Section 5.1 – The Linear Conjugate Gradient Method

AMS 230 Numerical Optimization

Qi Gong
Dept. of Applied Math & Stats.
Baskin School of Engineering
University of California
Santa Cruz, CA

Linear Conjugate Gradient Method – Introduction

- Conjugate gradient methods is one of the most useful techniques for solving large linear systems of equations.

$$\boxed{\begin{array}{l}\text{Solve } Ax = b \\ \text{where } A = A^T \text{ is p.d.}\end{array}} \quad \Longleftrightarrow \quad \boxed{\begin{array}{l}\min\limits_{x \in R^n} \; f(x) = \frac{1}{2} x^T A x - b^T x \\ \text{where } A = A^T \text{ is positive definite.}\end{array}}$$

  - When dimension of x is not too large, linear systems can be solved by direct factorizations (LU, QR, Cholesky, etc.)
  - If the dimension of the problem is large, linear systems are often solved iteratively based on numerical optimization algorithms.
  - The steepest descent method is an iterative method for linear systems; but it's too slow.
  - Conjugate gradient methods can achieve fast convergence rate (in theory, CG converges in at most n number of steps) with less computational cost than direct factorizations.

- Conjugate gradient methods can be adapted to solve nonlinear optimization problems.

Linear CG – a Simple Example

Suppose A is a simple 2D diagonal pd. matrix

$$A = \begin{bmatrix} a_{11} & 0 \\ 0 & a_{22} \end{bmatrix}, \quad a_{ii} > 0$$

$$\min_{x \in R^n} \; f(x) = \frac{1}{2} x^T A x - b^T x$$

where $A = A^T$ is positive definite.

$$\min f(x_1, x_2) = \left(\frac{1}{2} a_{11} x_1^2 - b_1 x_1\right) + \left(\frac{1}{2} a_{22} x_2^2 - b_2 x_2\right)$$

$\left(\frac{1}{2} a_{11} x_1^2 - b_1 x_1\right)$ has a minimum at $\frac{b_1}{a_{11}}$

$\left(\frac{1}{2} a_{22} x_2^2 - b_2 x_2\right)$ has a minimum at $\frac{b_2}{a_{22}}$

$$\Longrightarrow \; x^* = \begin{bmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \end{bmatrix}$$

Linear CG – a Simple Example

Suppose A is a simple 2D diagonal pd. matrix

$$A = \begin{bmatrix} a_{11} & 0 \\ 0 & a_{22} \end{bmatrix}, \quad a_{ii} > 0$$

$$\min_{x \in R^n} f(x) = \frac{1}{2} x^T A x - b^T x$$
where $A = A^T$ is positive definite.

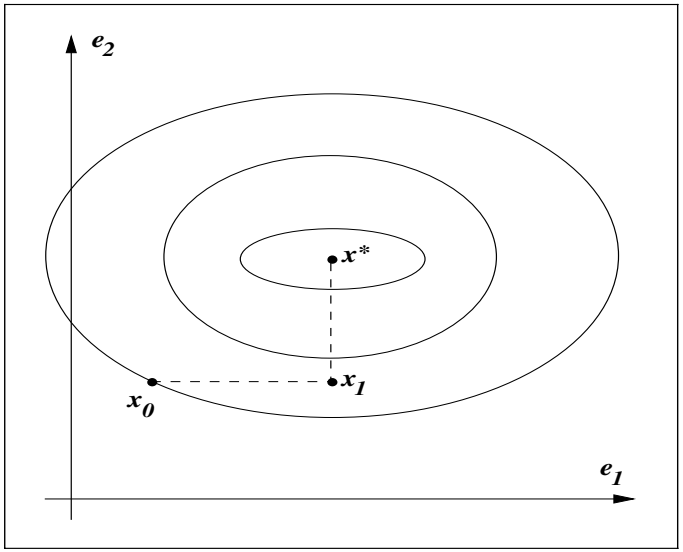
$$\min f(x_1, x_2) = \left(\frac{1}{2} a_{11} x_1^2 - b_1 x_1\right) + \left(\frac{1}{2} a_{22} x_2^2 - b_2 x_2\right)$$

has a minimum at $\frac{b_1}{a_{11}}$

has a minimum at $\frac{b_2}{a_{22}}$

$$\Longrightarrow x^* = \begin{bmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \end{bmatrix}$$

- Split a 2D optimization into 2 independent 1D optimizations.
- Global min can be found by performing two steps of line search along coordinate directions e_1 and e_2.



