Chapter 5. Unconstrained Optimization

- Unconstrained minimization problems
- Descent methods
- Newton's method
- Conjugate direction methods
- Majorization minimization

Unconstrained minimization problems

- Model problem

$$\min_{\mathbf{x}} \quad f(\mathbf{x}), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and twice continuously differentiable (which implies that $\mathbf{dom} f$ is open). We will assume that the problem is solvable, *i.e.*, there exists an optimal point $\mathbf{x}^*$.

Since f is differentiable and convex, a necessary and sufficient condition for a point $\mathbf{x}^*$ to be optimal is

$$\boxed{\nabla f(\mathbf{x}^*) = \mathbf{0}.} \tag{2}$$

Optimility condition

Usually we need to design an algorithm that computes a sequence of points $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots \in \mathbf{dom} f$ with $f(\mathbf{x}^{(k)}) \to p^*$ as $k \to \infty$. Such a sequence of points is called a minimizing sequence for the problem (1). The algorithm is terminated when $\boxed{f(\mathbf{x}^{(k)}) - p^* \leq \epsilon,}$ where $\epsilon > 0$ is some specified tolerance.

Checked by: $\|\nabla f(\mathbf{x}^k)\| \leq \varepsilon$ (stopping criterion)

Unconstrained minimization problems

- Directly solving the optimality condition

$$\nabla f(\mathbf{x}) = \mathbf{0}$$

Examples. 1. Quadratic minimization and least-squares:

$$\min_{\mathbf{x}} (1/2)\mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x} + r,$$

where $\mathbf{P} \in \mathbb{S}_+^n, \mathbf{q} \in \mathbb{R}^n$, and $r \in \mathbb{R}$.

2. The optimality condition is:

$$\begin{aligned}\mathbf{0} \in & \mu_i(\phi^{-1}(X^i) - W^{i-1}X^{i-1}) \\ & + \mu_{i+1}((W^i)^T(\phi(W^iX^i) - X^{i+1})).\end{aligned}$$

Jia Li, Cong Fang, and Zhouchen Lin, Lifted Proximal Operator Machines, AAAI 2019.

Unconstrained minimization problems

- Fixed-point algorithm

$$\mathbf{x} = \varphi(\mathbf{x}) \quad \Rightarrow \quad \mathbf{x}_{k+1} = \varphi(\mathbf{x}_k)$$

www.docin.com

Unconstrained minimization problems

- Fixed-point algorithm

Theorem 1 (Banach Fixed Point Theorem)**.** *Let (X, d) be a non-empty complete metric space with a contraction mapping $T : X \to X$:*

$$d(T(\mathbf{x}), T(\mathbf{y})) \leq Ld(\mathbf{x}, \mathbf{y}), \text{ where } 0 \leq L < 1.$$

Then T admits a unique fixed-point $\mathbf{x}^*$ in X. Furthermore, $\mathbf{x}^*$ can be found as follows: start with an arbitrary element $\mathbf{x}_0$ in X and define a sequence $\{\mathbf{x}_n\}$ by $\mathbf{x}_n = T(\mathbf{x}_{n-1})$, then $\mathbf{x}_n \to \mathbf{x}^*$.

Corollary 1. *If φ is a Lipschitz continuous function*

$$\|\varphi(\mathbf{x}) - \varphi(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \text{ where } 0 \leq L < 1,$$

then iteration $\mathbf{x}_{k+1} = \varphi(\mathbf{x}_k)$ converges globally to the unique fixed point of φ.

Unconstrained minimization problems

- Fixed-point algorithm for solving the optimality condition

$$\nabla f(\mathbf{x}) = \mathbf{0} \iff \mathbf{x} = \mathbf{x} - \alpha \nabla f(\mathbf{x})$$

$$\Downarrow$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)$$

gradient descent

Unconstrained minimization problems

- Strong convexity and implications

Strong convexity:

$$\nabla^2 f(\mathbf{x}) \succeq m\mathbf{I}.$$

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \frac{m}{2}\|\mathbf{y}-\mathbf{x}\|_2^2, \tag{1}$$

Bound $f(\mathbf{x}) - p^*$ in terms of $\|\nabla f(\mathbf{x})\|_2$. The righthand side of (1) is a convex quadratic function of $\mathbf{y}$ (for fixed $\mathbf{x}$). Setting the gradient with respect to $\mathbf{y}$ equal to zero, we find that $\tilde{\mathbf{y}} = \mathbf{x} - (1/m)\nabla f(\mathbf{x})$ minimizes the righthand side. Therefore we have

$$\begin{aligned} f(\mathbf{y}) \geq & f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \frac{m}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 \\ \geq & f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\tilde{\mathbf{y}}-\mathbf{x}) + \frac{m}{2}\|\tilde{\mathbf{y}}-\mathbf{x}\|_2^2 \\ = & f(\mathbf{x}) - \frac{1}{2m}\|\nabla f(\mathbf{x})\|_2^2. \end{aligned}$$

Unconstrained minimization problems

- Strong convexity and implications

Since this holds for any $\mathbf{y} \in S$, we have

$$p^* \geq f(\mathbf{x}) - \frac{1}{2m}\|\nabla f(\mathbf{x})\|_2^2. \tag{2}$$

$$\|\nabla f(\mathbf{x})\|_2 \leq (2m\epsilon)^{1/2} \Rightarrow f(\mathbf{x}) - p^* \leq \epsilon \tag{3}$$

Unconstrained minimization problems

- Strong convexity and implications

We can also derive a bound on $\|\mathbf{x} - \mathbf{x}^*\|_2$ in terms of $\|\nabla f(\mathbf{x})\|_2$.

$$\|\mathbf{x} - \mathbf{x}^*\|_2 \leq \frac{2}{m}\|\nabla f(\mathbf{x})\|_2. \tag{4}$$

Proof. We apply (1) with $\mathbf{y} = \mathbf{x}^*$ to obtain

$$\begin{aligned} p^* = f(\mathbf{x}^*) \geq & f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{x}^* - \mathbf{x}) + \frac{m}{2}\|\mathbf{x}^* - \mathbf{x}\|_2^2 \\ \geq & f(\mathbf{x}) - \|\nabla f(\mathbf{x})\|_2\|\mathbf{x}^* - \mathbf{x}\|_2 + \frac{m}{2}\|\mathbf{x}^* - \mathbf{x}\|_2^2. \end{aligned}$$

Since $p^* \leq f(\mathbf{x})$, we must have

$$-\|\nabla f(\mathbf{x})\|_2\|\mathbf{x}^* - \mathbf{x}\|_2 + \frac{m}{2}\|\mathbf{x}^* - \mathbf{x}\|_2^2 \leq 0,$$

from which (4) follows. One consequence of (4) is that $\mathbf{x}^*$ is unique.

Unconstrained minimization problems

- Strong convexity and implications

Upper bound on $\nabla^2 f(\mathbf{x})$

Therefore the maximum eigenvalue of $\nabla^2 f(\mathbf{x})$, which is a continuous function of $\mathbf{x}$ on S, is bounded above on S, *i.e.*, there exists a constant M such that

$$\nabla^2 f(\mathbf{x}) \preceq M\mathbf{I}, \tag{5}$$

for all $\mathbf{x} \in S$. This upper bound on the Hessian implies for any $\mathbf{x}, \mathbf{y} \in S$,

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{M}{2}\|\mathbf{y} - \mathbf{x}\|_2^2, \tag{6}$$

which is analogous to (1). Minimizing each side over $\mathbf{y}$ yields

$$p^* \leq f(\mathbf{x}) - \frac{1}{2M}\|\nabla f(\mathbf{x})\|_2^2, \tag{7}$$

the counterpart of (2).

