Notes for 2016-11-18

1 Conjugate gradients

We now turn to the method of conjugate gradients (CG), perhaps the best known of the Krylov subspace solvers. The CG iteration can be characterized as the iteration for a symmetric positive definite A that minimizes the energy

$$\phi(x) = \frac{1}{2} x^T A x - x^T b$$

over a Krylov subspace; as we have already seen,

$$\phi(x) + \frac{1}{2} b^T A^{-1} b = \frac{1}{2} \|x - A^{-1} b\|_A^2 = \frac{1}{2} \|Ax - b\|_{A^{-1}}^2,$$

so this minimization corresponds to minimizing the error in the A-norm or the residual in the A^{-1} norm. We also have seen the shape of the standard error analysis, which involves looking at a Chebyshev polynomial on an interval containing the spectrum. The iteration turns out to be forward unstable, so the behavior in floating point arithmetic is not the same as the behavior in theory; but this does not prevent the iteration from being highly effective, partly because we can write the iteration in a form that involves an explicit residual, and looking at a freshly-computed residual gives the method a self-correcting property.

Our goal for today is to look at the mechanics of the method.

1.1 CG via Lanczos

Last time, we discussed the Lanczos iteration, which produces the Lanczos decomposition

$$AQ_k = Q_{k+1} \bar{T}_k$$

via the iteration

$$\beta_k q_{k+1} = Aq_k - \alpha_k q_k - \beta_{k-1} q_{k-1}$$

where $\alpha_k = q_k^T A q_k$. One of the simplest derivations for the *conjugate gradient* (CG) method is in terms of the Lanczos decomposition.

In terms of the energy

$$\phi(x) = \frac{1}{2} x^T A x - x^T b,$$

the problem of finding the "best" (minimum energy) approximate solution in the space becomes

$$\text{minimize } \phi(Q_k y_k) = \frac{1}{2} y_k^T T_k y_k - y_k^T e_1 \|b\|,$$

which is solved by

$$T_k y_k = e_1 \|b\|.$$

Now let us suppress the indices for a moment and write $T = LU$ (which can be computed stably without pivoting, as T is SPD). Then we can write the approximate solution $\hat{x}$ as

$$\hat{x} = QU^{-1}L^{-1}e_1\|b\|,$$

which we will group as

$$\hat{x} = V\hat{y}, \quad VU = Q, \quad Ly = e_1\|b\|.$$

Solving the system for y by forward substitution yields

$$\begin{aligned} y_1 &= \|b\| \\ y_k &= -l_{k,k-1} y_{k-1}. \end{aligned}$$

Similarly, we can compute the columns of V by forward substitution:

$$\begin{aligned} v_1 &= q_1/u_{11} \\ v_k &= \frac{1}{u_{kk}} \left(q_k - v_{k-1} u_{k-1,k} \right). \end{aligned}$$

The advantage of this formulation is that if we extend the Krylov subspace, we simply extend the tridiagonal (and associated factorization), add another component to y, and bring in a new vector v — all without disturbing the computations done before. Hence, we have a sequence of coupled recurrences for the columns of Q and of V that allow us to incrementally update the solution at the cost of a matrix-vector multiply and a constant amount of vector arithmetic per step.

This is a useful approach, but it does not shed much insight into how the method could be extended to optimize more general objectives than quadratics. For that, we need the approach that gives the CG method its name.

1.2 Another approach to CG

An alternate approach to the conjugate gradient method does not directly invoke Lanczos, but instead relies on properties that must be satisfied at each step by the residual $r_m = b - Ax_m$ and the update $d_m = x_{m+1} - x_m$. We assume throughout that x_m is drawn from $\mathcal{K}_m(A, b)$, which implies that $r_m \in \mathcal{K}_{m+1}(A, b)$ and $d_m \in \mathcal{K}_{m+1}(A, b)$.

