Optimization in Machine Learning: Lecture 6

Solving Algorithm for Non-constrained Problems

by Xiaolin Huang xiaolinhuang@sjtu.edu.cn SEIEE 2-429

Institute of Image Processing and Pattern Recognition

http://www.pami.sjtu.edu.cn/

上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

目录 Contents

1 Optimality Condition

2 Gradient Descent Algorithm

3 Newton Method

4 Nesterov Acceleration

目录 Contents

1 Optimality Condition

2 Gradient Descent Algorithm

3 Newton Method

4 Nesterov Acceleration

Starting from Least Squares

- unconstrained minimization

$$\min_{x} f(x)$$

- **basic assumption**
  - f is convex, twice continuously differentiable
  - the optimal value $p^* = \inf f(x)$ is attained

- **example: Least Squares**

$$\min_{x} \sum_{i=1}^{m} \left(a_i^\top x - y_i\right)^2$$

$$\min_{x} \|Ax - Y\|_2^2 = \min_{x} \ (Ax - Y)^\top (Ax - Y)$$

Optimality Condition

- unconstrained minimization

$$\min_x f(x)$$

- **optimality condition**

$$\nabla f(x) = 0$$

- **example: Least Squares**

$$\nabla \|Ax - Y\|_2^2 = \nabla (Ax - Y)^\top (Ax - Y) \ = 2A^\top (Ax - Y) = 0$$

$$\Downarrow$$

$$x^* = (A^\top A)^{-1} A^\top Y$$

Pseudo Inverse

- unconstrained minimization

$$\min_x f(x)$$

- **optimality condition**

$$\nabla f(x) = 0$$

- **example: Least Squares**

$$\nabla \|Ax - Y\|_2^2 = \nabla (Ax - Y)^\top (Ax - Y) \ = 2A^\top (Ax - Y) = 0$$

$$\Downarrow$$

$$x^* = (A^\top A)^{-1} A^\top Y$$

pseudo inverse

$$A(A^\top A)^{-1} A^\top = AA^{-1} (A^\top)^{-1} A^\top = \mathbf{I}$$

$$\Downarrow$$

$$(A^\top A)^{-1} A^\top = A^{-1}$$

pseudo inverse = inverse?

Pseudo Inverse

- unconstrained minimization

$$\min_x f(x)$$

- **optimality condition**

$$\nabla f(x) = 0$$

- **example: Ridge Regression** $\min_x \|Ax - Y\|_2^2 + \lambda\|x\|_2^2$

$$\nabla\|Ax - Y\|_2^2 + \lambda\|x\|_2^2 = \nabla(Ax - Y)^\top(Ax - Y) + \lambda x^\top x = 2A^\top(Ax - Y) + 2\lambda x = 0$$

$$\Downarrow$$

$$x^* = (\lambda\mathbf{I} + A^\top A)^{-1}A^\top Y$$

- ill-posed problem
- keep optimization property

Iterative Reweighted Least Squares

- unconstrained minimization

$$\min_{x} f(x)$$

- **optimality condition**

$$\nabla f(x) = 0$$

  - analytical solution when the equation is linear
  - solving nonlinear equations > optimization

- **I**teratively **R**eweighted **L**east **S**quares

$$\min_{x} \sum_{i=1}^{m} p_i^k (a_i^\top x - y_i)^2 \quad \Longrightarrow \quad x^{k+1} = \left(A^\top P^k A\right)^{-1} A^\top P^k Y$$

  - approximate the function by a quadratic function
  - link to Newton's method

Iterative Reweighted Least Squares

- unconstrained minimization

$$\min_x f(x)$$

- **optimality condition**

$$\nabla f(x) = 0$$

  - analytical solution when the equation is linear
  - solving nonlinear equations > optimization

