

Descent methods

- Convergence analysis

Analysis for backtracking line search The backtracking exit condition,

$$\tilde{f}(t) \leq f(\mathbf{x}) - \alpha t \|\nabla f(\mathbf{x})\|_2^2,$$

is satisfied whenever $0 \leq t \leq 1/M$. First note that $0 \leq t \leq 1/M \implies -t + \frac{Mt^2}{2} \leq -t/2$. Using this result and the bound (8), we have, for $0 \leq t \leq 1/M$,

$$\begin{aligned}\tilde{f}(t) &\leq f(\mathbf{x}) - t \|\nabla f(\mathbf{x})\|_2^2 + \frac{Mt^2}{2} \|\nabla f(\mathbf{x})\|_2^2 \\ &\leq f(\mathbf{x}) - (t/2) \|\nabla f(\mathbf{x})\|_2^2 \leq f(\mathbf{x}) - \alpha t \|\nabla f(\mathbf{x})\|_2^2\end{aligned}$$

since $\alpha < 1/2$. Therefore the backtracking line search terminates either with $t = 1$ or with a value $t \geq \beta/M$. In the first case we have

$$f(\mathbf{x}^+) \leq f(\mathbf{x}) - \alpha \|\nabla f(\mathbf{x})\|_2^2,$$

and in the second case we have

$$f(\mathbf{x}^+) \leq f(\mathbf{x}) - (\beta\alpha/M) \|\nabla f(\mathbf{x})\|_2^2.$$

Descent methods

- Convergence analysis

Putting these together, we always have

$$f(\mathbf{x}^+) \leq f(\mathbf{x}) - \min\{\alpha, (\beta\alpha/M)\}\|\nabla f(\mathbf{x})\|_2^2.$$

We subtract p^* from both sides to get

$$f(\mathbf{x}^+) - p^* \leq f(\mathbf{x}) - p^* - \min\{\alpha, (\beta\alpha/M)\}\|\nabla f(\mathbf{x})\|_2^2,$$

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

$$f(\mathbf{x}^+) - p^* \leq (1 - \min\{2m\alpha, 2\beta\alpha m/M\})(f(\mathbf{x}) - p^*).$$

From this we conclude

$$f(\mathbf{x}^{(k)}) - p^* \leq c^k(f(\mathbf{x}^{(0)}) - p^*),$$

where $c = 1 - \min\{2m\alpha, 2\beta\alpha m/M\} < 1$.