Linear CG – Motivation

Consider a general non-diagonal, symmetric and pd. matrix A.

$$\min_{x \in R^n} \; f(x) = \frac{1}{2} x^T A x - b^T x$$
$$\text{where } A = A^T \text{ is positive definite.}$$

Let $p_0, p_1, \ldots, p_{n-1}$ be a set of nonzero vectors in R^n such that

$$p_i^T A p_j = 0, \qquad \forall i \neq j$$

← Existence of such vectors p_i will be shown later.

and define $S = [p_0, p_1, \ldots, p_{n-1}]$. ← S is always invertible.

$$S^T A S = \begin{bmatrix} * & & & \\ & * & & \\ & & \ddots & \\ & & & * \end{bmatrix}$$

diagonal matrix

Linear CG – Motivation

Consider a general non-diagonal, symmetric and pd. matrix A.

$$\min_{x \in R^n} \; f(x) = \frac{1}{2} x^T A x - b^T x$$
$$\text{where } A = A^T \text{ is positive definite.}$$

Let $p_0, p_1, \ldots, p_{n-1}$ be a set of nonzero vectors in R^n such that

$$p_i^T A p_j = 0, \qquad \forall i \neq j$$

Existence of such vectors p_i will be shown later.

and define $S = [p_0, p_1, \ldots, p_{n-1}]$.

S is always invertible.

$$S^T A S = \begin{bmatrix} * & & & \\ & * & & \\ & & \ddots & \\ & & & * \end{bmatrix}$$

diagonal matrix

Let $y = S^{-1}x$. Then

$$\hat{f}(y) \triangleq f(Sy) = \frac{1}{2}(Sy)^T A (Sy) - b^T (Sy)$$
$$= \frac{1}{2} y^T (S^T A S) y - (S^T b)^T y$$

Linear CG – Motivation

A conceptual algorithm:

1. Construct vectors $p_0, p_1, \ldots, p_{n-1}$ such that
$$p_i^T A p_j = 0, \qquad \forall i \neq j$$
2. Compute diagonal matrix $S^T AS$ and vector $S^T b$.
3. For all i=1,2,…,n, find scalar y_i^* to minimize
$$\frac{1}{2} y^T (S^T AS) y - (S^T b)^T y$$
along direction e_i

$\Longleftrightarrow$

In x-coordinate, it's equivalent to performing line search in the direction p_i

4. Transfer the solution back to x coordinate, i.e.,
$$x^* = Sy^*, \text{ where } y^* = [y_1^*, y_2^*, \ldots, y_n^*]^T.$$

- **Will converge in at most n number of steps!**
- **Remaining questions:**
  - — Existence of $p_0, p_1, \ldots, p_{n-1}$ with the required property.
  - — How to compute $p_0, p_1, \ldots, p_{n-1}$ efficiently.

Inner Product and Orthogonality

Inner product of two vectors v_1, v_2 in R^n, denoted as $\langle v_1, v_2 \rangle$, is a scalar-valued function, such that the following properties hold:

1. $\langle v, v \rangle \geq 0, \forall v \in R^n$; and $\langle v, v \rangle = 0$ if and only if $v = 0$;
2. $\langle v_1, v_2 \rangle = \langle v_2, v_1 \rangle$;
3. $\langle v_1 + v_2, v_3 \rangle = \langle v_1, v_3 \rangle + \langle v_2, v_3 \rangle$;
4. $\langle \alpha v_1, v_2 \rangle = \alpha \langle v_1, v_2 \rangle$, where α is a scalar.

Examples: $\langle v_1, v_2 \rangle \triangleq v_1^T v_2$. ← standard inner product in R^n

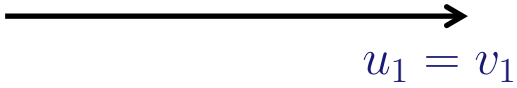
$\langle v_1, v_2 \rangle \triangleq v_1^T A v_2$, where $A = A^T$ is positive definite.