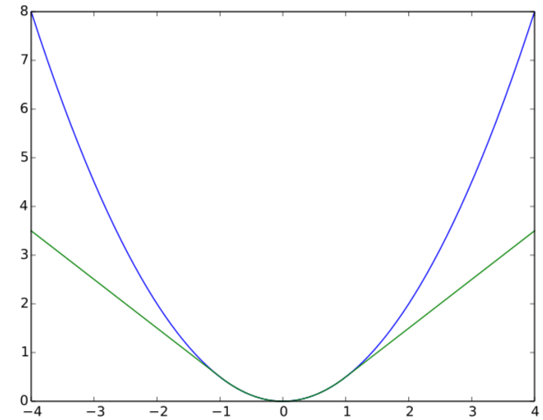
- **I**teratively **R**eweighted **L**east **S**quares

$$L_{\text{huber}}(r) = \begin{cases} r^2/2 & \text{if } |r| < c \\ c(|r| - c/2) & \text{if } |r| \geq c \end{cases}$$

example: Huber loss optimization

$$\min_x \quad \sum_{i=1}^{m} p_i^k (a_i^\top x - y_i)^2$$

  - solution update $\quad x^{k+1} = (A^\top P^k A)^{-1} A^\top P^k Y$
  - weight update $\quad p_i^{k+1} = \begin{cases} 1 & \text{if } |r_i| < c \\ c/r_i & \text{if } |r_i| \geq c \end{cases}$



Iterative Reweighted Least Squares

- unconstrained minimization

$$\min_{x} f(x)$$

- **optimality condition**

$$\nabla f(x) = 0$$

- **I**teratively **R**eweighted **L**east **S**quares

  the convergence should be carefully checked.

  especially for non-smooth problem, e.g., in compressive sensing

$$\min_{x} \|Ax - Y\|_2^2 + 2\lambda\|x\|_1$$

Ingrid Daubechies, et al. Iteratively reweighted least squares minimization for sparse recovery, *Communications on Pure and Applied Mathematics*, 2009

目录 Contents

1 Optimality Condition

2 **Gradient Descent Algorithm**

3 Newton Method

4 Nesterov Acceleration

Large Scale

- pseudo inverse $x^{*}=(A^{\top}A)^{-1}A^{\top}Y$
  - $A^{\top}A$ is an $n\times n$ matrix
  - inverse operation has complexity $O(n^{3})$
  - it is impossible for modern big data which may have million features

- finding an analytical solution does not mean the problem is solved
- analytical solution could help if it could be efficiently calculated

problem size	possible operations
small size	any operations
medium size	matrix inverse A^{-1}
large size	multiplication Ax
huge size	addition $x+y$

Yurii Nesterov

Descent Method

$$\min \quad f(x)$$

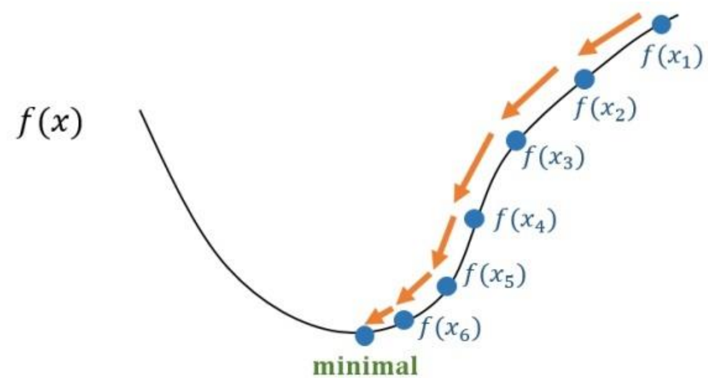
- to produce a sequence of points $x^{(k)}$ to approach the optimum

$$x^{(k+1)} = x^{(k)} + t^{(k)} \Delta x^{(k)} \quad \text{with} \quad f\left(x^{(k+1)}\right) < f\left(x^{(k)}\right)$$

General descent method.

given a starting point $x \in \mathbf{dom}\, f$.
repeat
1. Determine a descent direction Δx.
2. *Line search.* Choose a step size $t > 0$.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.



https://zhuanlan.zhihu.com/p/36564434

Descent Method

$$\min \quad f(x)$$

- to produce a sequence of points $x^{(k)}$ to approach the optimum

$$x^{(k+1)} = x^{(k)} + t^{(k)}\Delta x^{(k)} \quad \text{with} \quad f\left(x^{(k+1)}\right) < f\left(x^{(k)}\right)$$