Descent methods

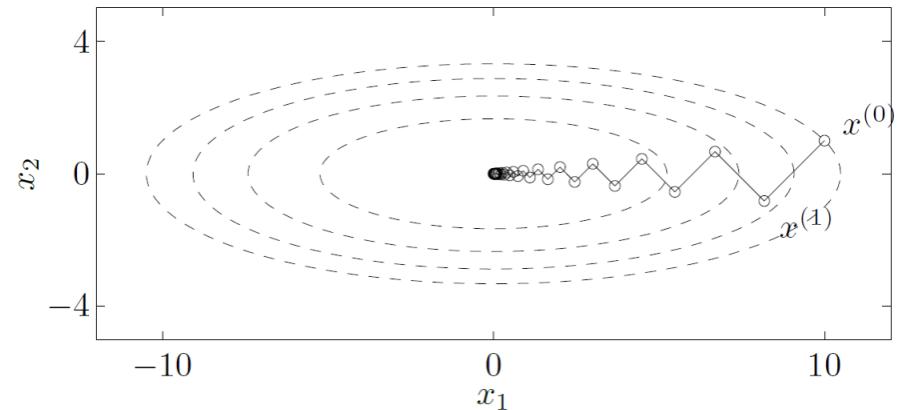
- Examples

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

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



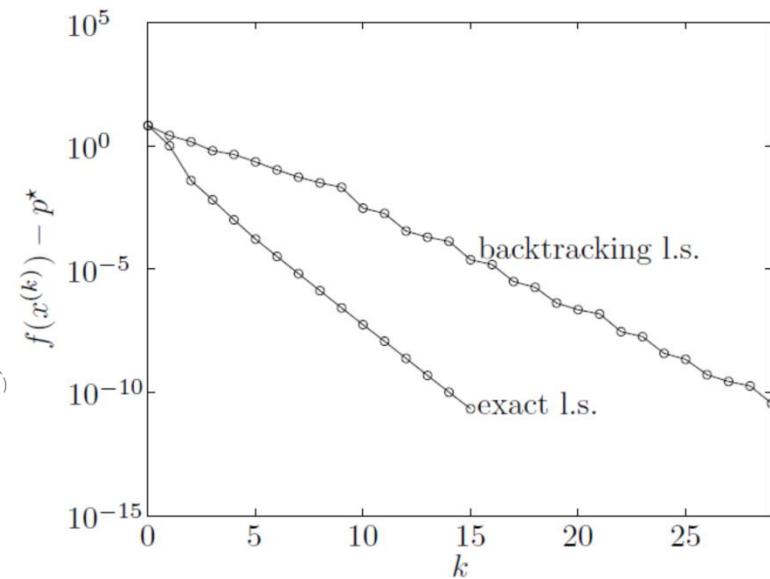
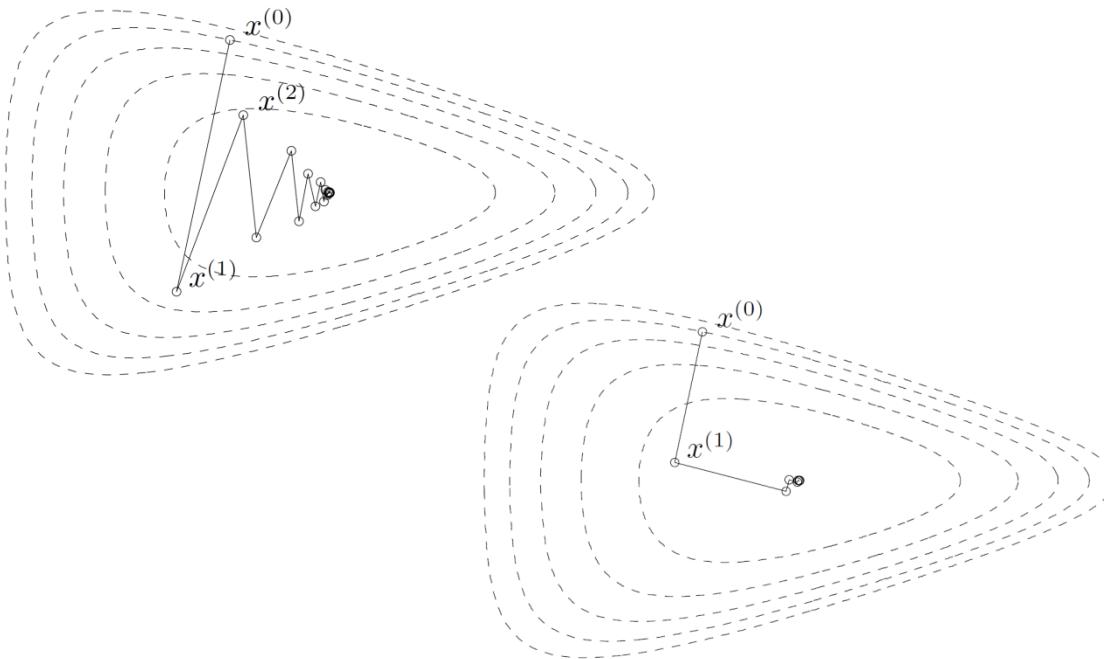
Descent methods

- 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 $\alpha = 0.1$, $\beta = 0.7$.



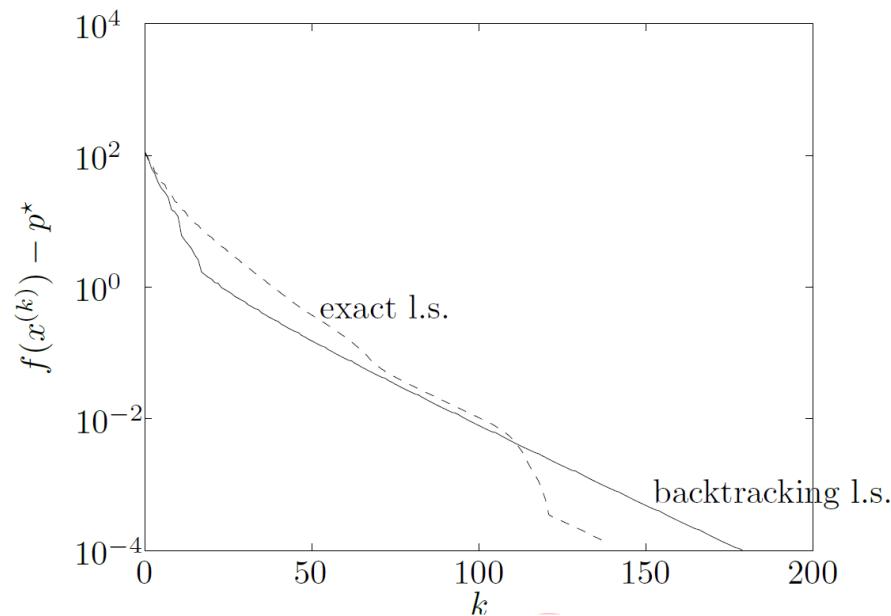
Descent methods

- 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

$$f(\mathbf{x} + \mathbf{v}) \approx \hat{f}(\mathbf{x} + \mathbf{v}) = f(\mathbf{x}) + \boxed{\nabla f(\mathbf{x})^T \mathbf{v}} \quad \begin{array}{|c|} \hline \text{directional derivative} \\ \hline \end{array}$$