Descent methods

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \boxed{t^{(k)}}\boxed{\Delta\mathbf{x}^{(k)}},$$

search direction

step size

where $t^{(k)} > 0$ (except when $\mathbf{x}^{(k)}$ is optimal).

Descent methods mean that

$$f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)}),$$

except when $\mathbf{x}^{(k)}$ is optimal.

From convexity we know that $\nabla f(\mathbf{x}^{(k)})^T(\mathbf{y} - \mathbf{x}^{(k)}) \geq 0$ implies $f(\mathbf{y}) \geq f(\mathbf{x}^{(k)})$, so the search direction in a descent method must satisfy

$$\nabla f(\mathbf{x}^{(k)})^T \Delta\mathbf{x}^{(k)} < 0,$$

i.e., it must make an acute angle with the negative gradient. We call such a direction a *descent direction* (for f, at $\mathbf{x}^{(k)}$).

Descent methods

Algorithm 5.1 *General descent method.*

given a starting point $\mathbf{x} \in \mathbf{dom} f$.

repeat

1. Determine a descent direction $\Delta\mathbf{x}$.
2. Line search. Choose a step size $t > 0$.
3. Update. $\mathbf{x} := \mathbf{x} + t\Delta\mathbf{x}$.

until stopping criterion is satisfied.

The stopping criterion is often of the form $\|\nabla f(\mathbf{x})\|_2 \leq \eta$, where η is small and positive.

Descent methods

- Exact line search

$$t = \operatorname*{argmin}_{s \geq 0} f(\mathbf{x} + s\Delta\mathbf{x}).$$

- Backtracking line search

Algorithm 9.2 *Backtracking line search.*
given a descent direction $\Delta\mathbf{x}$ for f at $\mathbf{x} \in \mathbf{dom} f, \alpha \in (0,1), \beta \in (0,1)$.
$t := 1$.
while $f(\mathbf{x} + t\Delta\mathbf{x}) > f(\mathbf{x}) + \alpha t \nabla f(\mathbf{x})^T \Delta\mathbf{x}, t := \beta t$.

Descent methods

- Backtracking line search

The line search is called *backtracking* because it starts with unit step size and then reduces it by the factor β until the stopping condition

$$f(\mathbf{x} + t\Delta\mathbf{x}) \leq f(\mathbf{x}) + \alpha t \nabla f(\mathbf{x})^T \Delta\mathbf{x}$$

holds. Since $\Delta\mathbf{x}$ is a descent direction, we have $\nabla f(\mathbf{x})^T \Delta\mathbf{x} < 0$, so for small enough t we have

$$f(\mathbf{x} + t\Delta\mathbf{x}) \approx f(\mathbf{x}) + t\nabla f(\mathbf{x})^T \Delta\mathbf{x} < f(\mathbf{x}) + \alpha t \nabla f(\mathbf{x})^T \Delta\mathbf{x},$$

which shows that the backtracking line search eventually terminates.

Descent methods

- Gradient descent method

A natural choice for the search direction is the negative gradient

$$\Delta \mathbf{x} = -\nabla f(\mathbf{x}).$$

The resulting algorithm is called the *gradient algorithm* or *gradient descent method*.

Algorithm 5.3 *Gradient descent method.*

given a starting point $\mathbf{x} \in \mathbf{dom} f$.

repeat

1. $\Delta \mathbf{x} := -\nabla f(\mathbf{x})$.
2. Line search. Choose step size t via exact or backtracking line search.
3. Update. $\mathbf{x} := \mathbf{x} + t\Delta \mathbf{x}$.

until stopping criterion is satisfied.

The stopping criterion is usually of the form $\|\nabla f(\mathbf{x})\|_2 \leq \eta$, where η is small and positive.

Descent methods

- Convergence analysis

Lighter notations: $\mathbf{x}^+ = \mathbf{x} + t\Delta\mathbf{x}$ for $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\Delta\mathbf{x}^{(k)}$, where $\Delta\mathbf{x} = -\nabla f(\mathbf{x})$. We assume f is strongly convex on S, so there are positive constants m and M such that $m\mathbf{I} \preceq \nabla^2 f(\mathbf{x}) \preceq M\mathbf{I}$ for all $\mathbf{x} \in S$. Define the function $\tilde{f} : \mathbb{R} \to \mathbb{R}$ by $\tilde{f}(t) = f(\mathbf{x} - t\nabla f(\mathbf{x}))$. From the inequality (6), with $\mathbf{y} = \mathbf{x} - t\nabla f(\mathbf{x})$, we obtain a quadratic upper bound on $\tilde{f}$:

$$\tilde{f}(t) \leq f(\mathbf{x}) - t\|\nabla f(\mathbf{x})\|_2^2 + \frac{Mt^2}{2}\|\nabla f(\mathbf{x})\|_2^2. \tag{8}$$

Descent methods

- Convergence analysis

Analysis for exact line search

Minimize over t both sides of the inequality (8). We have

$$f(\mathbf{x}^+) = \tilde{f}(t_{exact}) \leq f(\mathbf{x}) - \frac{1}{2M}\|\nabla(f(\mathbf{x}))\|_2^2.$$

Subtracting p^* from both sides, we get

$$f(\mathbf{x}^+) - p^* \leq f(\mathbf{x}) - p^* - \frac{1}{2M}\|\nabla f(\mathbf{x})\|_2^2.$$

We combine this with $\|\nabla f(\mathbf{x})\|_2^2 \geq 2m(f(\mathbf{x}) - p^*)$ (from (2)) to conclude

$$f(\mathbf{x}^+) - p^* \leq (1 - m/M)(f(\mathbf{x}) - p^*).$$

Applying this inequality recursively, we find that

linear convergence!
M/m: condition number

$$f(\mathbf{x}^{(k)}) - p^* \leq c^k(f(\mathbf{x}^{(0)}) - p^*), \text{ where } c = 1 - m/M < 1. \tag{9}$$

Descent methods

- Convergence analysis

Analysis for backtracking line search

From (6),

$$f(\mathbf{x}_k - t\nabla f(\mathbf{x}_k)) \leq f(\mathbf{x}_k) - t\|\nabla f(\mathbf{x}_k)\|^2 + \frac{Mt^2}{2}\|\nabla f(\mathbf{x}_k)\|^2. \tag{10}$$

By the choice of $t_k = \beta^k t_0$,

$$f(\mathbf{x}_k - t_k\nabla f(\mathbf{x}_k)) \leq f(\mathbf{x}_k) - \alpha t_k\|\nabla f(\mathbf{x}_k)\|^2, \tag{11}$$

$$f(\mathbf{x}_k - (t_k/\beta)\nabla f(\mathbf{x}_k)) > f(\mathbf{x}_k) - \alpha(t_k/\beta)\|\nabla f(\mathbf{x}_k)\|^2. \tag{12}$$

By (10),

$$f(\mathbf{x}_k - (t_k/\beta)\nabla f(\mathbf{x}_k)) \leq f(\mathbf{x}_k) - (t_k/\beta)\|\nabla f(\mathbf{x}_k)\|^2 + \frac{M}{2}(t_k/\beta)^2\|\nabla f(\mathbf{x}_k)\|^2. \tag{13}$$

Descent methods

- Convergence analysis

Analysis for backtracking line search

Together with (12), we obtain

$$-\alpha(t_k/\beta)\|\nabla f(\mathbf{x}_k)\|^2 \leq -(t_k/\beta)\|\nabla f(\mathbf{x}_k)\|^2 + \frac{M}{2}(t_k/\beta)^2\|\nabla f(\mathbf{x}_k)\|^2. \tag{14}$$