General descent method.

given a starting point $x \in \textbf{dom}\, f$.
repeat
1. Determine a descent direction Δx.
2. *Line search.* Choose a step size $t > 0$.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

- staring point $x^{(0)}$
- descent direction $\Delta x^{(k)}$
- step size /step length/ learning rate $t^{(k)}$

Line Search

- when the direction $\Delta x^{(k)}$ is found, the update

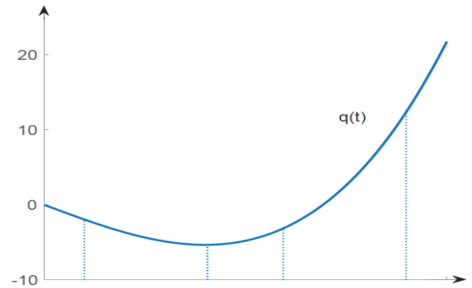
$$x^{(k+1)} = x^{(k)} + t^{(k)}\Delta x^{(k)}$$

becomes a univariate problem to find the step length.

- exact line search

$$t = \text{argmin}_t\, f(x + t\Delta x) \triangleq q(t)$$

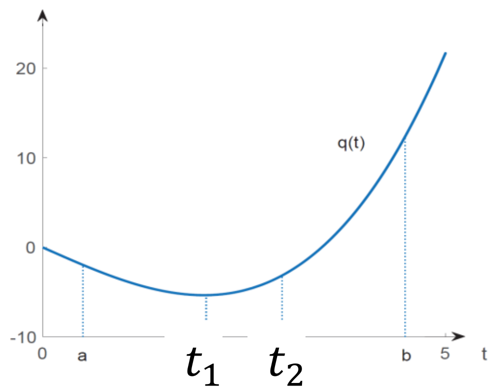
  - the basic idea is to use bisection to compress the interval
  - the key point is how to choose the breakpoint and how to judge the next interval

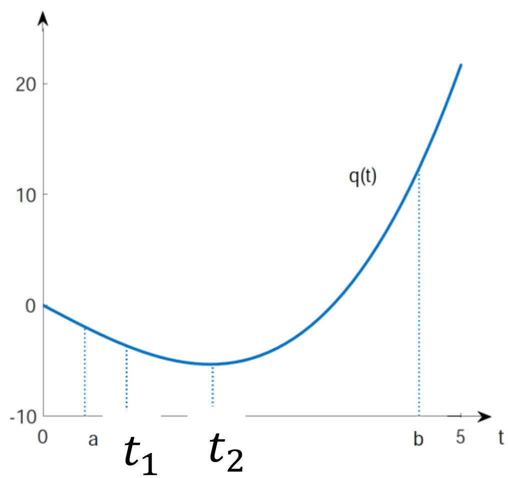


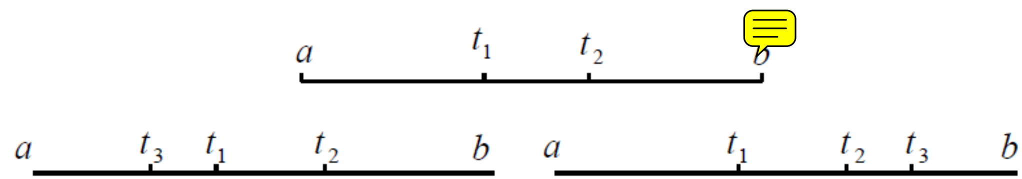
Line Search: by Function Value

$$t = \text{argmin}_t\, f(x + t\Delta x) \triangleq q(t)$$

- using at least two points, we can know where is the optimum







to have a constant compressive ratio c, the break point should satisfy

$$\frac{t_2 - a}{b - a} = \frac{b - t_1}{b - a} = c \qquad \frac{t_1 - a}{t_2 - a} = \frac{b - t_2}{b - t_1} = c$$