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

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

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

$$\Delta \mathbf{x}_{sd} = \|\nabla f(\mathbf{x})\|_* \Delta \mathbf{x}_{nsd}, \quad \begin{array}{|c|} \hline \Delta \mathbf{x}_{sd} = -\nabla f(\mathbf{x}) \text{ if } \|\cdot\| \text{ is the Euclidean norm.} \\ \hline \end{array}$$

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 $\mathbf{x} \in \text{dom } f$.

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

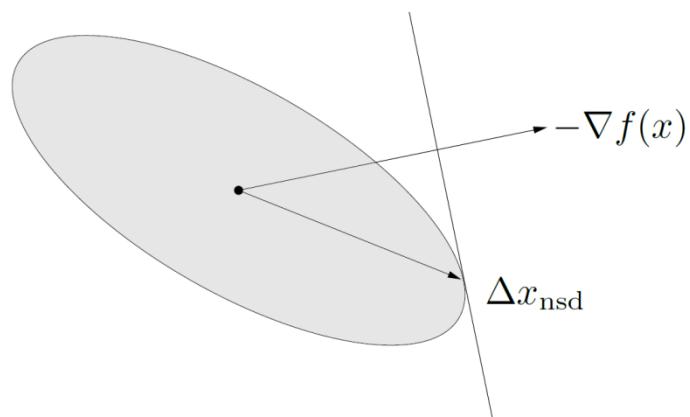
Steepest descent for quadratic norm

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 $x = \mathbf{P}^{-1/2}\bar{\mathbf{x}}$ 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

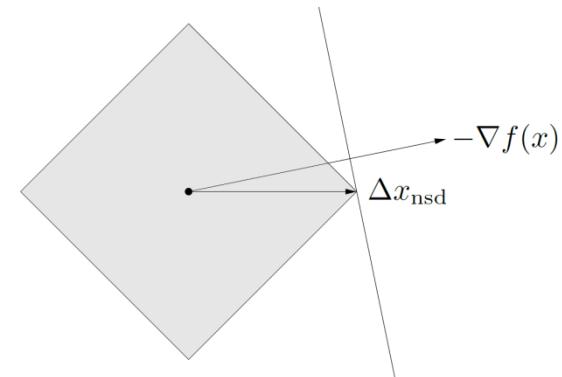
$$\Delta \mathbf{x}_{nsd} = \underset{\mathbf{v}}{\operatorname{argmin}} \{ \nabla f(\mathbf{x})^T \mathbf{v} | \|\mathbf{v}\|_1 \leq 1 \}$$

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 $x_i = 2 \log d_i$,

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

Keeping all components except the k th 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 $(\nabla f(\mathbf{x}))_i = \beta_i e^{-x_i} + \gamma_i e^{x_i}, i = 1, \dots, n.$
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 \tilde{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.