So

$$t_k \geq \frac{2(1-\alpha)\beta}{M}. \tag{15}$$

Plug in (11) and together with (2) to obtain

$$f(\mathbf{x}_{k+1}) - f^* \leq f(\mathbf{x}_k) - f^* - \frac{4\alpha(1-\alpha)\beta m}{M}(f(\mathbf{x}_k) - f^*). \tag{16}$$

Finally,

$$f(\mathbf{x}_{k+1}) - f^* \leq \left[1 - \frac{4\alpha(1-\alpha)\beta m}{M}\right](f(\mathbf{x}_k) - f^*). \tag{17}$$

Descent methods

- Examples

A quadratic problem in $\mathbb{R}^2$.

$$f(\mathbf{x}) = \frac{1}{2}(x_1^2 + \gamma x_2^2),$$

where $\gamma > 0$. Clearly, the optimal point is $\mathbf{x}^* = \mathbf{0}$, and the optimal value is 0. The Hessian of f is constant, and has eigenvalues 1 and γ, so the condition numbers of the sublevel sets of f are all exactly

$$\frac{\max\{1, \gamma\}}{\min\{1, \gamma\}} = \max\{\gamma, 1/\gamma\}.$$

Descent methods

- Examples

A nonquadratic problem in $\mathbb{R}^2$.

$$f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}.$$

We apply the gradient method with a backtracking line search, with $\alpha = 0.1$, $\beta = 0.7$.

Descent methods

- Examples

A problem in $\mathbb{R}^{100}$.

$$f(\mathbf{x}) = \mathbf{c}^T\mathbf{x} - \sum_{i=1}^{m} \log(b_i - \mathbf{a}_i^T\mathbf{x}),$$

with $m = 500$ terms and $n = 100$ variables. $\alpha = 0.1$ and $\beta = 0.5$

Descent methods

- Steepest descent method

The first-order Taylor approximation of $f(\mathbf{x}+\mathbf{v})$ around $\mathbf{x}$ is

$$f(\mathbf{x}+\mathbf{v}) \approx \hat{f}(\mathbf{x}+\mathbf{v}) = f(\mathbf{x}) + \boxed{\nabla f(\mathbf{x})^T \mathbf{v}}.$$

directional derivative

Normalized steepest descent direction (with respect to the norm $\|\cdot\|$):

$$\Delta \mathbf{x}_{nsd} = \underset{\mathbf{v}}{\operatorname{argmin}}\{\nabla f(\mathbf{x})^T \mathbf{v} | \|\mathbf{v}\| = 1\}.$$

It is also convenient to consider a steepest descent step $\Delta \mathbf{x}_{sd}$ that is unnormalized:

$$\Delta \mathbf{x}_{sd} = \|\nabla f(\mathbf{x})\|_* \Delta \mathbf{x}_{nsd},$$

$\Delta \mathbf{x}_{sd} = -\nabla f(\mathbf{x})$ if $\|\cdot\|$ is the Euclidean norm.

where $\|\cdot\|_*$ denotes the dual norm. Note that for the steepest descent step, we have

$$\nabla f(\mathbf{x})^T \Delta \mathbf{x}_{sd} = \|\nabla f(\mathbf{x})\|_* \nabla f(\mathbf{x})^T \Delta \mathbf{x}_{nsd} = -\|\nabla f(\mathbf{x})\|_*^2.$$

Descent methods

- Steepest descent method

Algorithm 5.4 *Steepest descent method.*

given a starting point $\mathbf{x} \in \mathbf{dom} f$.

repeat

1. Compute steepest descent direction $\Delta \mathbf{x}_{sd}$.
2. Line search. Choose t via backtracking or exact line search.
3. Update. $\mathbf{x} := \mathbf{x} + t\Delta \mathbf{x}_{sd}$.

until stopping criterion is satisfied.

Scale factors in the descent direction have no effect. So the unnormalized direction can be used.

Descent methods

- Steepest descent for Euclidean and quadratic norms

Steepest descent for Euclidean norm

If we take the norm $\|\cdot\|$ to be the Euclidean norm we find that the steepest descent direction is simply the negative gradient, *i.e.*, $\Delta\mathbf{x}_{sd} = -\nabla f(\mathbf{x})$. The steepest descent method for the Euclidean norm coincides with the gradient descent method.

Descent methods

- Steepest descent for Euclidean and quadratic norms

Steepest descent for quadratic norm

We consider the quadratic norm $\|\mathbf{z}\|_{\mathbf{P}} = (\mathbf{z}^T\mathbf{P}\mathbf{z})^{1/2} = \|\mathbf{P}^{1/2}\mathbf{z}\|_2$, where $\mathbf{P} \in \mathbb{S}^n_{++}$. The normalized steepest descent direction is given by

$$\Delta\mathbf{x}_{nsd} = -(\nabla f(\mathbf{x})^T\mathbf{P}^{-1}\nabla f(\mathbf{x}))^{-1/2}\mathbf{P}^{-1}\nabla f(\mathbf{x}).$$

The dual norm is given by $\|\mathbf{z}\|_* = \|\mathbf{P}^{-1/2}\mathbf{z}\|_2$, so the steepest descent step with respect to $\|\cdot\|_{\mathbf{P}}$ is given by

$$\Delta\mathbf{x}_{nsd} = -\mathbf{P}^{-1}\nabla f(\mathbf{x}).$$

Descent methods

- Steepest descent for Euclidean and quadratic norms

Interpretation via change of coordinates

Define $\bar{\mathbf{u}} = \mathbf{P}^{1/2}\mathbf{u}$, so we have $\|\mathbf{u}\|_{\mathbf{P}} = \|\bar{\mathbf{u}}\|_2$. The equivalent problem of minimizing

$$\bar{f}(\bar{\mathbf{u}}) = f(\mathbf{P}^{-1/2}\bar{\mathbf{u}}) = f(\mathbf{u}).$$

If we apply the gradient method to $\bar{f}$, the search direction at a point $\bar{\mathbf{x}}$ (which corresponds to the point $\mathbf{x} = \mathbf{P}^{-1/2}\bar{\mathbf{x}}$ for the original problem) is

$$\Delta\bar{\mathbf{x}} = -\nabla\bar{f}(\bar{\mathbf{x}}) = -\mathbf{P}^{-1/2}\nabla f(\mathbf{P}^{-1/2}\bar{\mathbf{x}}) = -\mathbf{P}^{-1/2}\nabla f(\mathbf{x}).$$

This gradient search direction corresponds to the direction

$$\Delta\mathbf{x} = \mathbf{P}^{-1/2}(-\mathbf{P}^{-1/2}\nabla f(\mathbf{x})) = -\mathbf{P}^{-1}\nabla f(\mathbf{x}),$$

for the original variable $\mathbf{x}$. So the steepest descent method in the norm $\|\cdot\|_{\mathbf{P}}$ can be regarded as the gradient method applied to the problem after the change of coordinates $\bar{\mathbf{x}} = \mathbf{P}^{1/2}\mathbf{x}$.

Descent methods

- Steepest descent for l_1-norm

$$\Delta \mathbf{x}_{nsd} = \underset{\mathbf{v}}{\operatorname{argmin}}\{\nabla f(\mathbf{x})^T \mathbf{v} \mid \|\mathbf{v}\|_1 \leq 1\}$$

can be easily characterized. Let i be any index for which $\|\nabla f(\mathbf{x})\|_\infty = |(\nabla f(\mathbf{x}))_i|$. Then $\Delta \mathbf{x}_{nsd}$ for the l_1-norm is given by

$$\Delta \mathbf{x}_{nsd} = -\operatorname{sign}\left(\frac{\partial f(\mathbf{x})}{\partial x_i}\right) \mathbf{e}_i.$$

An unnormalized steepest descent direction is then

$$\Delta \mathbf{x}_{sd} = \Delta \mathbf{x}_{nsd} \|\nabla f(\mathbf{x})\|_\infty = -\frac{\partial f(\mathbf{x})}{\partial x_i} \mathbf{e}_i.$$

not exactly!