First, note that $r_m \perp \mathcal{K}_m(A, b)$ and $d_m \perp_A \mathcal{K}_m(A, b)$.[1] The former statement comes from the Galerkin criterion in the previous section. The latter statement comes from recognizing that $r_{m+1} = Ad_m + r_m \perp \mathcal{K}_m(A, b)$; with Galerkin condition $r_m \perp \mathcal{K}_m(A, b)$, this means $Ad_m \perp \mathcal{K}_m(A, b)$. Together, these statements give us r_m and d_m to within a scalar factor, since there is only one direction in $\mathcal{K}_{m+1}(A, b)$ that is orthogonal to all of $\mathcal{K}_m(A, b)$, and similarly there is only one direction that is A-orthogonal. This suggests the following idea to generate the sequence of approximate solutions x_k:

1. Find a direction $p_{k-1} \in \mathcal{K}_k(A, b)$ that is A-orthogonal to $\mathcal{K}_{k-1}(A, b)$.

2. Compute $x_k = x_{k-1} + \alpha_k p_{k-1}$ so that

   $$r_k = r_{k-1} - \alpha_k A p_{k-1} \perp r_{k-1},$$

   i.e. set $\alpha_k = (r_{k-1}^T r_{k-1})/(p_{k-1}^T A p_{k-1})$. Orthogonality to the rest of $\mathcal{K}_k(A, b)$ follows automatically from the construction.

3. Take $r_k \in \mathcal{K}_{k+1}(A, b)$ and A-orthogonalize against everything in $\mathcal{K}_k(A, b)$ to generate the new direction p_k. As with the Lanczos procedure, the real magic in this idea is that we have to do very little work to generate p_k from r_k. Note that for any $j < k-1$, we have $p_j^T A r_k = (Ap_j)^T r_k = 0$, because $Ap_j \in \mathcal{K}_{j+2}(A, b) \subset \mathcal{K}_k(A, b)$ is automatically orthogonal to r_k. Therefore, we really only need to choose

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

   such that $p_{k-1}^T A p_k$, i.e. $\beta_k = -(p_{k-1}^T A r_k)/(p_{k-1}^T A p_{k-1})$. Note, though, that $Ap_{k-1} = -(r_k - r_{k-1})/\alpha_k$; with a little algebra, we find

   $$\beta_k = -\frac{r_k^T A p_k}{p_{k-1}^T A p_{k-1}} = \frac{(r_k^T r_k)/\alpha_k}{r_{k-1}^T r_{k-1}/\alpha_k} = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}.$$

[1] $u \perp_A v$ means u and v are orthogonal in the A-induced inner product, i.e. $u^T A v = 0$.

Putting everything together, we have the following coupled recurrences for the solutions x_k, residuals r_k, and search directions p_k:

$$\begin{aligned}
\alpha_k &= (r_{k-1}^T r_{k-1})/(p_{k-1}^T A p_{k-1}) \\
x_k &= x_{k-1} + \alpha_k p_{k-1} \\
r_k &= r_{k-1} - \alpha_k A p_{k-1} \\
\beta_k &= (r_k^T r_k)/(r_{k-1}^T r_{k-1}) \\
p_k &= r_k + \beta_k p_{k-1}.
\end{aligned}$$

The sequences r_k and p_k respectively form orthogonal and A-orthogonal bases for the nested Krylov subspaces generated by A and b.

1.3 Preconditioning

What happens if we want to compute not on the space $\mathcal{K}_k(A, b)$, but the preconditioned space $\mathcal{K}_k(M^{-1}A, M^{-1}b)$ where M is some symmetric positive definite matrix? Unfortunately, we cannot apply CG directly to a system involving $M^{-1}A$, since even if M and A are SPD, the product will generally not be. On the other hand, we can certainly work with the related system

$$(M^{-1/2}AM^{-1/2})(M^{1/2}x) = M^{-1/2}b.$$

This is a symmetric positive definite system, and the eigenvalues of $M^{-1/2}AM^{-1/2}$ are the same as the generalized eigenvalues of the pencil (A, M). Moreover, we can work with this system *implicitly* without ever having to form the awkward square root.