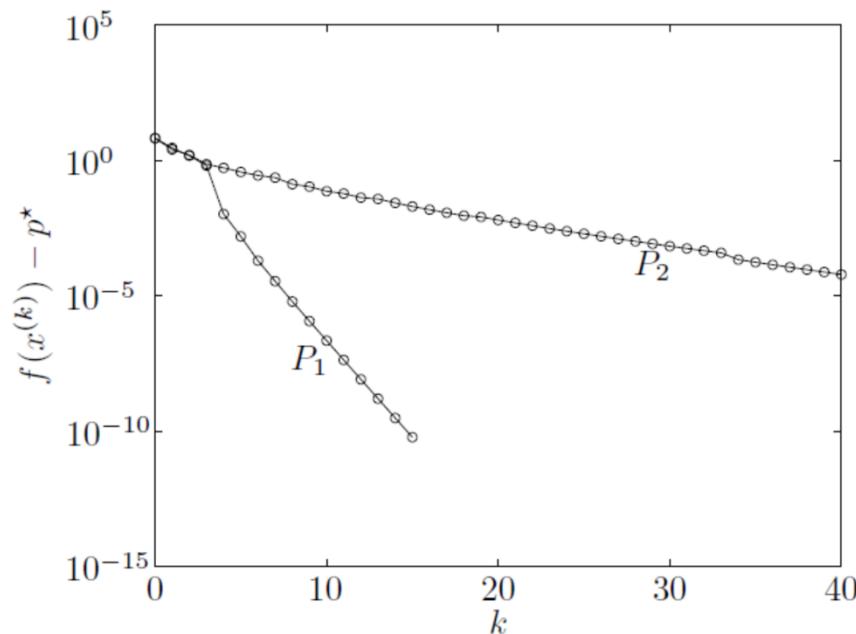
Descent methods

- Examples

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

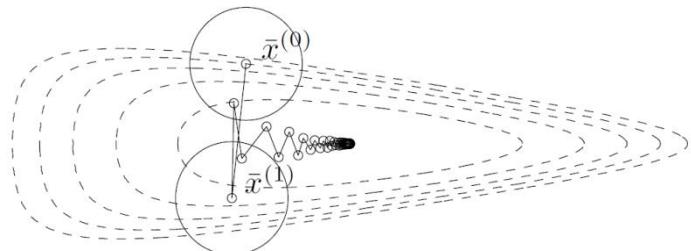
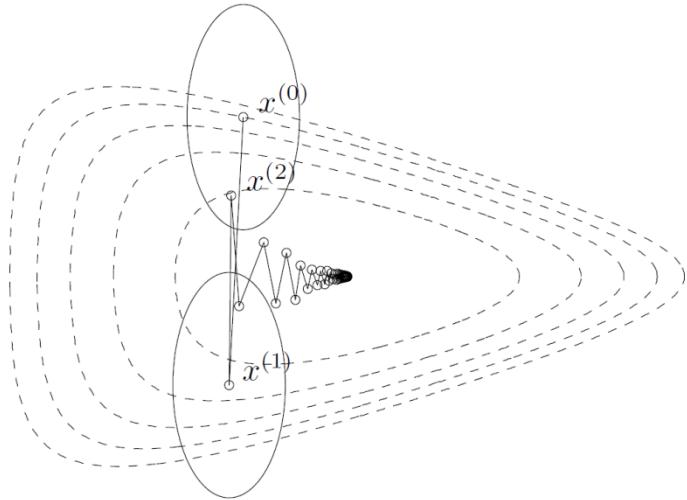
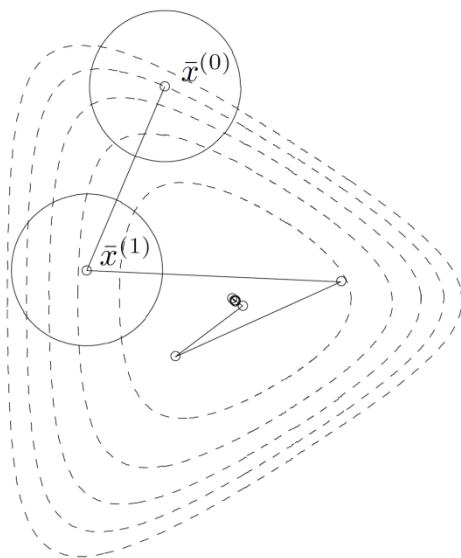
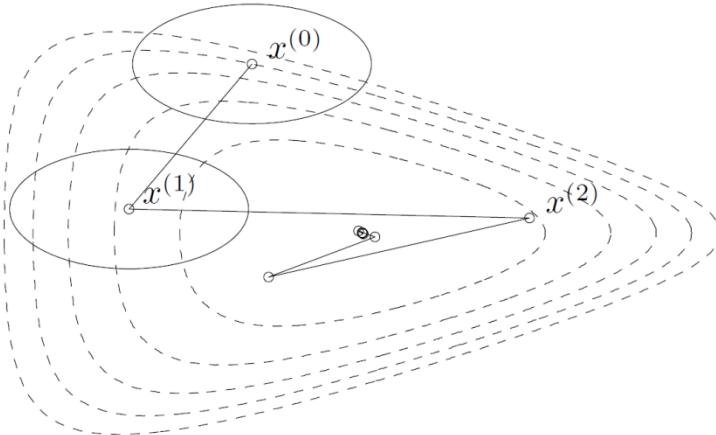
$$\mathbf{P}_1 = \begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix}, \mathbf{P}_2 = \begin{bmatrix} 8 & 0 \\ 0 & 2 \end{bmatrix}.$$