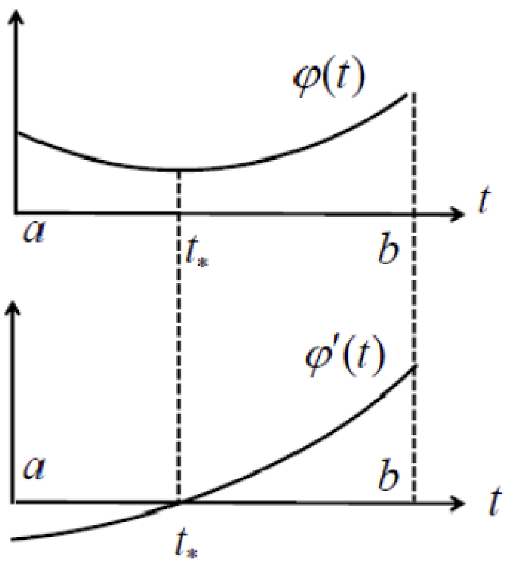
$$c = \frac{1}{2}\left(\sqrt{5} - 1\right) \approx 0.618$$

golden-section

Line Search: by Gradient

$$t = \mathrm{argmin}_t\, f(x + t\Delta x) \triangleq q(t)$$

- if we know the gradient, we can improve the compressive ratio to $c = 0.5$



$$f'(t_*) = 0$$

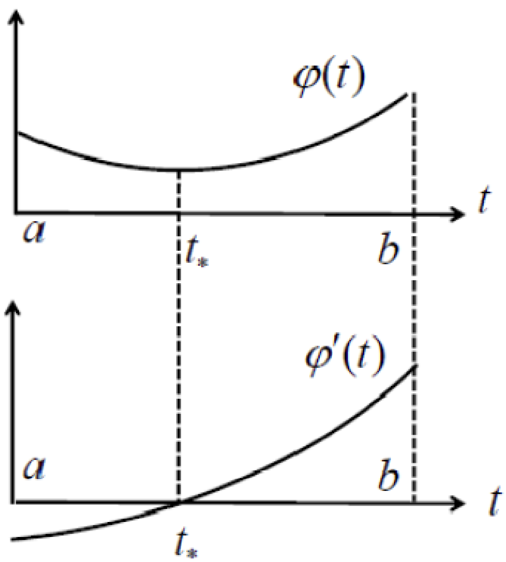
$$\text{If } f'(t)f(a) > 0, \text{ then } a \leftarrow t$$

$$\text{If } f'(t)f(b) > 0, \text{ then } b \leftarrow t$$

Line Search: by Gradient

$$t = \operatorname{argmin}_t f(x + t\Delta x) \triangleq q(t)$$

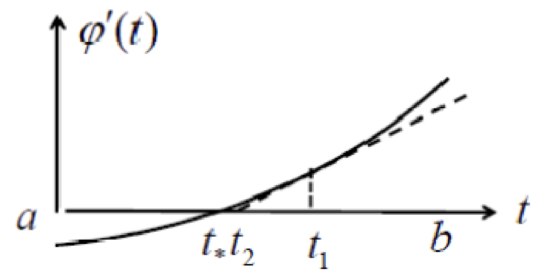
- if we know the gradient, we can improve the compressive ratio to $c = 0.5$



