CS5321 Numerical Optimization

05 Conjugate Gradient Methods

Conjugate gradient methods

- For convex quadratic problems,
  - the steepest descent method is slow in convergence.
  - the Newton's method is expensive in solving $Ax=b$.
  - the conjugate gradient method solves $Ax=b$ iteratively.
- Outline
  - Conjugate directions
  - Linear conjugate gradient method
  - Nonlinear conjugate gradient method

Quadratic optimization problem

- Consider the quadratic optimization problem

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

  - A is symmetric positive definite
  - The optimal solution is at $\nabla f(x) = 0$

$$\nabla(\frac{1}{2}x^T A x - b^T x) = Ax - b = 0$$

  - Define $r(x) = \nabla f(x) = Ax - b$ (the *residual*).
- Solve $Ax=b$ without inverting A. (Iterative method)

Steepest descent+line search

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

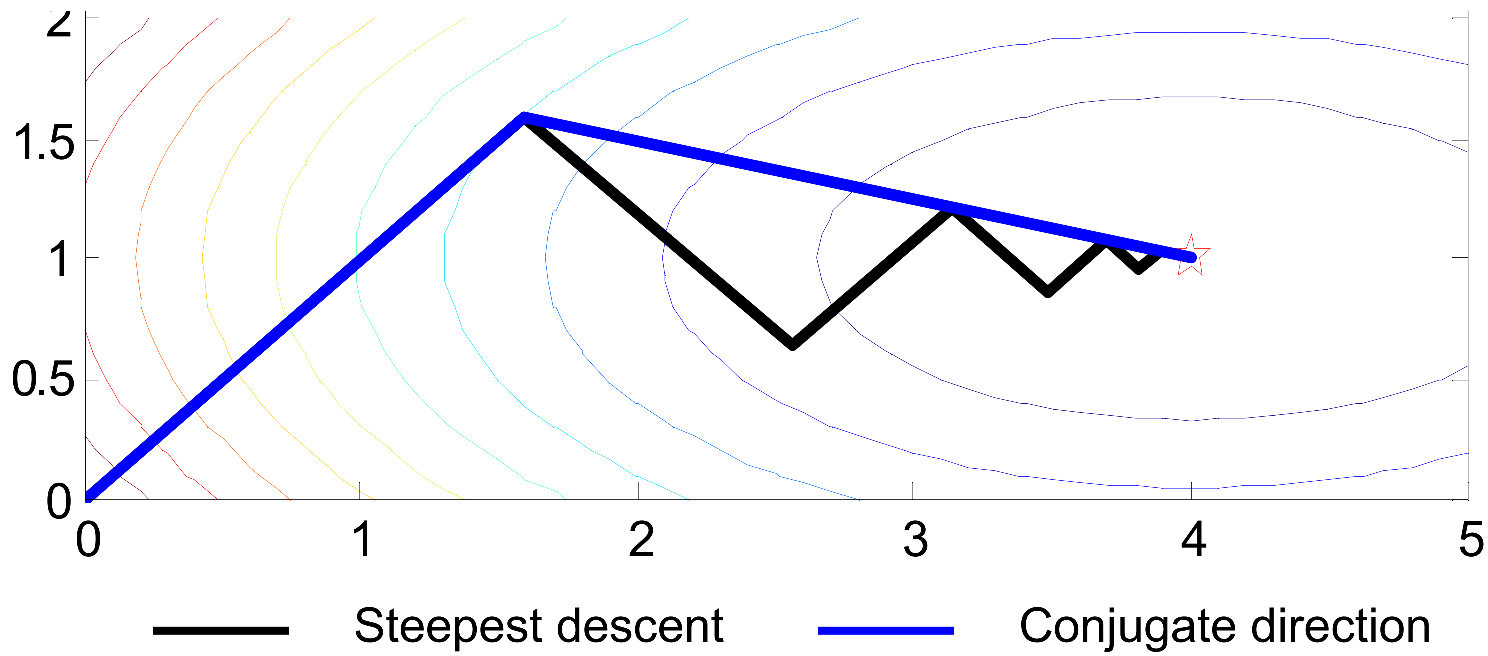
1. Given an initial guess x_0.
2. The search direction: $p_k = -\nabla f_k = -r_k = b - Ax_k$
3. The optimal step length: $\min_{\alpha} f(x_k + \alpha_k p_k)$
   - The optimal solution is $\alpha_k = -\frac{r_k^T r_k}{p_k^T A p_k}$
4. Update $x_{k+1} = x_k + \alpha_k p_k$. Goto 2.

