Lecture 3: Additional First Order Methods

- Incremental Gradient Method
- Conjugate Directions
- Conjugate Gradient Method
- Coordinate Descent Method
- Quasi-Newton Methods

Least-Squares Problems and Incremental Gradient Methods

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) = \frac{1}{2}\|g(\boldsymbol{x})\|^2 = \frac{1}{2}\sum_{i=1}^{m} g_i(\boldsymbol{x})^2 \\ \text{subject to} & \boldsymbol{x} \in \mathbb{R}^n, \end{array}$$

where $\boldsymbol{g} = (g_1, ..., g_m)^T$, $g_i : \mathbb{R}^n \mapsto \mathbb{R}^{r_i}$.

- Steepest descent method

$$\boldsymbol{x}^{r+1} = \boldsymbol{x}^r - \alpha_r \nabla f(\boldsymbol{x}^r) = \boldsymbol{x}^r - \alpha_r \sum_{i=1}^{m} \nabla g_i(\boldsymbol{x}^r) g_i(\boldsymbol{x}^r)$$

- Incremental gradient method:

$$\begin{aligned} \psi^i &= \psi^{i-1} - \alpha_r \nabla g_i(\psi^{i-1}) g_i(\psi^{i-1}), \quad i = 1, ..., m \\ \psi^0 &= \boldsymbol{x}^r, \quad \boldsymbol{x}^{r+1} = \psi^m \end{aligned}$$

- Advantage of incrementalism

View as Gradient Method W/ Errors

- Can write incremental gradient method as

$$\begin{aligned} \boldsymbol{x}^{r+1} &= \boldsymbol{x}^r - \alpha_r \sum_{i=1}^{m} \nabla g_i(\boldsymbol{x}^r) g_i(\boldsymbol{x}^r) \\ &\quad + \alpha_r \sum_{i=1}^{m} \left(\nabla g_i(\boldsymbol{x}^r) g_i(\boldsymbol{x}^r) - \nabla g_i(\psi^{i-1}) g_i(\psi^{i-1}) \right) \end{aligned}$$

- Error term is proportional to stepsize α_r
- Convergence (generically) for a diminishing stepsize (under a Lipschitz condition on $g_i \nabla g_i$)
- Convergence to a neighborhood of $\boldsymbol{x}^*$ (the minimizer of f) for a constant stepsize

Convergence of Incremental Gradient Method

Example: Consider minimizing $f(x) = \frac{1}{2}(x - c_1)^2 + \frac{1}{2}(x - c_2)^2$. Clearly, f is strongly convex and $x^* = (c_1 + c_2)/2$. Let $x^0 = 0$. The incremental gradient method is

$$\begin{aligned} x^r(2) &= x^r(1) - \alpha(x^r(1) - c_1) \\ x^{r+1}(1) &= x^r(2) - \alpha(x^r(2) - c_2), \end{aligned}$$

where $x^r(i)$, $i = 1, 2$, denotes the iterate just before the i-th component in the r-th cycle.

It can be checked

$$x^{r+1}(1) = (1 - \alpha)^2 x^r(1) + (1 - \alpha)\alpha c_1 + \alpha c_2.$$

For $0 < \alpha < 1$, the sequence $x^r(1) \to \frac{(1-\alpha)c_1+c_2}{2-\alpha} = x_\alpha(1)$, and similarly, $\lim_{r\to\infty} x^r(2) = x_\alpha(2) = \frac{(1-\alpha)c_2+c_1}{2-\alpha}$.

Thus, for fixed step size α, the sequence of iterates will oscillate between two limiting points $x_\alpha(1)$ and $x_\alpha(2)$. Notice that

$$|x_\alpha(1) - x^*| = |x_\alpha(2) - x^*| = O(\alpha),$$

suggesting when $\alpha \to 0$, both $x_\alpha(1)$ and $x_\alpha(2)$ will converge $x^* = (c_1 + c_2)/2$.

Dynamically decreasing step sizes $\alpha^r \to 0$:

- too slow $\Rightarrow$ $\{x^r(1)\}$ and $\{x^r(2)\}$ still converge to two different limit points.
- too fast $\Rightarrow$ the iterates will not reach x^*.