$$f'(t_*) = 0$$

$$\text{If } f'(t)f(a) > 0,\ \text{ then } \ a \leftarrow t$$

$$\text{If } f'(t)f(b) > 0,\ \text{ then } \ b \leftarrow t$$

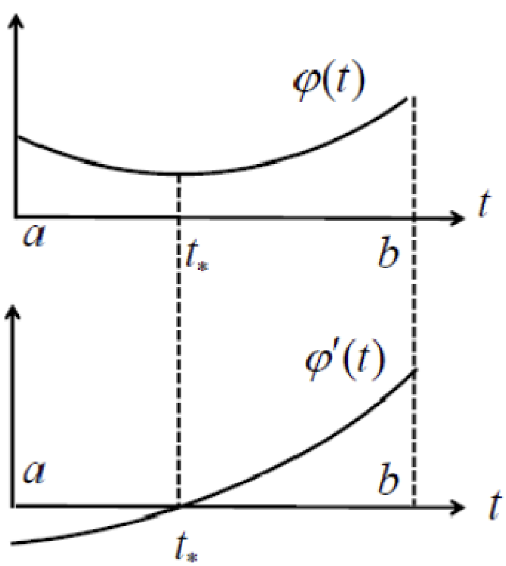


- if we know the second-order gradient, the convergence is very fast

Line Search: by Gradient

$$t = \operatorname{argmin}_t f(x + t\Delta x) \triangleq q(t)$$

- if we know the gradient, we can improve the compressive ratio to $c = 0.5$

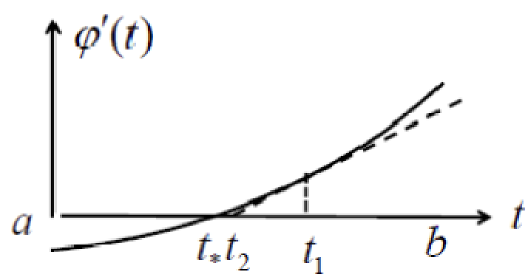


$f'(t_*) = 0$

does that mean we should always use second-order gradient?

If $f'(t)f(a) > 0$, then $a \leftarrow t$

If $f'(t)f(b) > 0$, then $b \leftarrow t$

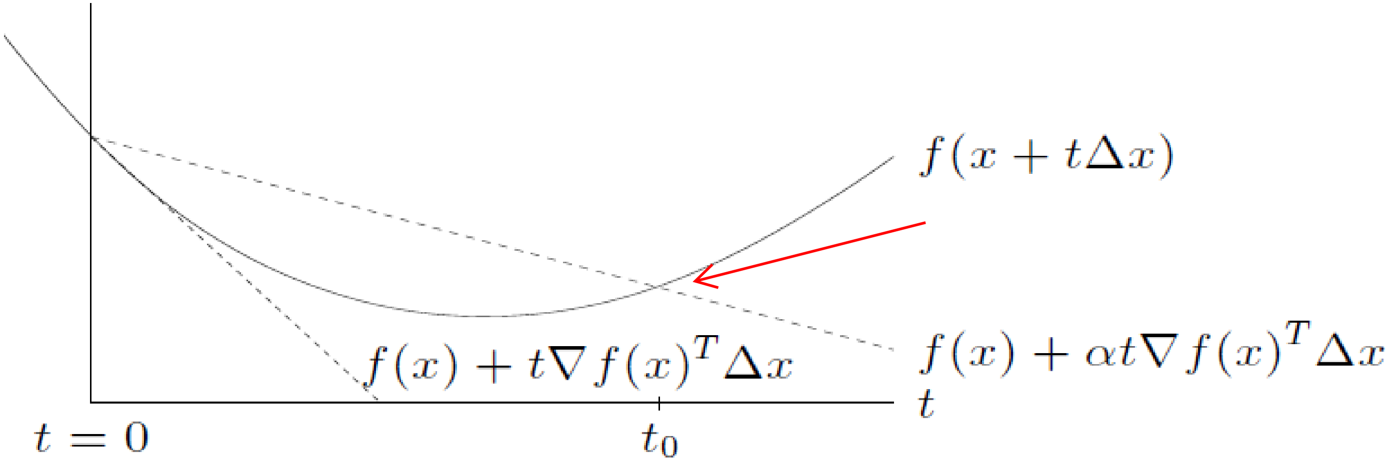


- if we know the second-order gradient, the convergence is very fast

Line Search: Backtracking

- exact search: $t = \text{argmin}_t\, f(x + t\Delta x) \triangleq q(t)$
- backtracking line search, with $\alpha \in (0,0.5), \beta \in (0,1)$
  - starting from $t = 1$, repeat $t = \beta t,$ until

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



Descent Direction

$$\min \quad f(x)$$

- to produce a sequence of points $x^{(k)}$ to approach the optimum

$$x^{(k+1)} = x^{(k)} + t^{(k)} \Delta x^{(k)} \quad \text{with} \quad f\left(x^{(k+1)}\right) < f(x^{(k)})$$