- Orthogonality: if $\langle v_1, v_2 \rangle = 0$, we say v_1 is orthogonal to v_2 with respect to inner product $\langle \cdot, \cdot \rangle$.
- The required property on $p_0, p_1, \ldots, p_{n-1}$ that $p_i^T A p_j = 0, \forall i \neq j$, is equivalent to say $\{p_0, p_1, \ldots, p_{n-1}\}$ is a set of mutually orthogonal vectors with respect to inner product $\langle v_1, v_2 \rangle \triangleq v_1^T A v_2$.
- The existence of such vectors is guaranteed by Gram-Schmidt orthogonalization.

Gram-Schmidt Orthogonalization

Gram–Schmidt orthogonalization is a process for orthogonalising a set of linearly independent vectors $\{v_1, v_2, \ldots, v_n\}$ in R^n with respect to any chosen inner product.

Step 1: set $u_1 = v_1$;

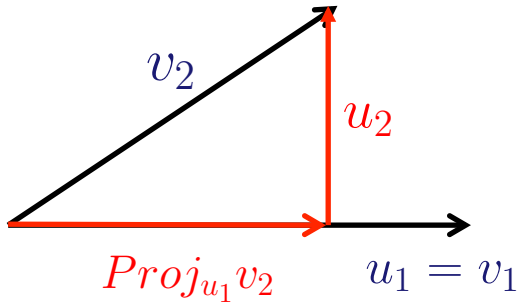


Gram-Schmidt Orthogonalization

Gram–Schmidt orthogonalization is a process for orthogonalising a set of linearly independent vectors $\{v_1, v_2, \ldots, v_n\}$ in R^n with respect to any chosen inner product.

Step 1: set $u_1 = v_1$;

Step 2: set $u_2 = v_2 - \dfrac{\langle v_2, u_1 \rangle}{\langle u_1, u_1 \rangle} u_1$; ← projection of v_2 on u_1



Gram-Schmidt Orthogonalization

Gram–Schmidt orthogonalization is a process for orthogonalising a set of linearly independent vectors $\{v_1, v_2, \ldots, v_n\}$ in R^n with respect to any chosen inner product.

Step 1: set $u_1 = v_1$;

Step 2: set $u_2 = v_2 - \dfrac{\langle v_2, u_1 \rangle}{\langle u_1, u_1 \rangle} u_1$;

Step 3: set $u_3 = v_3 - \dfrac{\langle v_3, u_1 \rangle}{\langle u_1, u_1 \rangle} u_1 - \dfrac{\langle v_3, u_2 \rangle}{\langle u_2, u_2 \rangle} u_2$;

projection of v_3 on the plane spanned by u_1 and u_2

$\vdots$

Step n: set $u_n = v_n - \displaystyle\sum_{i=1}^{n-1} \frac{\langle v_n, u_i \rangle}{\langle u_i, u_i \rangle} u_i$.

projection of v_n on the plane spanned by $u_1, u_2, \ldots, u_{n-1}$

Gram-Schmidt Orthogonalization

Gram–Schmidt orthogonalization is a process for orthogonalising a set of linearly independent vectors $\{v_1, v_2, \ldots, v_n\}$ in R^n with respect to any chosen inner product.

Step 1: set $u_1 = v_1$;

Step 2: set $u_2 = v_2 - \dfrac{\langle v_2, u_1 \rangle}{\langle u_1, u_1 \rangle} u_1$;

Step 3: set $u_3 = v_3 - \dfrac{\langle v_3, u_1 \rangle}{\langle u_1, u_1 \rangle} u_1 - \dfrac{\langle v_3, u_2 \rangle}{\langle u_2, u_2 \rangle} u_2$;

$\vdots$

Step n: set $u_n = v_n - \displaystyle\sum_{i=1}^{n-1} \frac{\langle v_n, u_i \rangle}{\langle u_i, u_i \rangle} u_i$.

