3. Conjugate gradient method

- conjugate gradient method for linear equations
- convergence analysis
- conjugate gradient method as iterative method
- nonlinear conjugate gradient method

Unconstrained quadratic minimization

$$\text{minimize} \quad f(x) = \frac{1}{2}x^T Ax - b^T x$$

with $A \in \mathbf{S}^n_{++}$

- equivalent to solving $Ax = b$
- residual $r = b - Ax$ is negative gradient at x: $r = -\nabla f(x)$

conjugate gradient method

- invented by Hestenes and Stiefel around 1951
- the most widely used iterative method for solving $Ax = b$, with $A \succ 0$
- can be extended to non-quadratic unconstrained minimization

Krylov subspaces

definition: a sequence of nested subspaces $(\mathcal{K}_0 \subseteq \mathcal{K}_1 \subseteq \mathcal{K}_2 \subseteq \cdots)$

$$\mathcal{K}_0 = \{0\}, \qquad \mathcal{K}_k = \mathrm{span}\{b, Ab, \ldots, A^{k-1}b\} \quad \text{for } k \geq 1$$

if $\mathcal{K}_{k+1} = \mathcal{K}_k$, then $\mathcal{K}_i = \mathcal{K}_k$ for all $i \geq k$

key property: $A^{-1}b \in \mathcal{K}_n$ (even when $\mathcal{K}_n \neq \mathbf{R}^n$)

Cayley-Hamilton theorem: $p(A) = A^n + a_1 A^{n-1} + \cdots + a_n I = 0$ where

$$p(\lambda) = \det(\lambda I - A) = \lambda^n + a_1 \lambda^{n-1} + \cdots + a_{n-1}\lambda + a_n$$

therefore $A^{-1}b = -\dfrac{1}{a_n}\left(A^{n-1}b + a_1 A^{n-2}b + \cdots + a_{n-1}b\right)$

Krylov sequence

CG algorithm is a recursive method for computing the *Krylov sequence*

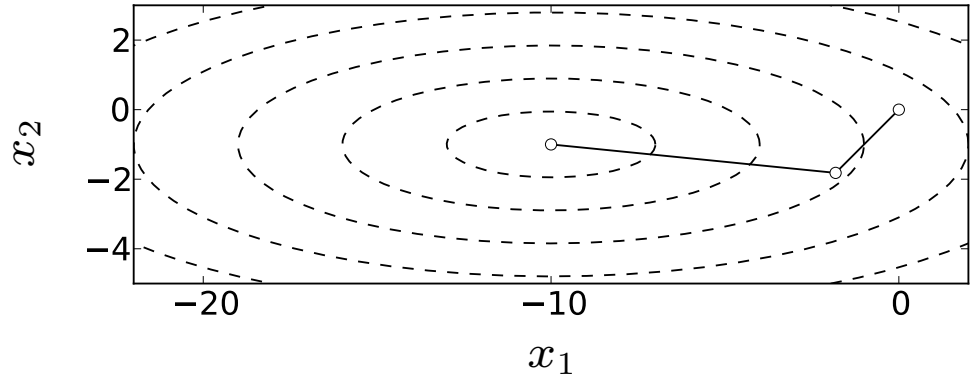
$$x^{(k)} = \operatorname*{argmin}_{x \in \mathcal{K}_k} f(x), \quad k \geq 0$$

- from previous page, $x^{(n)} = A^{-1}b$
- we will see there is a simple two-term recurrence

$$x^{(k+1)} = x^{(k)} - a_k \nabla f(x^{(k)}) + b_k(x^{(k)} - x^{(k-1)})$$

example

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}, \quad b = \begin{bmatrix} 10 \\ 10 \end{bmatrix}$$



Residuals of Krylov sequence

- optimality conditions in definition of Krylov sequence

$$x^{(k)} \in \mathcal{K}_k, \qquad \nabla f(x^{(k)}) = Ax^{(k)} - b \in \mathcal{K}_k^{\perp}$$