- if $\nabla f\left(x^{(k)}\right)^{\top} \Delta x^{(k)} \geq 0$, from convexity

$$f\left(x^{(k+1)}\right) \geq f\left(x^{(k)}\right) + \nabla f \Delta x^{(k)} \geq 0$$

$$\Downarrow$$

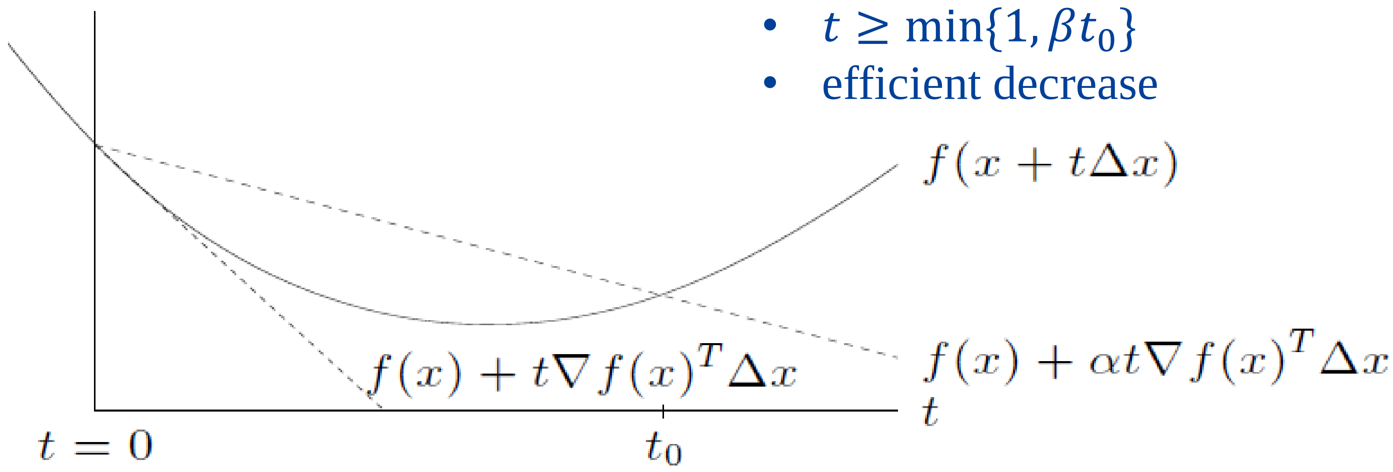
$$f\left(x^{(k+1)}\right) < f\left(x^{(k)}\right) \rightarrow \nabla f\left(x^{(k)}\right)^{\top} \Delta x^{(k)} < 0$$

Line Search: Backtracking

- exact search: $t = \mathrm{argmin}_t\, f(x + t\Delta x) \triangleq q(t)$
- backtracking line search, with $\alpha \in (0,0.5), \beta \in (0,1)$
  - starting from $t = 1$, repeat $t = \beta t,$ until

$$f(x + t\Delta x) < f(x) + \alpha t \nabla f(x)^\top \Delta x$$

- descent $\nabla f(x)^\top \Delta x < 0$
- $t \geq \min\{1, \beta t_0\}$
- efficient decrease



when t is small enough,

$$f(x + t\Delta x) \approx f(x) + t\nabla f(x)^\top \Delta x < f(x) + \alpha t \nabla f(x)^\top \Delta x$$

Descent Direction

$$\min \quad f(x)$$

- to produce a sequence of points $x^{(k)}$ to approach the optimum

$$x^{(k+1)} = x^{(k)} + t^{(k)}\Delta x^{(k)} \quad \text{with} \quad f\left(x^{(k+1)}\right) < f(x^{(k)})$$

- if $\nabla f\left(x^{(k)}\right)^{\top}\Delta x^{(k)} \geq 0$, from convexity

$$f\left(x^{(k+1)}\right) \geq f\left(x^{(k)}\right) + \nabla f \Delta x^{(k)} \geq 0$$

$$\Downarrow$$

$$f\left(x^{(k+1)}\right) < f\left(x^{(k)}\right) \rightarrow \nabla f\left(x^{(k)}\right)^{\top}\Delta x^{(k)} < 0$$