Conjugate direction

- For a symmetric positive definite matrix A, one can define A-inner-product as $\langle x, y\rangle_A = x^T A y$.
  - A-norm is defined as $\|x\|_A = \sqrt{x^T A x}$
- Two vectors x and y are A-conjugate for a symmetric positive definite matrix A if $x^T A y = 0$
  - x and y are orthogonal under A-inner-product.
- The conjugate directions are a set of search directions $\{p_0, p_1, p_2, \ldots\}$, such that $p_i A p_j = 0$ for any $i \neq j$.

Example

$$A = \begin{pmatrix} 1/4 & 0 \\ 0 & 1 \end{pmatrix}, b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, x_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$



Conjugate gradient

- A better result can be obtained if the current search direction combines the previous one

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

- Let p_{k+1} be A-conjugate to p_k. ($p_{k+1}^T A p_k = 0$)

$$p_{k+1}^T A p_k = -r_k^T A p_k + \beta_k p_k^T A p_k = 0$$

$$\beta_k = \frac{p_k^T A r_k}{p_k^T A p_k} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

The linear CG algorithm

- With some linear algebra, the algorithm can be simplified as
    1. Given x_0, $r_0 = Ax_0 - b$, $p_0 = -r_0$
    2. For $k = 0,1,2,\ldots$ until $\|r_k\| = 0$

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

Properties of linear CG

- One matrix-vector multiplication per iteration.
- Only four vectors are required. (x_k, r_k, p_k, Ap_k)
  - Matrix A can be stored implicitly
- The CG guarantees convergence in r iterations, where r is the number of distinct eigenvalues of A
- If A has eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$,

$$\|x_{k+1} - x^*\|_A^2 \leq \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1} \right)^2 \|x_0 - x^*\|_A^2$$

CG for nonlinear optimization

The Fletcher-Reeves method

1. Given x_0. Set $p_0 = -\nabla f_0$,
2. For $k = 0,1,\ldots,$ until $\nabla f_0 = 0$
   - Compute optimal step length α_k and set $x_{k+1} = x_k + \alpha_k p_k$
   - Evaluate ∇f_{k+1}

$$\beta_{k+1} = \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}$$

$$p_{k+1} = -\nabla f_{k+1} + \beta_{k+1} p_k$$

Other choices of β

- Polak-Ribiere: $\beta_{k+1} = \dfrac{\nabla f_{k+1}^T (\nabla f_{k+1} - f_k)}{\nabla f_k^T \nabla f_k}$
- Hestens-Siefel: $\beta_{k+1} = \dfrac{\nabla f_{k+1}^T (\nabla f_{k+1} - f_k)}{(\nabla f_{k+1} - \nabla f_k)^T p_k}$
- Y.Dai and Y.Yuan $\beta_{k+1} = \dfrac{\|\nabla f_{k+1}\|^2}{(\nabla f_{k+1} - \nabla f_k)^T p_k}$
  - (1999)
- WW.Hager and H.Zhang
  - (2005)

$$\beta_{k+1} = \left(y_k - 2 p_k \frac{\|y_k\|^2}{y_k^T p_k} \right)^T \frac{\nabla f_{k+1}}{y_k^T p_k}$$