Thus, the normalized steepest descent direction in l_1-norm can always be chosen along which the approximate decrease in f is greatest.

The algorithm is sometimes called a *coordinate-descent* algorithm. This can greatly simplify, or even trivialize, the line search.

Descent methods

- Steepest descent for l_1-norm

Example: *Frobenius norm scaling.*

$$\min_{\mathbf{d}} \sum_{i,j=1}^{n} M_{ij}^2 d_i^2 / d_j^2.$$

Using the change of variables $x_i = 2 \log d_i$,

$$\min_{\mathbf{x}} f(\mathbf{x}) = \sum_{i,j=1}^{n} M_{ij}^2 e^{x_i - x_j}.$$ Convex!

Keeping all components except the kth fixed, we can write $f(\mathbf{x}) = \alpha_k + \beta_k e^{-x_k} + \gamma_k e^{x_k}$, where

$$\alpha_k = M_{kk}^2 + \sum_{i,j \neq k} M_{ij}^2 e^{x_i - x_j}, \beta_k = \sum_{i \neq k} M_{ij}^2 e^{x_i}, \gamma_k = \sum_{j \neq k} M_{ij}^2 e^{-x_j}.$$

The minimum of $f(\mathbf{x})$, as a function of x_k, is obtained for $x_k = \log(\beta_k / \gamma_k)/2$.

Descent methods

- Steepest descent for l_1-norm

The l_1-steepest descent algorithm with exact line search consists of repeating the following steps.

1. Compute the gradient $(\nabla f(\mathbf{x}))_i = \beta_i e^{-x_i} + \gamma_i e^{x_i}, i = 1, ..., n.$
2. Select a largest (in absolute value) component of $\nabla f(\mathbf{x})$: $|\nabla f(\mathbf{x})|_k = \|\nabla f(\mathbf{x})\|_\infty$.
3. Minimize f over the scalar variable x_k, by setting $x_k = \log(\beta_k/\gamma_k)/2$.

Descent methods

- Choice of norm for steepest descent

Prescription for choosing $\mathbf{P}$: It should be chosen so that the sublevel sets of f, transformed by $\mathbf{P}^{-1/2}$, are well conditioned. For example if an approximation $\hat{\mathbf{H}}$ of the Hessian at the optimal point $\mathbf{H}(\mathbf{x}^*)$ were known, a very good choice of $\mathbf{P}$ would be $\mathbf{P} = \hat{\mathbf{H}}$, since the Hessian of $\bar{f}$ at the optimum is then

$$\hat{\mathbf{H}}^{-1/2}\nabla^2 f(\mathbf{x}^*)\hat{\mathbf{H}}^{-1/2} \approx \mathbf{I},$$

and so is likely to have a low condition number. This same idea can be described without a change of coordinates. Saying that a sublevel set has low condition number after the change of coordinates $\hat{\mathbf{x}} = \mathbf{P}^{1/2}\mathbf{x}$ is the same as saying that the ellipsoid

$$\varepsilon = \{\mathbf{x} | \mathbf{x}^T\mathbf{P}\mathbf{x} \leq 1\}$$

approximates the shape of the sublevel set.

Descent methods

- Examples

$f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$.

$$\mathbf{P}_1 = \begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix}, \mathbf{P}_2 = \begin{bmatrix} 8 & 0 \\ 0 & 2 \end{bmatrix}.$$

In both cases we use a backtracking line search with $\alpha = 0.1$ and $\beta = 0.7$.

Descent methods

- Examples

Newton's method

$$f(\mathbf{x}) \approx f(\mathbf{x}^{(k)}) + \mathbf{g}^{(k)T}(\mathbf{x} - \mathbf{x}^{(k)}) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^{(k)})^T F(\mathbf{x}^{(k)})(\mathbf{x} - \mathbf{x}^{(k)}) \triangleq q(\mathbf{x}),$$

where, for simplicity, we use the notation $\mathbf{g}^{(k)} = \nabla f(\mathbf{x}^{(k)})$. Applying the First-Order Necessary Condition (FONC) to q yields

$$\mathbf{0} = \nabla q(\mathbf{x}) = \mathbf{g}^{(k)} + \mathbf{F}(\mathbf{x}^{(k)})(\mathbf{x} - \mathbf{x}^{(k)}).$$

If $\mathbf{F}(\mathbf{x}^{(k)}) \succ \mathbf{0}$, then q achieves a minimum at

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{F}(\mathbf{x}^{(k)})^{-1}\mathbf{g}^{(k)}.$$

t

different from Boyd's book, which is called *damped Newton method*

Newton's Method

- Analysis of Newton's Method

1. There is no guarantee that Newton's algorithm heads in the direction of decreasing values of the objective function if $\mathbf{F}(\mathbf{x}^{(k)})$ is not positive definite.

2. Even if $\mathbf{F}(\mathbf{x}^{(k)}) \succ 0$, Newton's method may not be a descent method; that is, it is possible that $f(\mathbf{x}^{(k+1)}) \geq f(\mathbf{x}^{(k)})$.

3. Newton's method has superior convergence properties when the starting point is near the solution.

Newton's Method

- Analysis of Newton's Method

Theorem 1. *Suppose that $f \in \mathcal{C}^3$, and $\mathbf{x}^* \in \mathbb{R}^n$ is a point such that $\nabla f(\mathbf{x}^*) = \mathbf{0}$ and $\mathbf{F}(\mathbf{x}^*)$ is invertible. Then, for all $\mathbf{x}^{(0)}$ sufficiently close to $\mathbf{x}^*$, Newton's method is well defined for all k, and converges to $\mathbf{x}^*$ with order of convergence at least 2.*

Warning: In Theorem 1, we did not state that $\mathbf{x}^*$ is a local minimizer. For example, if $\mathbf{x}^*$ is a local maximizer then provided that $f \in \mathbb{C}^3$ and $\mathbf{F}(\mathbf{x}^*)$ is invertible, Newton's method would converge to $\mathbf{x}^*$ if we start close enough to it.

Newton's Method

- Analysis of Newton's Method

Theorem 2. *Suppose that $f \in \mathcal{C}^2$ is strongly convex, with Lipschitz continuous second order derivative:*

$$\nabla^2 f(\mathbf{x}) \succeq m\mathbf{I}, \quad \|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})\|_F \leq L\|\mathbf{x} - \mathbf{y}\|_2,$$

and $\mathbf{x}^* \in \mathbb{R}^n$ is a local minimizer. Then, for all $\mathbf{x}^{(0)}$ sufficiently close to $\mathbf{x}^*$, Newton's method converges to $\mathbf{x}^*$ with order of convergence at least 2.

$$\begin{aligned}
\|\nabla f(\mathbf{x}^+)\|_2 &= \|\nabla f(\mathbf{x} + \Delta\mathbf{x}_{nt}) - (\nabla f(\mathbf{x}) + \nabla^2 f(\mathbf{x})\Delta\mathbf{x}_{nt})\|_2 \\
&= \left\| \int_0^1 (\nabla^2 f(\mathbf{x} + t\Delta\mathbf{x}_{nt}) - \nabla^2 f(\mathbf{x}))\Delta\mathbf{x}_{nt} \mathrm{d}t \right\|_2 \\
&\leq \frac{L}{2}\|\Delta\mathbf{x}_{nt}\|_2^2 = \frac{L}{2}\|\nabla^2 f(\mathbf{x})^{-1}\nabla f(\mathbf{x})\|_2^2 \leq \frac{L}{2m^2}\|\nabla f(\mathbf{x})\|_2^2.
\end{aligned}$$

Then use (2) to establish the rate w.r.t. $f(\mathbf{x}_k) - p^*$.