- hence, residuals $r_k = b - Ax^{(k)}$ satisfy

$$r_k \in \mathcal{K}_{k+1}, \qquad r_k \in \mathcal{K}_k^{\perp}$$

  (first property follows from $b \in \mathcal{K}_1$ and $x^{(k)} \in \mathcal{K}_k$)

(nonzero) residuals form an orthogonal basis for the Krylov subspaces

$$\mathcal{K}_k = \mathrm{span}\{r_0, r_1, \ldots, r_{k-1}\}, \qquad r_i^T r_j = 0 \quad (i \neq j)$$

Conjugate directions

the vectors $v_i = x^{(i)} - x^{(i-1)}$ satisfy

$$v_i^T A v_j = 0 \quad \text{for } i \neq j, \qquad v_i^T A v_i = v_i^T r_{i-1}$$

- directions are 'conjugate': orthogonal for inner product $\langle v, w \rangle = v^T A w$
- in particular, if $v_i \neq 0$ it is independent of $v_1, \ldots, v_{i-1}$
- $\mathcal{K}_k = \mathrm{span}\{v_1, v_2, \ldots, v_k\}$

(proofs on next page)

conjugate vectors: defined as $p_i = v_i/\alpha_i$, scaled so that $r_{i-1}^T p_i = \|r_{i-1}\|_2^2$

$$\alpha_i = \frac{v_i^T r_{i-1}}{\|r_{i-1}\|_2^2} = \frac{\|r_{i-1}\|_2^2}{p_i^T A p_i}$$

proof of properties on page 3-6 (assume $j < i$)

- $v_j^T A v_i = 0$ because $v_j = x^{(j)} - x^{(j-1)} \in \mathcal{K}_j \subseteq \mathcal{K}_{i-1}$ and

$$Av_i = A(x^{(i)} - x^{(i-1)}) = -r_i + r_{i-1} \in \mathcal{K}_{i-1}^{\perp}$$

- expression for $v_i^T A v_i$ follows from the fact that $t = 1$ minimizes

$$f(x^{(i-1)} + tv_i) = f(x^{(i-1)}) + \frac{1}{2}t^2 v_i^T A v_i - t v_i^T r_{i-1}$$

- second expression for α_i follows from

$$\alpha_i = \frac{v_i^T r_{i-1}}{\|r_{i-1}\|_2^2} = \frac{v_i^T A v_i}{\|r_{i-1}\|_2^2} = \alpha_i^2 \frac{p_i^T A p_i}{\|r_{i-1}\|_2^2}$$

Recursion for p_k

$\mathcal{K}_k = \mathrm{span}\{p_1, p_2, \ldots, p_{k-1}, r_{k-1}\}$, so we can express p_k as

$$p_1 = \delta r_0, \qquad p_k = \delta r_{k-1} + \beta p_{k-1} + \sum_{i=1}^{k-2} \gamma_i p_i \quad (k > 1)$$

- $\gamma_i = 0$: take inner products with Ap_i for $i \leq k-2$:

  $p_i^T A p_k = p_i^T A p_{k-1} = 0, \quad p_i^T A r_{k-1} = 0$ (because $Ap_i \in \mathcal{K}_{i+1} \subseteq \mathcal{K}_{k-1}$)

- $\delta = 1$: take inner product with r_{k-1} and use $r_{k-1}^T p_k = \|r_{k-1}\|_2^2$

- expression for β: take inner product with Ap_{k-1}

$$\beta = -\frac{p_{k-1}^T A r_{k-1}}{p_{k-1}^T A p_{k-1}}$$

Basic conjugate gradient algorithm

$x^{(0)} = 0$, $r_0 = b$

for $k = 1, 2, \ldots$

1. return $x^{(k-1)}$ if $\|r_{k-1}\|_2 \leq \epsilon \|b\|_2$
2. if $k = 1$, $p_k = r_0$; otherwise

$$p_k = r_{k-1} + \beta p_{k-1} \quad \text{where} \quad \beta = -\frac{p_{k-1}^T A r_{k-1}}{p_{k-1}^T A p_{k-1}}$$