Applying Gram-Schmidt orthogonalization with respect to inner product

$\langle v_1, v_2 \rangle \triangleq v_1^T A v_2$

generates conjugate directions

$\{p_0, p_1, \ldots, p_{n-1}\}$

with the property

$p_i^T A p_j = 0, \forall i \neq j$

The resulting set of vectors $\{u_1, u_2, \ldots, u_n\}$ satisfies $\langle u_i, u_j \rangle = 0, \forall\, i \neq j$.

Conjugate Direction Algorithm

Conjugate direction algorithm:

1. Applying Gram-Schmidt orthogonalization to construct search directions $p_0, p_1, \ldots, p_{n-1}$ such that

$$p_i^T A p_j = 0, \qquad \forall i \neq j$$

2. Performing line search along $p_0, p_1, \ldots, p_{n-1}$

$$\begin{cases} r_k &= b - Ax_k & \text{(The residual at step k.)} \\ \alpha_k &= \frac{r_k^T p_k}{p_k^T A p_k} & \text{(The step length of exact line search.)} \\ x_{k+1} &= x_k + \alpha_k p_k \end{cases}$$

Theorem 5.1.

For any $x_0 \in \mathbb{R}^n$ the sequence $\{x_k\}$ generated by the conjugate direction algorithm (5.6), (5.7) converges to the solution x^* of the linear system (5.1) in at most n steps.

Conjugate Gradient Methods

Conjugate Direction Algorithm is not practical, since generating conjugate directions, $p_0, p_1, \ldots, p_{n-1}$ by G-S is expansive (~n^3).

Is it possible to simplify G-S?

G-S: Given a set of linearly independent vectors $\{v_0, v_1, \ldots, v_{n-1}\}$

$$\{v_0\} \Longrightarrow \{p_0\}$$
$$\{p_0, v_1\} \Longrightarrow \{p_0, p_1\}$$
$$\{p_0, p_1, v_2\} \Longrightarrow \{p_0, p_1, p_2\}$$
$$\vdots$$
$$\{p_0, p_1, \ldots, p_{n-2}, v_{n-1}\} \Longrightarrow \{p_0, p_1, \ldots, p_{n-1}\}$$

Conjugate Gradient Methods

Conjugate Direction Algorithm is not practical, since generating conjugate directions, $p_0, p_1, \ldots, p_{n-1}$ by G-S is expansive (~n^3).

Is it possible to simplify G-S?

G-S: Given a set of linearly independent vectors $\{v_0, v_1, \ldots, v_{n-1}\}$

$$\{v_0\} \Longrightarrow \{p_0\}$$
$$\{p_0, v_1\} \Longrightarrow \{p_0, p_1\}$$
$$\{p_0, p_1, v_2\} \Longrightarrow \{p_0, p_1, p_2\}$$
$$\vdots$$
$$\{p_0, p_1, \ldots, p_{n-2}, v_{n-1}\} \Longrightarrow \{p_0, p_1, \ldots, p_{n-1}\}$$

$$\{r_0\} \Longrightarrow \{p_0\}$$
$$\{p_0, r_1\} \Longrightarrow \{p_0, p_1\}$$
$$\{p_0, p_1, r_2\} \Longrightarrow \{p_0, p_1, p_2\}$$
$$\vdots$$
$$\{p_0, p_1, \ldots, p_{n-2}, r_{n-1}\} \Longrightarrow \{p_0, p_1, \ldots, p_{n-1}\}$$

A fundamental idea of Conjugate Gradient methods:

At k-th step of G-S, pick v_k to be the residual, i.e., set $v_k = r_k = b - Ax_k$.

Conjugate Gradient Methods

At a given initial point x_0, the residual is $r_0 = b - Ax_0$.