Newton's Method

- Damped Newton's Method

Theorem 3. *Let* $\{\mathbf{x}^{(k)}\}$ *be the sequence generated by Newton's method for minimizing a given objective function* $f(\mathbf{x})$*. If the Hessian* $\mathbf{F}(\mathbf{x}^{(k)}) \succ \mathbf{0}$ *and* $\mathbf{g}^{(k)} = \nabla f(\mathbf{x}^{(k)}) \neq \mathbf{0}$*, then the direction*

$$\mathbf{d}^{(k)} = -\mathbf{F}(\mathbf{x}^{(k)})^{-1}\mathbf{g}^{(k)}$$

from $\mathbf{x}^{(k)}$ *to* $\mathbf{x}^{(k+1)}$ *is a descent direction for* f *in the sense that*

$$\left\langle \mathbf{d}^{(k)}, \mathbf{g}^{(k)} \right\rangle < 0.$$

The above theorem motivates the following *Damped Newton's method*:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \mathbf{F}(\mathbf{x}^{(k)})^{-1}\mathbf{g}^{(k)},$$

where α_k is obtained by backtracking line search.

Note that α_k will eventually be chosen as 1.

Newton's Method

- Damped Newton's Method

Theorem 4. *There are numbers $\eta \in (0, m^2/L]$ and $\gamma > 0$ such that:*

- *If $\|\nabla f(\mathbf{x}^{(k)})\|_2 \geq \eta$, then*

$$f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)}) \leq -\gamma.$$

- *If $\|\nabla f(\mathbf{x}^{(k)})\|_2 < \eta$, then the backtracking line search selects $\alpha^{(k)} = 1$ and*

$$\frac{L}{2m^2}\|\nabla f(\mathbf{x}^{(k+1)})\|_2 \leq \left(\frac{L}{2m^2}\|\nabla f(\mathbf{x}^{(k)})\|_2\right)^2.$$

Newton's Method

- Drawbacks of Newton's method

1. Evaluation of $\mathbf{F}(\mathbf{x}^{(k)})$ for large n can be computationally expensive.
2. Solve the system of linear equations $\mathbf{F}(\mathbf{x}^{(k)})\mathbf{d}^{(k)} = -\mathbf{g}^{(k)}$ is also computationally expensive.
3. The Hessian matrix may not be positive definite.

Newton's Method

- Levenberg-Marquardt modification

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k(\mathbf{F}(\mathbf{x}^{(k)}) + \mu_k \mathbf{I})^{-1}\mathbf{g}^{(k)},$$

It is actually locally minimizing

$$f(\mathbf{x}) + \frac{\mu_k}{2}\|\mathbf{x} - \mathbf{x}^{(k)}\|^2.$$

Pure Newton's method: $\mu_k \to 0$.
Pure gradient method with small step size: $\mu_k \to \infty$.
In practice, we may start with a small value of μ_k, and then slowly increase it until we find that the iteration is descent, that is $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$.

Newton's Method

- Newton’s method for nonlinear least-squares

$$\min_{\mathbf{x}} \sum_{i=1}^{m} (r_i(\mathbf{x}))^2,$$

where $r_i : \mathbb{R} \to \mathbb{R}, i = 1, \cdots, m$, are given functions. This particular problem is called a *nonlinear least-squares problem.*

Newton's Method

- Newton's method for nonlinear least-squares

Defining $\mathbf{r} = [r_1, \cdots, r_m]^T$, we write the objective function as $f(\mathbf{x}) = \mathbf{r}(\mathbf{x})^T\mathbf{r}(\mathbf{x})$. Denote the Jacobian matrix of $\mathbf{r}$ by $\mathbf{J}(\mathbf{x})$, then

$$\nabla f(\mathbf{x}) = 2\mathbf{J}(\mathbf{x})^T\mathbf{r}(\mathbf{x}).$$

$$\begin{aligned}\nabla^2 f(\mathbf{x}) = 2\frac{\mathbf{J}^T\mathbf{r}}{\partial\mathbf{x}^T}(\mathbf{x}) &= 2\left(\mathbf{J}^T\mathbf{J} + \sum_{i=1}^{m}\frac{\frac{\partial r_i}{\partial\mathbf{x}}}{\partial\mathbf{x}^T}r_i\right)(\mathbf{x}) \\ &= 2\left(\mathbf{J}^T\mathbf{J} + \sum_{i=1}^{m} r_i\nabla^2 r_i\right)(\mathbf{x}) = 2\left(\mathbf{J}^T\mathbf{J} + \mathbf{S}\right)(\mathbf{x}),\end{aligned}$$

where $\mathbf{S} = \sum_{i=1}^{m} r_i\nabla^2 r_i$. Therefore, Newton's method is given by

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\mathbf{J}(\mathbf{x})^T\mathbf{J}(\mathbf{x}) + \mathbf{S}(\mathbf{x}))^{-1}\mathbf{J}(\mathbf{x})^T\mathbf{r}(\mathbf{x}).$$

no need for line search!

Newton's Method

- Gauss-Newton method for nonlinear least-squares

In some applications, the matrix $\mathbf{S}(\mathbf{x})$ involving the second derivatives of the function $\mathbf{r}$ can be ignored because its components are negligibly small. In this case Newton's algorithm reduces to what is commonly called the *Gauss-Newton* method:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}))^{-1} \mathbf{J}(\mathbf{x})^T \mathbf{r}(\mathbf{x}).$$

Newton's Method

- Newton's method for nonlinear least-squares

Example: Suppose we are given m measurements of a process at m points in time. Let $t_1, \cdots, t_m$ denote the measurement times, and $y_1, \cdots, y_m$ the measurement values. We wish to fit a sinusoid to the measurement data. The equation of the sinusoid is

$$y = A\sin(\omega t + \phi)$$

with appropriate choices of the parameters A, ω, and ϕ. We construct the objective function

$$\sum_{i=1}^{m} (y_i - A\sin(\omega t_i + \phi))^2.$$

Let $\mathbf{x} = [A, \omega, \phi]^T$ represent the vector of decision variables. We therefore obtain a nonlinear least-squares problem with

$$r_i(\mathbf{x}) = y_i - A\sin(\omega t_i + \phi).$$

Conjugate Direction Methods

- Introduction

The class of *conjugate direction methods* can be viewed as being intermediate between the method of steepest descent and Newton's method. The conjugate direction methods have the following properties:

1. Solve quadratics of n variables in n steps;
2. The usual implementation, the conjugate gradient algorithm, requires no Hessian matrix evaluations;
3. No matrix inversion and no storage of an $n \times n$ matrix required.

The conjugate direction methods typically perform better than the method of steepest descent, but not as well as Newton's method.

Conjugate Direction Methods

- Introduction

Definition 1. *Let* $\mathbf{Q}$ *be a real symmetric* $n \times n$ *matrix. The directions* $\mathbf{d}^{(0)}, \mathbf{d}^{(1)}, \cdots, \mathbf{d}^{(m)}$ *are* $\mathbf{Q}$*-conjugate if, for all* $i \neq j$*, we have* $\mathbf{d}^{(i)T}\mathbf{Q}\mathbf{d}^{(j)} = 0$.

