k^\top A P_k)^{-1} P_k^\top r^{(0)}.$$

If, in addition, the $p^{(j)}$ are chosen to be A-orthogonal, the matrix $P_k^\top A P_k$ (the Gram matrix of the $p^{(j)}$ with respect to the A-inner product) becomes a diagonal matrix D_k, and a particularly simple recurrence can be found for the $x^{(k)}$. Namely,

$$\begin{aligned} x^{(k+1)} &= x^{(0)} + P_{k+1} D_{k+1}^{-1} P_{k+1}^\top r^{(0)} \\ &= x^{(0)} + \begin{bmatrix} P_k & p^{(k)} \end{bmatrix} \begin{bmatrix} D_k & \\ & \langle p^{(k)}, p^{(k)} \rangle_A \end{bmatrix}^{-1} \begin{bmatrix} P_k^\top \\ (p^{(k)})^\top \end{bmatrix} r^{(0)} \\ &= x^{(0)} + P_k D_k^{-1} P_k^\top r^{(0)} + p^{(k)} \langle p^{(k)}, p^{(k)} \rangle_A^{-1} (p^{(k)})^\top r^{(0)} \\ &= x^{(k)} + \alpha_k p^{(k)}, \end{aligned} \tag{X}$$

where

$$\alpha_k = \frac{\langle r^{(0)}, p^{(k)} \rangle}{\langle p^{(k)}, p^{(k)} \rangle_A} \,. \tag{A}$$

This also implies that

$$r^{(k+1)} = r^{(k)} - \alpha_k A p^{(k)}. \tag{R}$$

In order to generate such a basis, we employ Gram–Schmidt orthogonalization (with respect to the A-inner product). However, instead of orthogonalizing the vectors $A^j r^{(0)}$, we will orthogonalize the successive residuals $r^{(j)}$. We claim that the resulting vectors $p^{(j)}$ will still constitute bases of the Krylov subspaces. [2]

More precisely, suppose that the solution has not been found by the end of the k^{th} iteration, in the sense that $r^{(j)} \neq 0$ for all $j \leq k$, and that $p^{(j)}$ was generated by orthogonalizing $r^{(j)}$ against $p^{(i)}$ for each $i < j \leq k$. We claim then that $\{r^{(j)}\}_{j \leq k}$ is a basis for $\mathcal{K}_{k+1}$ (and hence that $\{p^{(j)}\}_{j \leq k}$ is an A-orthogonal basis for $\mathcal{K}_{k+1}$).

Indeed, if $r^{(0)} \neq 0$, then $\{r^{(0)}\}$ is a basis for $\mathcal{K}_1 = \operatorname{span}\{r^{(0)}\}$. Now suppose that the claim holds for the k^{th} iteration and that its hypotheses are satisfied after the $(k+1)^{\text{th}}$ iteration. Then

$$\begin{aligned} \operatorname{span}\{r^{(j)}\}_{j \leq k+1} &= \operatorname{span}\{r^{(j)}\}_{j \leq k} + \operatorname{span}\{r^{(k+1)}\} \\ &= \mathcal{K}_{k+1} + \operatorname{span}\{r^{(k+1)}\} \end{aligned}$$

and $r^{(k+1)} \notin \mathcal{K}_{k+1}$ since $r^{(k+1)} \perp \mathcal{K}_{k+1}$ and $r^{(k+1)} \neq 0$. Hence $\dim(\operatorname{span}\{r^{(j)}\}_{j \leq k+1}) \geq \dim(\mathcal{K}_{k+1}) + 1$. On the other hand,

$$\begin{aligned} \mathcal{K}_{k+1} + \operatorname{span}\{r^{(k+1)}\} &= \mathcal{K}_{k+1} + \operatorname{span}\{r^{(k)} - \alpha_k A p^{(k)}\} \\ &\subseteq \mathcal{K}_{k+1} + (\mathcal{K}_{k+1} + A\mathcal{K}_{k+1}) \\ &\subseteq \mathcal{K}_{k+2}, \end{aligned}$$

where $\dim(\mathcal{K}_{k+2}) \leq \dim(\mathcal{K}_{k+1}) + 1$, so all the subspaces (and dimensions) above must be equal.

An immediate consequence of this choice is that the residuals will be orthogonal: if (say) $i < j$, then $r^{(i)} \in \mathcal{K}_{i+1} \subseteq \mathcal{K}_j$, and we know that $r^{(j)} \perp \mathcal{K}_j$.