With $\alpha^r = 1/r$, we have

$$x^{r+1}(2) = \frac{r-1}{r+1} x^r(2) + \frac{c_1 + c_2}{r+1}, \qquad \forall r \geq 1.$$

This implies $x^r(2) \to x^*$. Similarly, $x^r(1) \to x^*$.

Convergence of Incremental Gradient Method

- Choose the component function f_i cyclicly.
- Convergence depends on the choice of stepsizes: square summable, infinite travel
- **Assumption:** X^* is nonempty; the iterates lie in a bounded set X, and for every i, the gradient $\nabla f_i(\boldsymbol{x})$ is uniformly bounded a constant C_i over X.
- With above stepsize rule and under this assumption, the sequence of iterates $\{\boldsymbol{x}^r\}$ converges to a solution in X^*.
- Convergence rate is typically sublinear and sensitive to the step size (e.g., choose $\alpha^r = \theta/r$, count one cycle of updates as one iteration)
- More detailed analysis will be given later when we deal with constrained optimization.

Example

The incremental gradient algorithm is sensitive to choice of θ. Consider

$$f(x) = \frac{1}{2}cx^2, \quad \text{with } c = 0.2$$

Suppose, further, that we take $\theta = 1$, i.e., $\alpha^r = 1/r$. Then the iteration process becomes

$$x^{r+1} = x^r - f'(x^r)/r = \left(1 - \frac{1}{5r}\right) x^r$$

and hence starting with $x^1 = 1$,

$$x^r = \prod_{i=1}^{r-1} \left(1 - \frac{1}{5i}\right) \geq 0.8r^{-1/5},$$

implying very slow convergence. For example, for $r = 10^9$ the solution error is still ≥ 0.015.

The optimal choice of $\theta = c^{-1} = 5$ generates the optimal solution $x^* = 0$ in one iteration.

Conjugate Direction Methods

- Aim to improve convergence rate of steepest descent, without incurring the overhead of Newton's method
- Analyzed for a quadratic model. They require n iterations to minimize $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}'\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{b}'\boldsymbol{x}$ with $\boldsymbol{Q}$ an $n \times n$ positive definite matrix.
- Analysis also applies to non-quadratic problems in the neighborhood of a nonsingular local min
- Directions $\boldsymbol{d}^1, ..., \boldsymbol{d}^r$ are $\boldsymbol{Q}$-conjugate, if $(\boldsymbol{d}^i)'\boldsymbol{Q}\boldsymbol{d}^j = 0$ for all $i \neq j$.
- Generic conjugate direction method:

$$\boldsymbol{x}^{r+1} = \boldsymbol{x}^r + \alpha_r \boldsymbol{d}^r$$

where the $\boldsymbol{d}^r$'s are $\boldsymbol{Q}$-conjugate and α_r is obtained by line minimization

Generating Conjugate Directions

- Given set of linearly independent vectors $\boldsymbol{\xi}^0, \ldots, \boldsymbol{\xi}^k$, we can construct a set of $\boldsymbol{Q}$-conjugate directions $\boldsymbol{d}^0, \ldots, \boldsymbol{d}^r$ s.t. $\text{Span}(\boldsymbol{d}^0, \ldots, \boldsymbol{d}^i) = \text{Span}(\boldsymbol{\xi}^0, \ldots, \boldsymbol{\xi}^i)$

- Gram-Schmidt procedure. Start with $\boldsymbol{d}^0 = \boldsymbol{\xi}^0$. If for some $i < r$, $\boldsymbol{d}^0, \ldots, \boldsymbol{d}^i$ are $\boldsymbol{Q}$-conjugate and the above property holds, take

$$\boldsymbol{d}^{i+1} = \boldsymbol{\xi}^{i+1} + \sum_{m=0}^{i} c_{(i+1),m} \boldsymbol{d}^m$$

choose $c_{(i+1),m}$ so $\boldsymbol{d}^{i+1}$ is $\boldsymbol{Q}$-conjugate to $\boldsymbol{d}^0, \ldots, \boldsymbol{d}^i$,

$$(\boldsymbol{d}^{i+1})' \boldsymbol{Q} \boldsymbol{d}^j = (\boldsymbol{\xi}^{i+1})' \boldsymbol{Q} \boldsymbol{d}^j + \left(\sum_{m=0}^{i} c_{(i+1),m} \boldsymbol{d}^m \right)' \boldsymbol{Q} \boldsymbol{d}^j = 0,$$

implying

$$c_{(i+1),j} = -\frac{(\boldsymbol{\xi}^{i+1})' \boldsymbol{Q} \boldsymbol{d}^j}{(\boldsymbol{d}^j)' \boldsymbol{Q} \boldsymbol{d}^j}, \quad \forall\ 0 \le j \le i.$$