Lemma 2. *Let* $\mathbf{Q}$ *be a symmetric positive definite* $n \times n$ *matrix. If the directions* $\mathbf{d}^{(0)}, \mathbf{d}^{(1)}, \cdots, \mathbf{d}^{(k)} \in \mathbb{R}^n, k \leq n-1,$ *are nonzero and* $\mathbf{Q}$*-conjugate, then they are linearly independent.*

$\mathbf{d}^{(0)}, \mathbf{d}^{(1)}, \cdots, \mathbf{d}^{(k)} \in \mathbb{R}^n, k \leq n-1,$ are nonzero and $\mathbf{Q}$-conjugate iff $\{\tilde{\mathbf{d}}^{(i)}\}$ are mutually orthogonal, where $\tilde{\mathbf{d}}^{(i)} = \mathbf{Q}^{1/2}\mathbf{d}^{(i)}$.

Conjugate Direction Methods

- The conjugate gradient algorithm for quadratic problems

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} - \mathbf{x}^T\mathbf{b}, \mathbf{x} \in \mathbb{R}^n,$$

where $\mathbf{Q} = \mathbf{Q}^T \succ \mathbf{0}$. Our first search direction from an initial point $\mathbf{x}^{(0)}$ is in the direction of steepest descent; that is

$$\mathbf{d}^{(0)} = -\mathbf{g}^{(0)}.$$

Thus,

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha_0\mathbf{d}^{(0)},$$

where

$$\alpha_0 = \arg\min_{\alpha>0} f(\mathbf{x}^{(0)} + \alpha\mathbf{d}^{(0)}) = -\frac{\mathbf{g}^{(0)T}\mathbf{d}^{(0)}}{\mathbf{d}^{(0)T}\mathbf{Q}\mathbf{d}^{(0)}}.$$

Conjugate Direction Methods

- The conjugate gradient algorithm for quadratic problems

In the next stage, we search in a direction $\mathbf{d}^{(1)}$ that is $\mathbf{Q}$-conjugate to $\mathbf{d}^{(0)}$. In general, at the $(k+1)$st step, we choose $\mathbf{d}^{(k+1)}$ to be a linear combination of $\mathbf{g}^{(k+1)}$ and $\mathbf{d}^{(k)}$. Specifically, we choose

$$\mathbf{d}^{(k+1)} = -\mathbf{g}^{(k+1)} + \beta_k \mathbf{d}^{(k)}, k = 0, 1, 2, \cdots$$

The coefficients $\beta_k, k = 1, 2, \cdots$ are chosen in such a way that $\mathbf{d}^{(k+1)}$ is $\mathbf{Q}$-conjugate to $\mathbf{d}^{(0)}, \mathbf{d}^{(1)}, \cdots, \mathbf{d}^{(k)}$. This is accomplished by choosing β_k to be

$$\beta_k = \frac{\mathbf{g}^{(k+1)T}\mathbf{Q}\mathbf{d}^{(k)}}{\mathbf{d}^{(k)T}\mathbf{Q}\mathbf{d}^{(k)}}.$$

Conjugate Direction Methods

- The conjugate gradient algorithm for quadratic problems

The conjugate gradient algorithm is summarized below.

1. Set $k := 0$; select the initial point $\mathbf{x}^{(0)}$.
2. $\mathbf{g}^{(0)} = \nabla f(\mathbf{x}^{(0)})$. If $\mathbf{g}^{(0)} = \mathbf{0}$, stop, else set $\mathbf{d}^{(0)} = -\mathbf{g}^{(0)}$.
3. $\alpha_k = -\frac{\mathbf{g}^{(k)T}\mathbf{d}^{(k)}}{\mathbf{d}^{(k)T}\mathbf{Q}\mathbf{d}^{(k)}}$.
4. $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$.
5. $\mathbf{g}^{(k+1)} = \nabla f(\mathbf{x}^{(k+1)})$. If $g^{(k+1)} = \mathbf{0}$, stop.
6. $\beta_k = \frac{\mathbf{g}^{(k+1)T}\mathbf{Q}\mathbf{d}^{(k)}}{\mathbf{d}^{(k)T}\mathbf{Q}\mathbf{d}^{(k)}}$.
7. $\mathbf{d}^{(k+1)} = -\mathbf{g}^{(k+1)} + \beta_k \mathbf{d}^{(k)}$.
8. Set $k := k + 1$; go to step 3.

Conjugate Direction Methods

- The conjugate gradient algorithm for quadratic problems

Lemma 1. *Let* $\mathbf{d}^{(0)}, \cdots, \mathbf{d}^{(n-1)}$ *be* $\mathbf{Q}$*-conjudate directions,* $\mathbf{g}^{(k)} = \mathbf{Q}\mathbf{x}^{(k)} - \mathbf{b}$, $\alpha_k = -\dfrac{\mathbf{g}^{(k)T}\mathbf{d}^{(k)}}{\mathbf{d}^{(k)T}\mathbf{Q}\mathbf{d}^{(k)}}$, $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$. *Then*

$$\mathbf{g}^{(k+1)T}\mathbf{d}^{(i)} = 0$$

for all $k, 0 \leq k \leq n-1$, *and* $0 \leq i \leq k$.

Proposition 1. *In the conjugate gradient algorithm, the directions* $\mathbf{d}^{(0)}, \mathbf{d}^{(1)}, \cdots, \mathbf{d}^{(n-1)}$ *are* $\mathbf{Q}$*-conjugate.*

Proof. By induction.

When there is no numerical error, the iteration of CG terminates when k=$n-1$.

Example: $\min_{\mathbf{x}} f(x_1, x_2, x_3) = \frac{3}{2}x_1^2 + 2x_2^2 + \frac{3}{2}x_3^2 + x_1x_3 + 2x_2x_3 - 3x_1 - x_3$.

Conjugate Direction Methods

- The conjugate gradient algorithm for non-quadratic problems

For a general nonlinear function the Hessian is a matrix that has to be reevaluated at each iteration of the algorithm. This can be computationally very expensive. Thus, an efficient implementation of the conjugate gradient algorithm that eliminates the Hessian evaluation at each step is desirable.

Observe that $\mathbf{Q}$ appears only in the computation of the scalars α_k and β_k. Because

$$\alpha_k = \arg\min_{\alpha>0} f(\mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)}),$$

the closed form formula for α_k in the algorithm can be replaced by a numerical line search procedure. Therefore, we only need to concern ourselves with the formula for β_k.

Conjugate Direction Methods

- The conjugate gradient algorithm for non-quadratic problems

Recall that in the quadratic case:

$$\beta_k = \frac{\mathbf{g}^{(k+1)T}\mathbf{Q}\mathbf{d}^{(k)}}{\mathbf{d}^{(k)T}\mathbf{Q}\mathbf{d}^{(k)}}.$$

Hessian-free

The *Hestenes-Stiefel* formula. It is based on replacing the term $\mathbf{Q}\mathbf{d}^{(k)}$ by the term $(\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)})/\alpha_k$.

The two terms are equal in the quadratic case, as we now show. $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$. Premultiplying both sides by $\mathbf{Q}$, and recognizing that $\mathbf{g}^{(k)} = \mathbf{Q}\mathbf{x}^{(k)} - \mathbf{b}$, we get $\mathbf{g}^{(k+1)} = \mathbf{g}^{(k)} + \alpha_k \mathbf{Q}\mathbf{d}^{(k)}$, which we can rewrite as $\mathbf{Q}\mathbf{d}^{(k)} = (\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)})/\alpha_k$.