3. compute

$$x^{(k)} = x^{(k-1)} + \alpha p_k \quad \text{where} \quad \alpha = \frac{\|r_{k-1}\|_2^2}{p_k^T A p_k}$$

and $r_k = b - Ax^{(k)}$

Improvements

step 3: compute residual recursively:

$$r_k = r_{k-1} - \alpha A p_k$$

step 2: simplify the expression for β by using

$$r_{k-1} = r_{k-2} - \frac{\|r_{k-2}\|_2^2}{p_{k-1}^T A p_{k-1}} A p_{k-1}$$

taking inner product with r_{k-1} gives

$$\beta = \frac{\|r_{k-1}\|_2^2}{\|r_{k-2}\|_2^2}$$

this reduces number of matrix multiplications to one per iteration

Conjugate gradient algorithm

$x^{(0)} = 0$, $r_0 = b$

for $k = 1, 2, \ldots$

1. return $x^{(k-1)}$ if $\|r_{k-1}\|_2 \leq \epsilon \|b\|_2$

2. if $k = 1$, $p_1 = r_0$; else

$$p_k = r_{k-1} + \frac{\|r_{k-1}\|_2^2}{\|r_{k-2}\|_2^2} p_{k-1}$$

3. compute

$$\alpha = \frac{\|r_{k-1}\|_2^2}{p_k^T A p_k}, \qquad x^{(k)} = x^{(k-1)} + \alpha p_k, \qquad r_k = r_{k-1} - \alpha A p_k$$

Outline

- conjugate gradient method for linear equations
- **convergence analysis**
- conjugate gradient method as iterative method
- nonlinear conjugate gradient method

Analysis of Krylov sequence

$$\text{minimize} \quad f(x) = \frac{1}{2}x^T Ax - b^T x$$

optimal value

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

suboptimality at x

$$f(x) - f^\star = \frac{1}{2}\|x - x^\star\|_A^2$$

relative error measure

$$\tau = \frac{f(x) - f^\star}{f(0) - f^\star} = \frac{\|x - x^\star\|_A^2}{\|x^\star\|_A^2}$$

here, $\|u\|_A = (u^T Au)^{1/2}$ is A-weighted norm

error after k steps in the Krylov sequence

- $x^{(k)} \in \mathcal{K}_k = \mathrm{span}\{b, Ab, A^2b, \ldots, A^{k-1}b\}$, so it can be expressed as

$$x^{(k)} = \sum_{i=1}^{k} \gamma_i A^{i-1} b = p(A)b$$

  where $p(s) = \sum_{i=1}^{k} \gamma_i s^{i-1}$, a polynomial of degree $k-1$ or less

- $x^{(k)}$ minimizes $f(x)$ over $\mathcal{K}_k$; hence

$$\begin{aligned} 2(f(x^{(k)}) - f^\star) &= \inf_{x \in \mathcal{K}_k} \|x - x^\star\|_A^2 \\ &= \inf_{\deg p < k} \left\|(p(A) - A^{-1})b\right\|_A^2 \end{aligned}$$

we now use the eigenvalue decomposition of A to bound this quantity

simplification using eigenvalue decomposition of A

$$A = Q\Lambda Q^T = \sum_{i=1}^n \lambda_i q_i q_i^T \qquad (Q^TQ = I, \quad \Lambda = \mathbf{diag}(\lambda_1, \ldots, \lambda_n))$$

with $\bar{b} = Q^Tb$, error expression simplifies to

$$\begin{aligned} \left\|(p(A) - A^{-1})b\right\|_A^2 &= \left\|(p(\Lambda) - \Lambda^{-1})\bar{b}\right\|_\Lambda^2 \\ &= \sum_{i=1}^n \frac{(\lambda_i p(\lambda_i) - 1)^2 \bar{b}_i^2}{\lambda_i} \end{aligned}$$

$$\begin{aligned} 2(f(x^{(k)}) - f^\star) &= \inf_{\deg p < k} \sum_{i=1}^n \frac{(\lambda_i p(\lambda_i) - 1)^2 \bar{b}_i^2}{\lambda_i} \\ &= \inf_{\deg q \leq k,\ q(0)=1} \sum_{i=1}^n \frac{q(\lambda_i)^2 \bar{b}_i^2}{\lambda_i} \end{aligned}$$