Step 1: $p_0 = r_0 \Longrightarrow x_1 = x_0 + \alpha_0 p_0 \Longrightarrow r_1 = b - Ax_1;$

Step 2: $p_1 = r_1 - \dfrac{\langle r_1, p_0 \rangle}{\langle p_0, p_0 \rangle} p_0 \Longrightarrow x_2 = x_1 + \alpha_1 p_1 \Longrightarrow r_2 = b - Ax_2;$

Step 3: $p_2 = r_2 - \dfrac{\langle r_2, p_0 \rangle}{\langle p_0, p_0 \rangle} p_0 - \dfrac{\langle r_2, p_1 \rangle}{\langle p_1, p_1 \rangle} p_1 \Longrightarrow x_3 = x_2 + \alpha_2 p_2 \Longrightarrow r_3 = b - Ax_3;$

$\vdots$

Step k: $p_k = r_k - \dfrac{\langle r_k, p_0 \rangle}{\langle p_0, p_0 \rangle} p_0 - \cdots - \dfrac{\langle r_k, p_{k-2} \rangle}{\langle p_{k-2}, p_{k-2} \rangle} p_{k-2} - \dfrac{\langle r_k, p_{k-1} \rangle}{\langle p_{k-1}, p_{k-1} \rangle} p_{k-1};$

$\vdots$

Conjugate Gradient Methods

At a given initial point x_0, the residual is $r_0 = b - Ax_0$.

Step 1: $p_0 = r_0 \Longrightarrow x_1 = x_0 + \alpha_0 p_0 \Longrightarrow r_1 = b - Ax_1;$

Step 2: $p_1 = r_1 - \dfrac{\langle r_1, p_0 \rangle}{\langle p_0, p_0 \rangle} p_0 \Longrightarrow x_2 = x_1 + \alpha_1 p_1 \Longrightarrow r_2 = b - Ax_2;$

Step 3: $p_2 = r_2 - \dfrac{\langle r_2, p_0 \rangle}{\langle p_0, p_0 \rangle} p_0 - \dfrac{\langle r_2, p_1 \rangle}{\langle p_1, p_1 \rangle} p_1 \Longrightarrow x_3 = x_2 + \alpha_2 p_2 \Longrightarrow r_3 = b - Ax_3;$

$\vdots$

Step k: $p_k = r_k - \dfrac{\langle r_k, p_0 \rangle}{\langle p_0, p_0 \rangle} p_0 - \cdots - \dfrac{\langle r_k, p_{k-2} \rangle}{\langle p_{k-2}, p_{k-2} \rangle} p_{k-2} - \dfrac{\langle r_k, p_{k-1} \rangle}{\langle p_{k-1}, p_{k-1} \rangle} p_{k-1};$

$\vdots$

- These term are guaranteed to be zero. (We will provide the mathematical justification later.)
- The computational cost involved in G-S orthogonalization is significantly reduced.

Conjugate Gradient Methods

At a given initial point x_0, the residual is $r_0 = b - Ax_0$.

Step 1: $p_0 = r_0 \implies x_1 = x_0 + \alpha_0 p_0 \implies r_1 = b - Ax_1$

Step 2: $p_1 = r_1 - \dfrac{\langle r_1, p_0 \rangle}{\langle p_0, p_0 \rangle} p_0 \implies x_2 = x_1 + \alpha_1 p_1 \implies r_2 = b - Ax_2$

Step 3: $p_2 = r_2 - \dfrac{\langle r_2, p_1 \rangle}{\langle p_1, p_1 \rangle} p_1 \implies x_3 = x_2 + \alpha_2 p_2 \implies r_3 = b - Ax_3$

$\vdots$

Step k: $p_k = r_k - \dfrac{\langle r_k, p_{k-1} \rangle}{\langle p_{k-1}, p_{k-1} \rangle} p_{k-1} \implies x_{k+1} = x_k + \alpha_k p_k \implies r_{k+1} = b - Ax_{k+1}$

$\vdots$

$$\begin{cases} \alpha_k &= \dfrac{r_k^T p_k}{p_k^T A p_k} \\ x_{k+1} &= x_k + \alpha_k p_k \\ r_{k+1} &= b - Ax_{k+1} \\ p_{k+1} &= r_{k+1} - \dfrac{\langle r_{k+1}, p_k \rangle}{\langle p_k, p_k \rangle} p_k \end{cases}$$

This is essentially linear conjugate gradient method. (the inner product used here is always $\langle v_1, v_2 \rangle \triangleq v_1^T A v_2$)