Substituting this into the original equation for β_k gives

$$\beta_k = \frac{\mathbf{g}^{(k+1)T}[\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}]}{\mathbf{d}^{(k)T}[\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}]},$$

Conjugate Direction Methods

- The conjugate gradient algorithm for non-quadratic problems

The Polak-Ribiere formula. Starting from the Hestenes-Stiefel formula, we multiply out the denominator to get

$$\beta_k = \frac{\mathbf{g}^{(k+1)T}[\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}]}{\mathbf{d}^{(k)T}\mathbf{g}^{(k+1)} - \mathbf{d}^{(k)T}\mathbf{g}^{(k)}}. \tag{1}$$

By Lemma 1, $\mathbf{d}^{(k)T}\mathbf{g}^{(k+1)} = 0$. Also, since $\mathbf{d}^{(k)} = -\mathbf{g}^{(k)} + \beta_{k-1}\mathbf{d}^{(k-1)}$, and premultiplying this by $\mathbf{g}^{(k)T}$, we get

$$\mathbf{g}^{(k)T}\mathbf{d}^{(k)} = -\mathbf{g}^{(k)T}\mathbf{g}^{(k)} + \beta_{k-1}\mathbf{g}^{(k)T}\mathbf{d}^{(k-1)} = -\mathbf{g}^{(k)T}\mathbf{g}^{(k)},$$

where once again we used Lemma 1. Hence, we get

$$\beta_k = \frac{\mathbf{g}^{(k+1)T}[\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}]}{\mathbf{g}^{(k)T}\mathbf{g}^{(k)}}.$$

Conjugate Direction Methods

- The conjugate gradient algorithm for non-quadratic problems

The Fletcher-Reeves formula. Starting with the Polak-Ribiere formula, we multiply out the numerator to get

$$\beta_k = \frac{\mathbf{g}^{(k+1)T}\mathbf{g}^{(k+1)} - \mathbf{g}^{(k+1)T}\mathbf{g}^{(k)}}{\mathbf{g}^{(k)T}\mathbf{g}^{(k)}}.$$

We now use the fact that $\mathbf{g}^{(k+1)T}\mathbf{g}^{(k)} = 0$, which we get by using the equation

$$\mathbf{g}^{(k+1)T}\mathbf{d}^{(k)} = -\mathbf{g}^{(k+1)T}\mathbf{g}^{(k)} + \beta_{k-1}\mathbf{g}^{(k+1)T}\mathbf{d}^{(k-1)}$$

and applying Lemma 1. This leads to

$$\beta_k = \frac{\mathbf{g}^{(k+1)T}\mathbf{g}^{(k+1)}}{\mathbf{g}^{(k)T}\mathbf{g}^{(k)}},$$

Conjugate Direction Methods

- Discussions

1. Reinitialize the direction vector to the negative gradient after every few iterations (e.g., n or $n+1$)

2. If the line search is known to be inaccurate, the Hestenes-Stiefel formula for β_k is recommended.

3. The choice of which formula for β_k to use depends on the objective function.

Majorization Minimization

- Basic framework

$$\min_{\mathbf{x}} \; f(\mathbf{x}), \quad s.t. \; \mathbf{x} \in \mathcal{C}.$$

$$\Downarrow$$

$$\min_{\mathbf{x}} \; g_k(\mathbf{x}), \quad s.t. \; \mathbf{x} \in \mathcal{C}.$$

1. $f(\mathbf{x}) \leq g_k(\mathbf{x}), \forall \mathbf{x} \in \mathcal{C};$ globally majorant
2. $f(\mathbf{x}_k) = g_k(\mathbf{x}_k).$

$$\boxed{\mathbf{x}_{k+1} = \operatorname*{argmin}_{\mathbf{x}} g_k(\mathbf{x}) \Longrightarrow f(\mathbf{x}_{k+1}) \leq g_k(\mathbf{x}_{k+1}) \leq g_k(\mathbf{x}_k) = f(\mathbf{x}_k).}$$

Majorization Minimization

- How to choose the majorant function?

Lipschitz Gradient Surrogate:

$$\min_{\mathbf{x}} \; g_k(\mathbf{x}), \quad s.t. \; \mathbf{x} \in \mathcal{C}.$$

$$g_k(\mathbf{x}) = f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle + \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{x}_k\|^2.$$

$$\Downarrow$$

$$\mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{C}}(\mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)).$$

projected gradient descent

$$\Downarrow \; (\mathcal{C} = \mathbb{R}^n)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k).$$

gradient descent

Majorization Minimization

- How to choose the majorant function?
  - How to choose α?

$$f(\mathbf{x}_{k+1}) = f(\mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)) < f(\mathbf{x}_k).$$

locally majorant

$$\Downarrow$$

If not satisfied $\alpha \leftarrow \mu\alpha$. $(\mu \in (0,1))$

backtracking

If f is L-smooth, i.e., $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$, then we may choose

$$\alpha = L^{-1}.$$

globally majorant

$$f(\mathbf{x}) \leq f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle + \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{x}_k\|^2 \triangleq g_k(\mathbf{x}).$$

Majorization Minimization

- How to choose the majorant function?

Quadratic Surrogate:

$$g_k(\mathbf{x}) = f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T \mathbf{H}_k (\mathbf{x} - \mathbf{x}_k), \text{ where } \mathbf{H}_k \succ \nabla^2 f.$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_k \nabla f. \quad (\mathcal{C} = \mathbb{R}^n)$$

Newton's method

Majorization Minimization

- How to choose the majorant function?

$$\min_{\mathbf{x}} \; g_k(\mathbf{x}), \quad s.t. \; \mathbf{x} \in \mathcal{C}.$$

$$g_k(\mathbf{x}) = f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle + \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{x}_k\|^2.$$

asymptotic smoothness

$$g_k(\mathbf{x}) - f(\mathbf{x}) \text{ is smooth.}$$

$$g_k(\mathbf{x}) \geq f(\mathbf{x}), \; \forall \mathbf{x}$$

globally majorant

$$\lim_{k\to\infty} \nabla g_k(\mathbf{x}_k, \mathbf{d}) - \nabla f(\mathbf{x}_k; \mathbf{d}) = 0.$$

$$g_k(\mathbf{x}_{k+1}) \geq f(\mathbf{x}_{k+1}).$$

locally majorant

Relaxed Majorization Minimization

Robust Matrix Factorization: $\min_{\mathbf{U}\in\mathcal{C}_U, \mathbf{V}\in\mathcal{C}_V} \|\mathbf{W}\odot(\mathbf{M}-\mathbf{U}\mathbf{V}^T)\|_1 + R_u(\mathbf{U}) + R_v(\mathbf{V}).$

Majorization Minimization

- How to choose the majorant function?

$$\min_{\mathbf{x}} \; f(\mathbf{x}) \triangleq \tilde{f}(\boldsymbol{\theta}^T \mathbf{x}), \quad s.t. \; \mathbf{x} \in \mathcal{C}. \qquad \boxed{\tilde{f}(x) \text{ is convex}}$$

Jensen Surrogate:

$$g_k(\mathbf{x}) = \sum_{i=1}^{n} w_i \tilde{f}\left(\frac{\theta_i}{w_i}(x_i - x_{k,i}) + \boldsymbol{\theta}^T \mathbf{x}_k\right),$$

where $\mathbf{w} \in \mathbb{R}_+^n$, $\|\mathbf{w}\|_1$ and $w_i \neq 0$ whenever $\theta_i \neq 0$.

Majorization Minimization

- EM Algorithm

Maximum Likelihood estimation of parameters of a Gaussian Mixture Model. Given data $\mathbf{x}_i$, $i = 1, \cdots, m$.

$$\max_{\boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}} \sum_{i=1}^{m} \log \sum_{k=1}^{n} \theta_k \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k) \right).$$

Z: the index of component that a sample belongs to.