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



Descent methods

- Examples



Newton's method

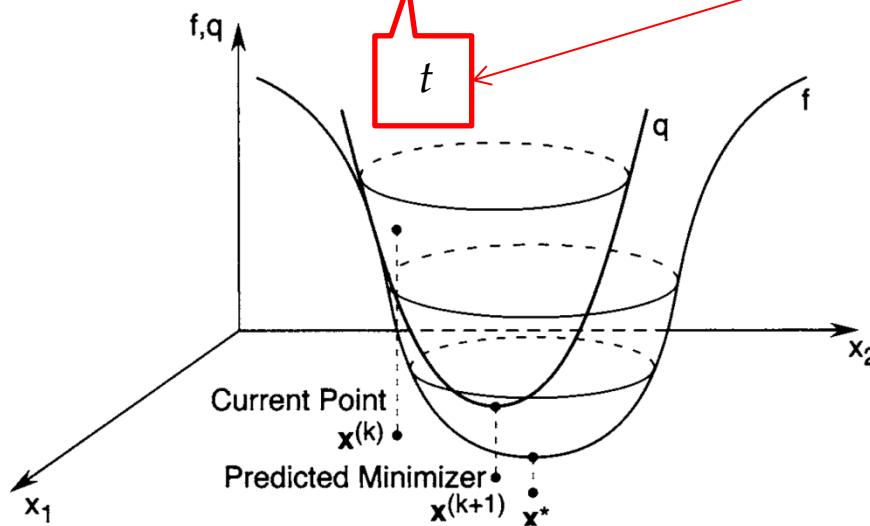
$$f(\mathbf{x}) \approx f(\mathbf{x}^{(k)}) + \mathbf{g}^{(k)T}(\mathbf{x} - \mathbf{x}^{(k)}) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^{(k)})^T F(\mathbf{x}^{(k)})(\mathbf{x} - \mathbf{x}^{(k)}) \triangleq q(\mathbf{x}),$$

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

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



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.

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

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

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

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: $\mu_k \rightarrow 0$.

Pure gradient method with small step size: $\mu_k \rightarrow \infty$.

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

- Gauss-Newton method for nonlinear least-squares

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

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

Newton's Method

- Gauss-Newton method for nonlinear least-squares

Defining $\mathbf{r} = [r_1, \dots, r_m]^T$, 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

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

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

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

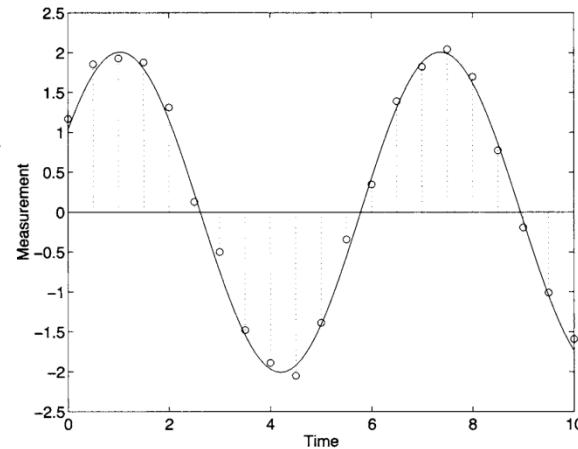
Example: Suppose we are given m measurements of a process at m points in time. Let t_1, \dots, t_m denote the measurement times, and y_1, \dots, y_m the measurement values. We wish to fit a sinusoid to the measurement data. The equation of the sinusoid is

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

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 $\mathbf{d}^{(0)}, \mathbf{d}^{(1)}, \dots, \mathbf{d}^{(m)}$ 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 $\mathbf{d}^{(0)}, \mathbf{d}^{(1)}, \dots, \mathbf{d}^{(k)} \in \mathbb{R}^n, k \leq n - 1$, are nonzero and \mathbf{Q} -conjugate, then they are linearly independent.

$\mathbf{d}^{(0)}, \mathbf{d}^{(1)}, \dots, \mathbf{d}^{(k)} \in \mathbb{R}^n, k \leq n - 1$, 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

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

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

$$\beta_k = \frac{\mathbf{g}^{(k+1)T} \mathbf{Q} \mathbf{d}^{(0)}}{\mathbf{d}^{(k)T} \mathbf{Q} \mathbf{d}^{(k)}}.$$

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 $\mathbf{g}^{(k+1)} = \mathbf{0}$, stop.
6. $\beta_k = \frac{\mathbf{g}^{(k+1)T} \mathbf{Q} \mathbf{d}^{(0)}}{\mathbf{d}^{(k)T} \mathbf{Q} \mathbf{d}^{(k)}}$.
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

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

Proof. By induction.

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

Lemma 1.

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

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

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

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

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.