June 6, 2014

Steepest descent method

- Local optimal direction

Consider $f'(\alpha) = p^T (A x^k - b) + \alpha\, p^T A p$

Then $f'(0) = p^T F'(x^k)$ $(F'(x) = A x - b)$

$= $ changes in F at x^k in the direction of p

Idea: make $f'(0)$ as negative as possible by varying p

Assume $||p|| = 1$. Then

$f'(0)$ max if $p = F'(x^k) / ||F'(x^k)||$ = steepest ascent

$f'(0)$ min if $p = - F'(x^k) / ||F'(x^k)||$ = steepest descent

$= r^k / ||r^k||$ $(F'(x) = -r)$

Steepest descent method:

$x^{k+1} = x^k + \alpha^k r^k$ α^k = step length

The optimal $\alpha^k = (r^k)^T(r^k) / (r^k)^T A r^k$ $(\alpha = p^T r^k / p^T A p)$

Also $$\begin{aligned} r^{k+1} &= b - A x^{k+1} \\ &= b - A (x^k + \alpha_k r^k) \\ &= b - A x^k - \alpha_k A r^k \\ &= r^k - \alpha_k A r^k \end{aligned}$$

Algorithm

Given x^0, compute $r^0 = b - A x^0$

for $k = 0, 1, 2, \ldots$

$\alpha_k = (r^k)^T(r^k) / (r^k)^T A r^k$

$x^{k+1} = x^k + \alpha_k r^k$

$r^{k+1} = r^k - \alpha_k A r^k$

end

Notes

1) Only 1 matrix-vector product ($A r^k$) per iteration

2) "Nonlinear" iterative method:

$$x^{k+1} = x^k + \alpha_k (b - A x^k)$$

i.e. $M = M^k = 1/\alpha_k I$

Method of conjugate directions

- Each new directions is "A-orthogonal" to previous search directions.

Def: Suppose A is SPD. The A-inner product is defined as:

$$(p, q)_A = p^T A q$$

The A-norm is defined as:

$$\|p\|_A = \sqrt{(p, p)_A}$$

Gram-Schmidt process

- Construct a set of orthogonal vectors.

- Suppose the previous search directions $p^0, p^1, \ldots, p^{k-1}$ are A-orth. Given the current r^k, construct p^k.

Let

$$p^k = r^k + \sum_{i=0}^{k-1} \beta_i p^i$$

$$(p^k, p^j)_A = 0 \quad \Rightarrow \quad (r^k, p^j)_A + (\sum_{i=0}^{k-1} \beta_i p^i, p^j)_A = 0$$

$$(r^k, p^j)_A + \beta_j (p^j, p^j)_A = 0$$

$$\beta_j = -\frac{(r^k, p^j)_A}{(p^j, p^j)_A}$$

Conjugate gradient method

Construct a set of A-orth search vectors $\{ p^k \}$ by the residual vectors $\{ r^k \}$.

i.e. $$p^k = r^k + \sum_{i=0}^{k-1} \beta_i p^i = r^k - \sum_{i=0}^{k-1} \frac{(r^k, p^i)_A}{(p^i, p^i)_A} p^i$$

CG Algorithm 1

x^0 = initial guess; $r^0 = b - A x^0$

for $k = 0, 1, 2, . . . , n-1$

Compute p^k as above.

$x^{k+1} = x^k + \alpha^k p^k$

$r^{k+1} = r^k - \alpha^k A p^k$

end

Notes

1) $\alpha^k = (r^k, p^k) / (p^k, p^k)_A$
2) $r^{k+1} = b - A x^{k+1}$

Useful facts

- span $\{ p^0, \ldots, p^{k-1} \}$ = span $\{ r^0, \ldots, r^{k-1} \}$

  = span $\{ r^0, Ar^0, \ldots, A^{k-1} r^0 \}$

  = $\mathcal{K}_k (A, r^0)$

  = k-dim Krylov subspace

- $r^k \perp$ span $\{ r^0, \ldots, r^{k-1} \}$ i.e. $(r^k, r^j) = 0$ $\quad j = 0, 1, \ldots, k-1$

  Hence $r^k \perp$ span $\{ p^0, \ldots, p^{k-1} \}$.