EM solves a Maximum Likelihood problem of the form:

$$\max_{\boldsymbol{\theta}} \log \int_{\mathbf{z}} p(\mathbf{x}, \mathbf{z} | \boldsymbol{\theta}) \mathrm{d}\, \mathbf{z},$$

$\boldsymbol{\theta}$: parameters of the probabilistic model we try to find
$\mathbf{x}$: observed variables
$\mathbf{z}$: unobserved variables

Majorization Minimization

- EM Algorithm

$$\max_{\boldsymbol{\theta}} \log \int_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) \mathrm{d}\ \mathbf{z} = \max_{\boldsymbol{\theta}} \log \int_{\mathbf{z}} q(\mathbf{z}) \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} \mathrm{d}\ \mathbf{z}$$

$$= \max_{\boldsymbol{\theta}} \log \mathbb{E}_{Z \sim q} \left[\frac{p(\mathbf{x}, Z|\boldsymbol{\theta})}{q(Z)} \right]$$

$$\geq \max_{\boldsymbol{\theta}} \mathbb{E}_{Z \sim q} \log \left[\frac{p(\mathbf{x}, Z|\boldsymbol{\theta})}{q(Z)} \right]$$

$$= \max_{\boldsymbol{\theta}} \left(\int_{\mathbf{z}} q(\mathbf{z}) \log p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) \mathrm{d}\ \mathbf{z} - \int_{\mathbf{x}} q(\mathbf{z}) \log q(\mathbf{z}) \mathrm{d}\ \mathbf{z} \right)$$

E step: Choose $q(\mathbf{z})$ to maximize the right hand side, which is

$$q(\mathbf{z}) = \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{\int_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) \mathrm{d}\ \mathbf{z}} = p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) \propto p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}).$$

M step: Solve $\max_{\boldsymbol{\theta}} \int_{\mathbf{z}} q(\mathbf{z}) \log p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) \mathrm{d}\ \mathbf{z}$.

Majorization Minimization

- How to choose the majorant function?

$$\min_{\mathbf{x}} \; f(\mathbf{x}) + h(\mathbf{x}), \quad s.t. \; \mathbf{x} \in \mathcal{C},$$

where f is convex and h is concave.

$$g_k(\mathbf{x}) = f(\mathbf{x}) + \langle \partial h(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle + h(\mathbf{x}_k),$$

convex concave procedure (CCCP)

where ∂h is a super-gradient of h.

low-rankness regularizer

$$\min_{\mathbf{X}} \boxed{\sum_{i=1}^{\min(m,n)} h(\sigma_i(\mathbf{X}))} + f(\mathbf{X}), \text{ where } h \text{ is concave on } \mathbb{R}_+.$$

$$h(\sigma_i) \leq h(\sigma_i^k) + w_i^k(\sigma_i - \sigma_i^k), \quad w_i^k \in \partial h(\sigma_i).$$

Majorization Minimization

- How to choose the majorant function?

$$\min_{\mathbf{x}} \; f(\mathbf{x}), \quad s.t. \; \mathbf{x} \in \mathcal{C}, \quad \text{where } f(\mathbf{x}) = \min_{\mathbf{y} \in \mathcal{Y}} h(\mathbf{x}, \mathbf{y}).$$

Variational surrogate: $g_k(\mathbf{x}) = h(\mathbf{x}, \mathbf{y}_k^*), \quad \text{where } \mathbf{y}_k^* = \operatorname*{argmin}_{\mathbf{y} \in \mathcal{Y}} h(\mathbf{x}_k, \mathbf{y}).$

Schatten-p norm: $\|\mathbf{X}\|_{S_p} = \left(\sum_i \sigma_i^p(\mathbf{X}) \right)^{1/p}$, low-rankness regularizer.

Theorem 1. *With compatiable dimensions and* $\frac{1}{p} = \sum_{i=1}^{I} \boxed{\frac{1}{p_i}}$:

$$\frac{1}{p} \|\mathbf{X}\|_{S_p}^p = \min_{\mathbf{X} = \Pi_{i=1}^{I} \mathbf{X}_i} \sum_{i=1}^{I} \frac{1}{p_i} \|\mathbf{X}_i\|_{S_{p_i}}^{p_i}.$$

If $0 < p < 1$, we can still choose $p_i \geq 1$.

Majorization Minimization

- How to choose the majorant function?

$$|x|^p = \min_{a>0} \frac{1}{c}(x^2/a + a^q),$$

half-quadratic

$$\text{where } q = \frac{p}{2-p}, c = q^{1/(q+1)} + q^{-q/(q+1)}, a^* = (x^2/q)^{1/(q+1)}.$$

l_p-norm, sparsity regularizer

$$\boxed{\|\mathbf{x}\|_p^p} \Longrightarrow \frac{1}{c}(\mathbf{x}^T\mathbf{A}_k^{-1}\mathbf{x} + \mathbf{1}^T\mathbf{A}_k^q\mathbf{1}) \Longrightarrow c^{-1}\mathbf{x}^T\mathbf{A}_k^{-1}\mathbf{x}$$

$$\Longrightarrow \frac{q}{q+1}\mathbf{x}^T \operatorname{Diag}(|\mathbf{x}_k|^{-2/(q+1)})\mathbf{x} \Longrightarrow \boxed{\frac{p}{2}\mathbf{x}^T \operatorname{Diag}(|\mathbf{x}_k|)^{p-2}\mathbf{x},}$$

$$\text{where } \mathbf{A}_k = \operatorname{Diag}(\mathbf{a}_k), a_{k,i} = (x_{k,i}^2/q)^{1/(q+1)}.$$

Iteratively Reweighted Least Squares (IRLS)

Majorization Minimization

- How to choose the majorant function?

$$\boxed{\|\mathbf{X}\|_{S_p}^p} = \|\boldsymbol{\sigma}(\mathbf{X})\|_p^p \Longrightarrow \frac{p}{2}\boldsymbol{\sigma}(\mathbf{X})^T \operatorname{Diag}(\boldsymbol{\sigma}(\mathbf{X}_k)^{p-2})\boldsymbol{\sigma}(\mathbf{X})$$

$$= \frac{p}{2}\operatorname{tr}(\operatorname{Diag}(\boldsymbol{\sigma}(\mathbf{X}))\operatorname{Diag}(\boldsymbol{\sigma}(\mathbf{X}_k)^{p-2})\operatorname{Diag}(\boldsymbol{\sigma}(\mathbf{X})))$$

$$= \frac{p}{2}\operatorname{tr}(\mathbf{X}^T(\mathbf{X}_k\mathbf{X}_k^T)^{p/2-1}\mathbf{X}).$$

Schatten-p norm, low-rankness regularizer

$$\boxed{\operatorname{tr}\left((\mathbf{X}\mathbf{X}^T)^{p/2}\right)} \Longrightarrow \operatorname{tr}\left((\mathbf{X}_k\mathbf{X}_k^T)^{p/2-1}\mathbf{X}\mathbf{X}^T\right) \Longrightarrow \operatorname{tr}\left(\mathbf{X}^T(\mathbf{X}_k\mathbf{X}_k^T)^{p/2-1}\mathbf{X}\right).$$

$$\operatorname{tr}\left((\mathbf{X}\mathbf{X}^T)^{p/2}\right) = \min_{\mathbf{A}\succ\mathbf{0}} \frac{1}{c}[\operatorname{tr}(\mathbf{A}^{-1}\mathbf{X}\mathbf{X}^T) + \operatorname{tr}(\mathbf{A}^q)].$$

$$\operatorname{tr}(\mathbf{A}^{-1}\mathbf{X}\mathbf{X}^T) + \operatorname{tr}(\mathbf{A}^q) \geq \sum_{i=1}^n \sigma_i(\mathbf{X}\mathbf{X}^T)\sigma_{n-i+1}(\mathbf{A}^{-1}) + \sum_{i=1}^n \sigma_i^q(\mathbf{A})$$

$$= \sum_{i=1}^n \frac{\sigma_i^2(\mathbf{X})}{\sigma_i(\mathbf{A})} + \sum_{i=1}^n \sigma_i^q(\mathbf{A})$$