Continuous Optimization

Unconstrained Optimization (part 2)

Sections covered in the textbook (2nd edition):

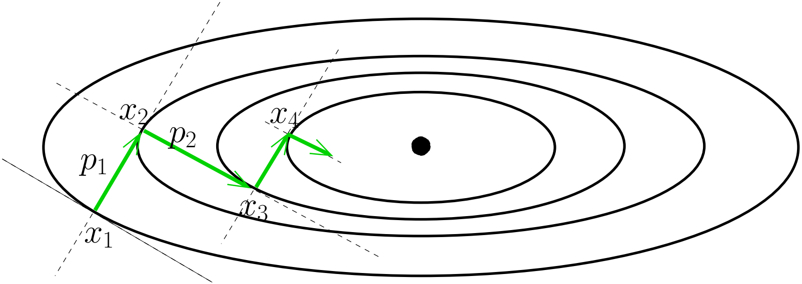
- Chapter 2: 1, 2
- Chapter 3: 1, 2, **3, 4**
- Chapter 5: **1, 2**
- Chapter 6: **1**
- Chapter 10: **1, 2, 3**

Steepest Decent $p_k = -\nabla f(x_k)$

When $f(x) = \frac{1}{2}x^t A x - b^t x$ with A positive definite,
$p_k = -\nabla f(x_k) = b - Ax_k = -r_k$.

$$\phi(\alpha) = f(x_k + \alpha p_k)$$

$$\phi'(\alpha_k) = 0 \implies \alpha_k = \frac{p_k^t p_k}{p_k^t A p_k} \implies x_{k+1} = x_k + \frac{p_k^t p_k}{p_k^t A p_k} p_k.$$



Steepest decent for $f(x) = \frac{1}{2}x^tAx - b^tx$

$$\boxed{x_{k+1} = x_k + \frac{p_k^t p_k}{p_k^t A p_k} p_k, \quad p_k = -\nabla f(x_k) = b - Ax_k.}$$

- $f(x_k) - f(x^*) = \dfrac{1}{2}(x - x^*)^t A(x - x_k)^t \triangleq \dfrac{1}{2}\|x - x^*\|_A^2$
- $f(x_1) \geq f(x_2) \geq \cdots \geq f(x^*)$ ☺
- (Convergence Rate) Let the eigenvalues of A be $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, then

$$\|x_{k+1} - x^*\|_A \leq \frac{\lambda_n - \lambda_1}{\lambda_n - \lambda_1}\|x_k - x^*\|_A$$

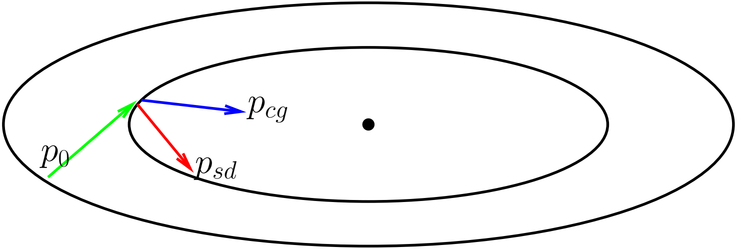
When the size n of the system is large, usually λ_n/λ_1 is large and this method converges slowly. ☹

Motivation for Conjugate Gradient Method

Let the minimizer for $f(x) = \frac{1}{2}x^tAx - b^tx$ be x^*. For n linearly independent vectors $p_1, p_2, \cdots, p_n$, if

$$x^* = x_0 + \alpha_1 p_1 + \cdots + \alpha_n p_n.$$

On way is to find the component $\alpha_k p_k$ step by step, such that $x_k = x_{k-1} + \alpha_k p_k$, with $\alpha_k = \frac{p_k^t p_k}{p_k^t A p_k}$.
For steepest decent, we have $p_k \cdot p_{k+1} = 0$.
For $n = 2$, we have $p_1 \parallel p_3 \parallel p_5 \parallel \cdots$, $p_2 \parallel p_4 \parallel \cdots$, not so efficient.