The Conjugate Gradient Method

- Apply Gram-Schmidt to the vectors $\boldsymbol{\xi}^r = -\boldsymbol{g}^r = -\nabla f(\boldsymbol{x}^r),\ r = 0, 1, \ldots, n-1$

$$\boldsymbol{d}^r = -\boldsymbol{g}^r + \sum_{j=0}^{r-1} \frac{(\boldsymbol{g}^r)'\boldsymbol{Q}\boldsymbol{d}^j}{(\boldsymbol{d}^j)'\boldsymbol{Q}\boldsymbol{d}^j}\boldsymbol{d}^j$$

- **Key fact:** Direction formula can be simplified! The directions of the CGM are generated by $\boldsymbol{d}^0 = -\boldsymbol{g}^0$, and

$$\boldsymbol{d}^r = -\boldsymbol{g}^r + \beta_r \boldsymbol{d}^{r-1}, \quad r = 1, \ldots, n-1, \tag{1}$$

  where β_r is given by

$$\beta_r = \frac{(\boldsymbol{g}^r)'\boldsymbol{g}^r}{(\boldsymbol{g}^{r-1})'\boldsymbol{g}^{r-1}} \quad \text{or} \quad \beta_r = \frac{(\boldsymbol{g}^r - \boldsymbol{g}^{r-1})'\boldsymbol{g}^r}{(\boldsymbol{g}^{r-1})'\boldsymbol{g}^{r-1}}$$

- Iterations: $\boldsymbol{x}^0 \to \nabla f(\boldsymbol{x}^0) \to \boldsymbol{d}^0 \to \boldsymbol{x}^1 \to \nabla f(\boldsymbol{x}^1) \to \boldsymbol{d}^1 \to \boldsymbol{x}^2 \to \nabla f(\boldsymbol{x}^2) \to \boldsymbol{d}^2 \cdots$
- Furthermore, the method terminates with an optimal solution after at most n steps.
- Extension to non-quadratic problems: loss of conjugacy, periodically restart with steepest descent, rate of convergence, preconditioned CG.

Convergence of CGM

- Use induction to show that for all $r \geq 0$, each $\boldsymbol{g}^{r+1}$ generated up to termination is orthogonal to Span$(\boldsymbol{d}^0, \boldsymbol{d}^1, ..., \boldsymbol{d}^r)$.
  - ⋆ For each $r \geq 0$, exact line search implies
$$\left.\frac{\partial f(\boldsymbol{x}^r + \alpha \boldsymbol{d}^r)}{\partial \alpha}\right|_{\alpha=\alpha^r} = \nabla f(\boldsymbol{x}^{r+1})' \boldsymbol{d}^r = 0.$$
  - ⋆ Moreover, for any $i < r$, we have

$$\begin{aligned} \nabla f(\boldsymbol{x}^{r+1})' \boldsymbol{d}^i &= (\boldsymbol{Q}\boldsymbol{x}^{r+1} + \boldsymbol{b})' \boldsymbol{d}^i \\ &= \left(\boldsymbol{x}^{i+1} + \sum_{j=i+1}^{r} \alpha_j \boldsymbol{d}^j \right)' \boldsymbol{Q}\boldsymbol{d}^i + \boldsymbol{b}'\boldsymbol{d}^i \\ &= (\boldsymbol{x}^{i+1})' \boldsymbol{Q}\boldsymbol{d}^i + \boldsymbol{b}'\boldsymbol{d}^i \\ &= \left(\nabla f(\boldsymbol{x}^{i+1})\right)' \boldsymbol{d}^i = 0. \end{aligned}$$

  - ⋆ Can use this property to show the simplified formula (1).

Convergence of CGM

- Use induction to show that for all $r \geq 1$, the gradient vectors $\{\boldsymbol{g}^0, \boldsymbol{g}^1, ..., \boldsymbol{g}^{r-1}\}$ generated up to termination are linearly independent (in fact orthogonal).