Define $\bar{p}_k = M^{-1/2}p_k$ and $\bar{r}_k = M^{1/2}r_k$; then CG iteration on the related system can be rephrased as

$$\begin{aligned}
\alpha_k &= (\bar{r}_{k-1}^T M^{-1} \bar{r}_{k-1})/(\bar{p}_{k-1}^T A \bar{p}_{k-1}) \\
x_k &= x_{k-1} + \alpha_k \bar{p}_{k-1} \\
\bar{r}_k &= \bar{r}_{k-1} - \alpha_k A \bar{p}_{k-1} \\
\beta_k &= (\bar{r}_k^T M^{-1} \bar{r}_k)/(\bar{r}_{k-1}^T M^{-1} \bar{r}_{k-1}) \\
\bar{p}_k &= M^{-1}\bar{r}_k + \beta_k \bar{p}_{k-1}.
\end{aligned}$$

Because expressions involving M^{-1} and the residual appear throughout, we

introduce a new variable $z_k = M^{-1} r_k$, leading to

$$\begin{aligned}
\alpha_k &= (\bar{r}_{k-1}^T z_{k-1})/(\bar{p}_{k-1}^T A \bar{p}_{k-1}) \\
x_k &= x_{k-1} + \alpha_k \bar{p}_{k-1} \\
\bar{r}_k &= \bar{r}_{k-1} - \alpha_k A \bar{p}_{k-1} \\
M z_k &= r_k \\
\beta_k &= (\bar{r}_k^T z_k)/(\bar{r}_{k-1}^T z_{k-1}) \\
\bar{p}_k &= z_k + \beta_k \bar{p}_{k-1}.
\end{aligned}$$

Another way of thinking about the preconditioned CG iteration is that it is ordinary CG, whether thought of in terms of conjugate directions or in terms of Lanczos, but with a different inner product: the M^{-1} inner product on residuals, or the M inner product in the Lanczos procedure.

1.4 Nonlinear CG

One of the advantages of the interpretation of CG in terms of search directions and residuals is that it generalizes beyond the case of quadratic optimization or linear system solving to more general optimization problems. To derive nonlinear CG, we generalize the quantities in the ordinary CG iteration in the following way:

- In ordinary CG, we let ϕ be a quadratic energy function. In nonlinear CG, ϕ is a more general (though ideally convex) objective function.
- In ordinary CG, we have $r_k = -\nabla\phi(x_k) = b - Ax_k$. In nonlinear CG, we take $r_k = -\nabla\phi(x_k)$, though the gradient expression will generally be more complicated.
- In ordinary CG, we choose a search direction $p_k = r_k + \beta_k p_{k-1}$ where $\beta_k = r_k^T r_k / r_{k-1}^T r_{k-1}$. In nonlinear CG, we may use the same formula (the *Fletcher-Reeves* formula), or we may choose any number of other formulas that are equivalent in the quadratic case but not in the more general case.
- In ordinary CG, once we choose a search direction p_{k-1}, we compute a step $x_k = x_{k-1} + \alpha_k p_{k-1}$. The α_k has the property

  $$\alpha_k = \operatorname{argmin}_\alpha \phi(x_k + \alpha p_{k-1})$$

  In nonlinear CG, we instead use a line search to choose the step size.

Like ordinary CG, nonlinear CG iterations can be preconditioned.

1.5 The many approaches to CG

The description I have given in these notes highlights (I hope) how orthogonality of the residuals and A-orthogonality of search directions follows naturally from the Galerkin condition, and how the rest of the CG iteration can be teased out of these orthogonality relations. However, this is far from the only way to "derive" the method of conjugate gradients. The discussion given by Demmel and by Saad (in *Iterative Methods for Sparse Linear Systems*) highlights the Lanczos connection, and uses this connection to show the existence of A-orthogonal search directions. Golub and Van Loan show the Lanczos connection, but also show how conjugate gradients can be derived as a general-purpose minimization scheme applied to the quadratic function $\phi(x)$. Trefethen and Bau give the iteration without derivation first, and then gradually explain some of its properties. If you find these discussions confusing, or simply wish to read something amusing, I recommend Shewchuk's "Introduction to the Conjugate Gradient Method Without the Agonizing Pain".