bounds on error

- absolute error

$$\begin{aligned} f(x^{(k)}) - f^\star &\leq \left(\sum_{i=1}^n \frac{\bar{b}_i^2}{2\lambda_i}\right) \inf_{\deg q \leq k,\ q(0)=1} \left(\max_{i=1,\ldots,n} q(\lambda_i)^2\right) \\ &= \frac{1}{2}\|x^\star\|_A^2 \inf_{\deg q \leq k,\ q(0)=1} \left(\max_{i=1,\ldots,n} q(\lambda_i)^2\right) \end{aligned}$$

  (equality follows from $\sum_i \bar{b}_i^2/\lambda_i = b^T A^{-1} b = \|x^\star\|_A^2$)

- relative error

$$\tau_k = \frac{\|x - x^\star\|_A^2}{\|x^\star\|_A^2} \leq \min_{\deg q \leq k,\ q(0)=1} \left(\max_{i=1,\ldots,n} q(\lambda_i)^2\right)$$

Convergence rate and spectrum of A

- if A has k distinct eigenvalues γ_1, . . . , γ_k, CG terminates in k steps

$$q(\lambda) = \frac{(-1)^k}{\gamma_1 \cdots \gamma_k}(\lambda - \gamma_1) \cdots (\lambda - \gamma_k)$$

  has degree k, $q(0) = 1$, $q(\lambda_i) = 0$ for all i; therefore $\tau_k = 0$

- if eigenvalues are clustered in k groups, then τ_k is small

  can find $q(\lambda)$ of degree k, with $q(0) = 1$, that is small on spectrum

- if $x^\star$ is a linear combination of k eigenvectors, termination in k steps

  take q of degree k with $q(\lambda_i) = 0$ where $\bar{b}_i \neq 0$; then

$$\sum_{i=1}^{n} \frac{q(\lambda_i)^2 \bar{b}_i^2}{\lambda_i} = 0$$

other bounds (without proof)

- in terms of condition number $\kappa = \lambda_{\max}/\lambda_{\min}$

$$\tau_k \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k$$

  derived by taking for q a Chebyshev polynomial on $[\lambda_{\min}, \lambda_{\max}]$

- in terms of sorted eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$

$$\tau_k \leq \left(\frac{\lambda_k - \lambda_n}{\lambda_k + \lambda_n}\right)^2$$

  derived by taking q with roots at $\lambda_1, \ldots, \lambda_{k-1}$ and $(\lambda_1 + \lambda_n)/2$

Outline

- conjugate gradient method for linear equations
- convergence analysis
- **conjugate gradient method as iterative method**
- nonlinear conjugate gradient method

Conjugate gradient method as iterative method

in exact arithmetic

- CG was originally proposed as a direct (non-iterative) method
- in theory, convergence in at most n steps

in practice

- due to rounding errors, CG method can take $\gg n$ steps (or fail)
- CG is now used as an iterative method
- with luck (good spectrum of A), good approximation in $\ll n$ steps
- attractive if matrix-vector products are inexpensive

Preconditioned conjugate gradient algorithm

preconditioner

- apply CG after linear change of coordinates $x = Ty$, $\det T \neq 0$
- use CG to solve $T^T A T y = T^T b$; then set $x^\star = T^{-1} y^\star$
- T or $M = TT^T$ is called *preconditioner*

implementation

- in naive implementation, each iteration requires multiplies by T and T^T (and A); also need to compute $x^\star = T^{-1} y^\star$ at end
- can re-arrange computation so each iteration requires one multiply by M (and A), and no final solve $x^\star = T^{-1} y^\star$

called *preconditioned conjugate gradient* (PCG) algorithm

Choice of preconditioner

- if spectrum of $T^T AT$ is clustered, PCG converges fast
- extreme case: $M = A^{-1}$
- trade-off between enhanced convergence, cost of multiplying with M

examples

- diagonal $M = \mathbf{diag}(1/A_{11}, \ldots, 1/A_{nn})$
- incomplete or approximate Cholesky factorization

best preconditioners are application-dependent

Outline

- conjugate gradient method for linear equations
- convergence analysis
- conjugate gradient method as iterative method
- **nonlinear conjugate gradient method**