It remains to derive a recurrence for the $p^{(j)}$. In view of the fact that $r^{(k+1)} \perp_A \mathcal{K}_k = \operatorname{span}\{p^{(j)}\}_{j<k}$, we find that

$$\begin{aligned} p^{(k+1)} &= r^{(k+1)} - \sum_{j<k+1} (\operatorname{proj}^A_{p^{(j)}} r^{(k+1)}) p^{(j)} \\ &= r^{(k+1)} - (\operatorname{proj}^A_{p^{(k)}} r^{(k+1)}) p^{(k)} \\ &= r^{(k+1)} + \beta_k p^{(k)}, \end{aligned} \tag{P}$$

where

$$\beta_k = -\frac{\langle r^{(k+1)}, p^{(k)} \rangle_A}{\langle p^{(k)}, p^{(k)} \rangle_A}. \tag{B}$$

This completes the mathematical derivation of the method.

The formulas for the scalars α_k and β_k can be further simplified (computation-wise):

$$\begin{aligned}\alpha_k &= \frac{\langle r^{(0)}, p^{(k)}\rangle}{\langle p^{(k)}, p^{(k)}\rangle_A} \\ &= \frac{\langle r^{(k)}, p^{(k)}\rangle + \langle r^{(0)} - r^{(k)}, p^{(k)}\rangle}{\langle p^{(k)}, p^{(k)}\rangle_A} \\ &= \frac{\langle r^{(k)}, p^{(k)}\rangle + \langle x^{(k)} - x^{(0)}, p^{(k)}\rangle_A}{\langle p^{(k)}, p^{(k)}\rangle_A} \\ &= \frac{\langle r^{(k)}, p^{(k)}\rangle}{\langle p^{(k)}, p^{(k)}\rangle_A}\end{aligned}$$

since $x^{(k)} - x^{(0)} \in \mathcal{K}_k$ and $p^{(k)} \perp_A \operatorname{span}\{p^{(j)}\}_{j<k} = \mathcal{K}_k$ by definition. Moreover, $\langle r^{(k)}, p^{(k)}\rangle = \langle r^{(k)}, r^{(k)} + \beta_{k-1}p^{(k-1)}\rangle = \langle r^{(k)}, r^{(k)}\rangle$ as $r^{(k)} \perp \mathcal{K}_k$, so

$$\alpha_k = \frac{\langle r^{(k)}, r^{(k)}\rangle}{\langle p^{(k)}, p^{(k)}\rangle_A}\,. \tag{A}$$

Finally,

$$\begin{aligned}\beta_k &= -\frac{\langle r^{(k+1)}, p^{(k)}\rangle_A}{\langle p^{(k)}, p^{(k)}\rangle_A} \\ &= -\alpha_k \frac{\langle r^{(k+1)}, p^{(k)}\rangle_A}{\langle r^{(k)}, r^{(k)}\rangle} \\ &= \frac{\langle r^{(k+1)}, r^{(k+1)} - r^{(k)}\rangle}{\langle r^{(k)}, r^{(k)}\rangle} \\ &= \frac{\langle r^{(k+1)}, r^{(k+1)}\rangle}{\langle r^{(k)}, r^{(k)}\rangle}\,.\end{aligned} \tag{B}$$

In summary,

$$\begin{aligned} r^{(0)} &= b - Ax^{(0)} \\ p^{(0)} &= r^{(0)} \\ \\ \alpha_k &= \frac{\langle r^{(k)}, r^{(k)}\rangle}{\langle p^{(k)}, p^{(k)}\rangle_A} && \text{A} \\ x^{(k+1)} &= x^{(k)} + \alpha_k p^{(k)} && \text{X} \\ r^{(k+1)} &= r^{(k)} - \alpha_k A p^{(k)} && \text{R} \\ \beta_k &= \frac{\langle r^{(k+1)}, r^{(k+1)}\rangle}{\langle r^{(k)}, r^{(k)}\rangle} && \text{B} \\ p^{(k+1)} &= r^{(k+1)} + \beta_k p^{(k)} && \text{P} \end{aligned}$$

---

1. The choice of this minimization problem can be partially motivated as follows. In view of the fact that $x^* = x^{(0)} + A^{-1}r^{(0)}$ and that A^{-1} is a polynomial in A of degree at most $n-1$, in the k^{th} iteration of the method, we seek an approximation to the solution of the form $x^{(0)} + p_{k-1}(A)r^{(0)}$, where p_{k-1} is a polynomial of degree at most $k-1$. This guarantees that the A-norm of the error decreases monotonically and that the solution is found in at most n iterations (in exact arithmetic). Although the choice of the objective function is not canonical, it turns out that this choice leads to a particularly tractable method. ↩

2. The computation of the residual in each iteration furnishes a useful measure of progress towards the solution (namely, the norm of the residual) at the cost of one matrix-vector multiplication. At the same time, since $r^{(k+1)} = r^{(k)} - \alpha_k A p^{(k)}$, it is inductively plausible that the sets $\{r^{(j)}\}_{j<k}$ would constitute bases of successive Krylov subspaces and that the $p^{(j)}$ could be generated 'online' from the $r^{(j)}$, so to speak. Importantly, no other matrix-vector multiplications would be needed in each iteration. ↩