Motivation for Conjugate Gradient Method

The best we can hope is that the directions p_1, $p_2, \cdots, p_n$ are "orthogonal" to each other. At kth step, we get the coefficient α_k in the expansion

$$x^* = x_0 + \alpha_1 p_1 + \cdots + \alpha_n p_n.$$

It is better to enforce the conjugate (orthogonal) condition like $p_i^t A p_j = 0$ instead of $p_i^t p_j = 0$ in the usual sense. In this case, the coefficient can be written in terms of x^*, x_0, p_i and A as

$$\alpha_k =$$

The conjugate gradient method generates the conjugate vectors p_k and α_k at the kth step.

Conjugate Gradient Method

Starting with x_0, $r_0 = Ax_0 - b$, $p_0 = -r_0$ (the only choice for the first step) and

$$x_1 = x_0 + \alpha_0 p_0, \qquad \alpha_0 =$$

Next $r_1 = Ax_1 - b = \alpha A p_0 - p_0$. We want to get p_1 by modifying r_1 such that $p_1^t A p_0 = 0$.

$$p_1 = -r_1 + \beta_1 p_0, \qquad \beta_1 =$$

Continuing with similar formula?

Conjugate Gradient Method

The conjugate condition $p_i^t A p_j = 0 (j < i)$ is satisfied automatically when at the $k$the step, we only require p_{k+1} is obtained from r_{k+1} by with a difference of p_k.

Given x_0;
Set $r_0 \leftarrow Ax_0 - b, p_0 \leftarrow -r_0, k \leftarrow 0$;
while $\|r_k\| > \epsilon$ **do**

$$\alpha_k \leftarrow -\frac{r_k^t p_k}{p_k^t A p_k};$$
$$x_{k+1} \leftarrow x_k + \alpha_k p_k;$$
$$r_{k+1} \leftarrow Ax_{k+1} - b;$$
$$\beta_{k+1} \leftarrow \frac{r_{k+1}^t A p_k}{p_k^t A p_k};$$
$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k;$$
$$k \leftarrow k + 1;$$

end

Conjugate Gradient Method: Properties

- $r_k^t r_i = 0$ for $i = 0, 1, \cdots, k-1$
- $\text{span}\{r_0, r_1, \cdots, r_k\} = \text{span}\{r_0, Ar_0, \cdots, A^k r_0\} = \text{span}\{p_0, p_1, \cdots, p_k\}$
- $p_k^t A p_i = 0$ for $i = 0, 1, \cdots, k-1$
- $\{x_k\}$ converges to x^* at most n steps
- Convergence rate $0 < \lambda_1 \leq \cdots \lambda_n$

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

and with the condition number $\kappa(A) = \lambda_n / \lambda_1$

$$\|x_{k+1} - x^*\|_A \leq \frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} - 1} \|x_0 - x^*\|_A$$

Comments on Steepest decent and CG

- When A is still nonsingular but not symmetric, we can still solve the *normal equation* $A^tAx = A^tb$, but the *condition number* (can be taken as λ_n/λ_1) is squared, and the convergence is slower and the accuracy of the solution may not be enough.
- They can be applied to nonlinear problems f other than the quadratic functions of the form $\tilde{f}(x) = \frac{1}{2}x^tAx - b^tx$ with

$$b = -\nabla f(x_k), \qquad A = \nabla^2 f(x_k).$$

Newton's Method

If the approximation x_k is close to the minimizer, for $d_k = x^* - x_k$

$$f(x^*) = f(x_k + d_k) \approx f(x_k) + d_k \cdot \nabla f(x_k) + \frac{1}{2} d_k^t \nabla^2 f(x_k) d_k.$$

The minimizer d_k^* for the quadratic function is

$$d_k^* =$$

and the approximation at next step is

$$x_{k+1} = x_k + d_k =$$

Newton's Method $x_{k+1} = x_k + d_k^*$

$$d_k^* = \text{argmin } f(x_k) + d \cdot \nabla f(x_k) + \frac{1}{2} d^t \nabla^2 f(x_k) d \qquad (1)$$

Theorem (Convergence Rate for Newton's Method)

If f'' is continuous and invertible near a solution x^*, then convergence of Newton's method is Q-superlinear. If, in addition, f''' is continuous, the convergence is Q-quadratic.