- $(r^k, p^k) = (r^k, r^k)$

  Pf: $(r^k, p^k) = (r^k, r^k + \text{sum } \beta_i p^i) = (r^k, r^k)$

- $(r^k, p^i)_A = 0 \quad i = 0, 1, \ldots, k-2$.

  Pf: $p^i \in \text{span}\{p^0, \ldots, p^i\} = \text{span}\{r^0, Ar^0, \ldots, A^i r^0\}$

  $$\Rightarrow \quad Ap^i \in \text{span}\{Ar^0, A^2 r^0, \ldots, A^{i+1} r^0\}$$

  $$\subset \text{span}\{r^0, Ar^0, \ldots, A^{i+1} r^0\}$$

  $$= \text{span}\{r^0, r^1, \ldots, r^{i+1}\}$$

  $$\text{But} \quad r^k \perp \text{span}\{r^0, r^1, \ldots, r^{i+1}\} \quad \forall\, i+1 \le k-1$$

  $$\therefore \quad (r^k, p^i)_A = (r^k, Ap^i) = 0 \quad \forall\, i \le k-2$$

- $$p^k = r^k + \sum_{i=0}^{k-1} \beta_i p^i = r^k - \sum_{i=0}^{k-1} \frac{(r^k, p^i)_A}{(p^i, p^i)_A} p^i$$

  $$= r^k - \frac{(r^k, p^{k-1})_A}{(p^{k-1}, p^{k-1})_A} p^{k-1}$$

- $r^k = r^{k-1} - \alpha_{k-1} A p^{k-1}$

  $(r^k, r^k) = \overset{0}{(r^k, r^{k-1})} - \alpha_{k-1} (r^k, A p^{k-1})$

  Thus $(r^k, p^{k-1})_A = (r^k, A p^{k-1}) = -1/\alpha_{k-1} (r^k, r^k)$

- $0 = (r^k, p^{k-1}) = (r^{k-1}, p^{k-1}) - \alpha_{k-1} (Ap^{k-1}, p^{k-1})$

  $\phantom{0 = (r^k, p^{k-1})} = (r^{k-1}, r^{k-1}) - \alpha_{k-1} (p^{k-1}, p^{k-1})_A$

Thus $(p^{k-1}, p^{k-1})_A = 1/\alpha_{k-1} (r^{k-1}, r^{k-1})$

$$\Rightarrow \quad \beta_{k-1} = -\frac{(r^k, p^{k-1})_A}{(p^{k-1}, p^{k-1})_A} = -\left(\frac{-1}{\alpha_{k-1}}(r^k, r^k)\right)\frac{\alpha_{k-1}}{(r^{k-1}, r^{k-1})} = \frac{(r^k, r^k)}{(r^{k-1}, r^{k-1})}$$

<u>Conjugate gradient (CG) algorithm</u>

x^0 = initial guess; $r^0 = b - A x^0$

for $k = 0, 1, \ldots, n-1$

$\quad \beta_{k-1} = (r^k, r^k) / (r^{k-1}, r^{k-1}) \qquad\qquad (\beta_{-1} = 0)$

$\quad p^k = r^k + \beta_{k-1} p^{k-1}$

$\quad \alpha_k = (r^k, r^k) / (p^k, Ap^k)$

$\quad x^{k+1} = x^k + \alpha_k p^k$

$\quad r^{k+1} = r^k - \alpha_k Ap^k$

end

<u>Notes</u>

1) Only 1 matrix-vector multiply; 2 inner-products.
2) At most n A-orth vectors in R^n. Terminate at most n steps $\rightarrow$ exact solution.