Applications in optimization

nonlinear conjugate gradient methods

- extend linear CG method to nonquadratic functions
- local convergence similar to linear CG
- limited global convergence theory

inexact and truncated Newton methods

- use conjugate gradient method to compute (approximate) Newton step
- less reliable than exact Newton methods, but handle very large problems

Nonlinear conjugate gradient

minimize $f(x)$

(f convex and differentiable)

modifications needed to extend linear CG algorithm of page 3-11

- replace $r_k = b - Ax^{(k)}$ with $-\nabla f(x^{(k)})$
- determine α by line search

Fletcher-Reeves CG algorithm

CG algorithm of page 3-11 modified to minimize non-quadratic convex f

given $x^{(0)}$

for $k = 1, 2, \ldots$

1. return $x^{(k-1)}$ if $\|\nabla f(x^{(k-1)})\|_2 \leq \epsilon$
2. if $k = 1$, $p_1 = -\nabla f(x^{(0)})$; else

$$p_k = -\nabla f(x^{(k-1)}) + \beta p_{k-1} \quad \text{where} \quad \beta = \frac{\|\nabla f(x^{(k-1)})\|_2^2}{\|\nabla f(x^{(k-2)})\|_2^2}$$

3. update $x^{(k)} = x^{(k-1)} + \alpha p_k$ where $\alpha = \operatorname{argmin}_t f(x^{(k-1)} + t p_k)$

some observations

- first iteration is a gradient step; practical implementations restart the algorithm by taking a gradient step, for example, every n iterations
- update is gradient step with momentum term

$$x^{(k)} = x^{(k-1)} - \alpha_k \nabla f(x^{(k-1)}) + \beta_k (x^{(k-1)} - x^{(k-2)})$$

- with exact line search, reduces to linear CG for quadratic f

line search

- exact line search in step 3 implies $\nabla f(x^{(k)})^T p_k = 0$
- therefore in step 2, p_k is a descent direction at $x^{(k-1)}$:

$$\nabla f(x^{(k-1)})^T p_k = -\|\nabla f(x^{(k-1)})\|_2^2 < 0$$

Variations

Polak-Ribière: in step 2, compute β from

$$\beta = \frac{\nabla f(x^{(k-1)})^T(\nabla f(x^{(k-1)}) - \nabla f(x^{(k-2)}))}{\|\nabla f(x^{(k-2)})\|_2^2}$$

Hestenes-Stiefel

$$\beta = \frac{\nabla f(x^{(k-1)})^T(\nabla f(x^{(k-1)}) - \nabla f(x^{(k-2)}))}{p_{k-1}^T(\nabla f(x^{(k-1)}) - \nabla f(x^{(k-2)}))}$$

formulas are equivalent for quadratic f and exact line search

Interpretation as restarted BFGS method

BFGS update (page 2-5) with $H_{k-1} = I$:

$$H_k^{-1} = I + (1 + \frac{y^T y}{s^T y})\frac{ss^T}{y^T s} - \frac{ys^T + sy^T}{y^T s}$$

where $y = \nabla f(x^{(k)}) - \nabla f(x^{(k-1)})$, $s = x^{(k)} - x^{(k-1)}$

- $\nabla f(x^{(k)})^T s = 0$ if $x^{(k)}$ is determined by exact line search
- quasi-Newton step in iteration k is

$$-H_k^{-1}\nabla f(x^{(k)}) = -\nabla f(x^{(k)}) + \frac{y^T \nabla f(x^{(k)})}{y^T s} s$$

  this is the Hestenes-Stiefel update

nonlinear CG can be interpreted as L-BFGS with $m = 1$

References

- G. H. Golub and C. F. Van Loan, *Matrix Computations* (1996), chap. 10
- J. Nocedal and S. J. Wright, *Numerical Optimization* (2006), chap. 5
- S. Boyd, lecture notes for EE364b, Convex Optimization II