Questions:

- Near a strict minimizer, why does the minimizer in (**??**) exist?
- What's the iterative scheme for finding the local maximizers of a function f?
- Any potential problem when f'' (or $\nabla^2 f$) is not invertible near x^*? Try $f(x) = x^4$ and $x_1 = 1$.
- How fast $\|\nabla f(x_k)\|$ decays to zero?

Newton's Method

Drawbacks

- Converges only when x_1 is close enough to x^*, otherwise diverges violently.
- The divergence is usually related to the fact that $\nabla^2 f(x_k)$ is singular. One way is to modify the Hessian matrix $\nabla^2 f(x_k)$ by a small identity matrix to be $\nabla^2 f(x_k) + \tau I$.
- Computational intensive when the dimension of the variable is large
- Is is clear $f(x_{k+1}) < f(x_k)$? The relaxed version may be more practical:

  $$x_{k+1} = x_k + \alpha_k d_k^*,$$

  where α_k is a scalar constant between 0 and 1 (very often just a small positive constant say $\alpha_k = 0.1$).

Quasi-Newton Method (for large scale problems)

The direction at each step for Steepest Decent and Newton's method

$$d_{sd} = -\nabla f(x_k), \qquad d_{newton} = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$$

Suggesting the general scheme
$d = -B_k^{-1} \nabla f(x_k) = -W_k \nabla f(x_k)$ such that B_k^{-1} is easier (faster) to compute then using linear search method to find the length α_k in $x_{k+1} = x_k + \alpha_k d_k$.

What kind of properties B_k or W_k should satisfy?

- B_k should be "close " to $\nabla^2 f(x_k)$
- The function $f(x_k + \alpha d)$ should decrease for α small and positive.

Quasi-Newton Method

Decent direction $p_k = -B_k^{-1}\nabla f(x_k)$.
Let

$$y_k = \nabla f(x_{k+1}) - \nabla f(x_k), \quad s_k = x_{k+1} - x_k = \alpha_k p_k,$$

by Taylor Expansion

$$y_k = \nabla^2 f(\xi_k)(x_{k+1} - x_k) = \nabla^2 f(\xi_k) s_k.$$

This suggest the **secant equation**

$$B_{k+1} s_k = y_k.$$

The approximation B_{k+1} to the Hessian matrix should be positive definite, or the **curvature condition**

$$s_k^t y_k > 0.$$

Different Quasi-Newton Method

Let $H_k = B_k^{-1}$, we update H_k instead of B_k^{-1}, to reduce the time in computing the inverse of a matrix.
Davidon-Fletcher-Powell (DFP)

$$H_{k+1} = H_k + \frac{s_k s_k^t}{y_k^t s_k} - \frac{H_k y_k y_k^t H_k}{y_k^t H_k y_k}.$$

Broyden-Fletcher-Goldfarb-Shanno (BFGS)

$$B_{k+1} = B_k + \frac{y_k y_k^t}{y_k^t s_k} - \frac{B_k s_k s_k^t B_k}{s_k^t B_k s_k}.$$

or

$$H_{k+1} = H_k + \left[1 + \frac{y_k^t H_k y_k}{y_k^t s_k}\right] \frac{s_k s_k^t}{y_k^t s_k} - \frac{s_k y_k^t H_k + H_k y_k s_k^t}{y_k^t H_k y_k}.$$

Comparison for Steepest Decent, CG, Newton and Quasi-Newton

- Required information: Gradient, with/without Hessian
- Different problems: applicable to min and/or max, quadratic functions or general nonlinear functions
- Different kind of approximation:
- Convergence rate: Q-linear, Q-superlinear, Q-quadratic