- True for $r = 1$. Suppose no termination after r steps, and $\boldsymbol{g}^0, ..., \boldsymbol{g}^{r-1}$ are linearly independent. Then, $\text{Span}(\boldsymbol{d}^0, ..., \boldsymbol{d}^{r-1}) = \text{Span}(\boldsymbol{g}^0, ..., \boldsymbol{g}^{r-1})$ and there are two possibilities:
  - ⋆ $\boldsymbol{g}^r = \boldsymbol{0}$, and the method terminates.
  - ⋆ $\boldsymbol{g}^r \neq \boldsymbol{0}$, in which case

$$\boldsymbol{g}^r \text{ is orthogonal to } \{\boldsymbol{d}^0, ..., \boldsymbol{d}^{r-1}\} \quad \Rightarrow \quad \boldsymbol{g}^r \text{ is orthogonal to } \{\boldsymbol{g}^0, ..., \boldsymbol{g}^{r-1}\}$$

  so $\boldsymbol{g}^r$ is linearly independent of $\boldsymbol{g}^0, ..., \boldsymbol{g}^{r-1}$, completing the induction.

- Since at most n linearly independent gradients can be generated, $\boldsymbol{g}^r = \boldsymbol{0}$ for some $r \leq n$.

- Let $\boldsymbol{\beta}_r = (\beta_0, \beta_1, \ldots, \beta_r)^T$ and $\boldsymbol{\alpha}_r = (\alpha_0, \alpha_1, \ldots, \alpha_r)^T$. Then

$$\left.\frac{\partial f(\boldsymbol{x}^{r+1} + \beta_0 \boldsymbol{d}^0 + \beta_1 \boldsymbol{d}^1 + \cdots + \beta_r \boldsymbol{d}^r)}{\partial \beta_i}\right|_{\boldsymbol{\beta}_r = \boldsymbol{\alpha}_r} = \nabla f(\boldsymbol{x}^{r+1})' \boldsymbol{d}^i = 0.$$

- Therefore, we have

$$\boldsymbol{x}^{r+1} = \arg\min_{\boldsymbol{x} \in \mathcal{M}^r} f(\boldsymbol{x}), \quad \text{where } \mathcal{M}^r = \{\boldsymbol{x} \mid \boldsymbol{x} = \boldsymbol{x}^0 + \boldsymbol{v},\ \boldsymbol{v} \in \text{Span}(\boldsymbol{d}^0, \boldsymbol{d}^1, \ldots, \boldsymbol{d}^r)\}.$$

- This further implies $f(\boldsymbol{x}^r) \downarrow f(\boldsymbol{x}^*)$ monotonically, and $\boldsymbol{x}^n$ minimizes $f(\boldsymbol{x})$ over $\mathbb{R}^n$.

Coordinate Descent Method

- Instead of fixing the stepsizes, we can fix search directions. For instances, choose search directions from the coordinate directions $\{e^1, e^2, ..., e^n\}$.
- The stepsizes can be either constant, Armijo or diminishing.
- Iterate through the list of search directions (almost) cyclically.
- Each cycle is equivalent to one gradient descent iteration.
- No improvement after one cycle $\Leftrightarrow$ stationarity.
- Caution: only works for smooth functions.

Coordinate Descent Method

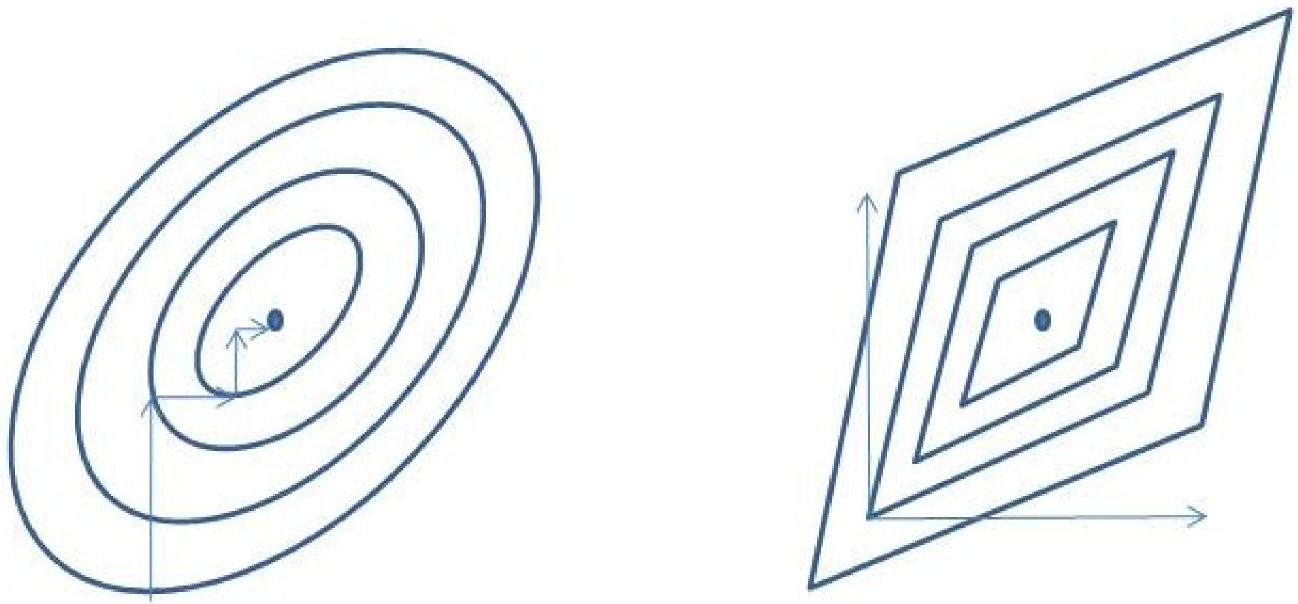

Figure 1: CD method for smooth/non-smooth minimization

Non-smoothness can cause the CD method to get stuck!

Quasi-Newton Methods

- $\boldsymbol{x}^{r+1} = \boldsymbol{x}^r - \alpha_r \boldsymbol{D}^r \nabla f(\boldsymbol{x}^r)$, where $\boldsymbol{D}^r$ is an inverse Hessian approximation
- Key idea: Successive iterates $\boldsymbol{x}^r, \boldsymbol{x}^{r+1}$ and gradients $\nabla f(\boldsymbol{x}^r),\ \nabla f(\boldsymbol{x}^{r+1})$, yield curvature info

$$\boldsymbol{q}^r \approx \nabla^2 f(\boldsymbol{x}^{r+1})\boldsymbol{p}^r,$$

$$\boldsymbol{p}^r = \boldsymbol{x}^{r+1} - \boldsymbol{x}^r, \quad \boldsymbol{q}^r = \nabla f(\boldsymbol{x}^{r+1}) - \nabla f(\boldsymbol{x}^r)$$

$$\nabla^2 f(\boldsymbol{x}^n) \approx \left[\boldsymbol{q}^0 \cdots \boldsymbol{q}^{n-1}\right]\left[\boldsymbol{p}^0 \cdots \boldsymbol{p}^{n-1}\right]^{-1}$$

- Most popular Quasi-Newton methods (e.g. BFGS) use clever ways to implement this idea

$$\boldsymbol{D}^{r+1} = \boldsymbol{D}^r + \frac{\boldsymbol{p}^r(\boldsymbol{p}^r)'}{\boldsymbol{p}^r(\boldsymbol{q}^r)'} - \frac{\boldsymbol{D}^r\boldsymbol{q}^r(\boldsymbol{q}^r)'(\boldsymbol{D}^r)'}{(\boldsymbol{q}^r)'\boldsymbol{D}^r\boldsymbol{q}^r} + \xi_r\tau_r\boldsymbol{v}^r(\boldsymbol{v}^r)',$$

$$\boldsymbol{v}^r = \frac{\boldsymbol{p}^r}{(\boldsymbol{p}^r)'\boldsymbol{q}^r} - \frac{\boldsymbol{D}^r\boldsymbol{q}^r}{\tau_r}, \quad \tau_r = (\boldsymbol{q}^r)'\boldsymbol{D}^r\boldsymbol{q}^r, \quad 0 \le \xi_r \le 1$$

and $\boldsymbol{D}^0 \succ \boldsymbol{0}$ is arbitrary, α_r by line minimization, and $\boldsymbol{D}^n = \boldsymbol{Q}^{-1}$ for a quadratic.