- a natural choice is gradient: **G**radient **D**escent algorithm

$$\Delta x^{(k)} = -\nabla f\left(x^{(k)}\right)$$

Gradient Descent Algorithm

given a starting point $x \in \mathbf{dom}\, f$.
repeat
1. $\Delta x := -\nabla f(x)$.
2. *Line search.* Choose step size t via exact or backtracking line search.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

- stopping criterion: $\|\nabla f(x)\|_2 \leq \varepsilon$
- convergence analysis
  - f is strongly convex on a set S: there exists an $m > 0$, such that

$$\nabla^2 f(x) \geq m\mathrm{I}, \forall x \in S$$

  - for any $x, y \in S$, there is

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

Gradient Descent Algorithm

given a starting point $x \in \mathbf{dom}\, f$.
repeat
1. $\Delta x := -\nabla f(x)$.
2. *Line search.* Choose step size t via exact or backtracking line search.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

- stopping criterion: $\|\nabla f(x)\|_2 \le \varepsilon$
- convergence analysis
  - f is strongly convex on a set S: there exists an $m > 0$, such that

$$\nabla^2 f(x) \succeq m\mathrm{I}, \forall x \in S$$

  - for any $x, y \in S$, there is

$$f(y) \ge f(x) + \nabla f(x)^{\mathsf{T}}(y - x) + \frac{m}{2}\|x - y\|_2^2$$

- how good the solution?

  bound the distance to the optimum
- how fast the convergence?

  the convergence rate

GD: Analysis

- solution quality

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

TRH is a quadratic function on y, take gradient, the minimum is at

$$\tilde{y} = x - \frac{1}{m}\nabla f(x)$$

- notice that the above holds for all y, then

$$f(x) - p^* \leq \frac{1}{2m}\|\nabla f(x)\|_2^2$$

- the stopping criterion $\|\nabla f(x)\|_2 \leq \varepsilon$ can control the distance to the optimum

GD: Analysis

- from strong convexity and boundedness

$$m\mathrm{I} \preceq \nabla^2 f(x) \preceq M\mathrm{I}, \qquad \forall x \in S$$

- using the right equality, we have

$$f\big(x - t\nabla f(x)\big) \le f(x) - t\|\nabla f(x)\|_2^2 + \frac{Mt^2}{2}\|\nabla f(x)\|_2^2$$

- with exact line search, we attain the best $t^* = \frac{1}{M}$, then

$$f\big(x - t\nabla f(x)\big) \le f(x) - \frac{1}{2M}\|\nabla f(x)\|_2^2$$

  - efficient decrease

GD: Analysis

- efficient decrease

$$f\left(x^{(k+1)}\right) \leq f\left(x^{(k)}\right) - \frac{1}{2M}\left\|\nabla f\left(x^{(k)}\right)\right\|_2^2$$

- solution quality

$$f\left(x^{(k)}\right) - p^* \leq \frac{1}{2m}\left\|\nabla f(x^{(k)})\ \right\|_2^2$$

- therefore,

$$f\left(x^{(k+1)}\right) - p^* \leq (1 - m/M)(f\left(x^{(k)}\right) - p^*)$$

- iteratively update show that

$$f\left(x^{(k)}\right) - p^* \leq (1 - c)^k(f\left(x^{(0)}\right) - p^*)$$

condition number

$c = m/M$

initial guess

- linear convergence (log err vs. iter curve is below a line)
- the speed depends on the Hessian

GD: Analysis

- from strong convexity and boundedness

$$mI \preceq \nabla^2 f(x) \preceq MI, \qquad \forall x \in S$$

- using the right equality, we have

$$f\big(x - t\nabla f(x)\big) \le f(x) - t\|\nabla f(x)\|_2^2 + \frac{Mt^2}{2}\|\nabla f(x)\|_2^2$$

- with backtracking line search for GD

$$f\big(x - t\nabla f(x)\big) < f(x) - \alpha t\|\nabla f(x)\|_2^2$$

when $0 \le t \le 1/M$, we have $-t + \frac{Mt^2}{2} \le -\frac{t}{2}$

$$f\big(x - t\nabla f(x)\big) \le f(x) - t\|\nabla f(x)\|_2^2 + \frac{Mt^2}{2}\|\nabla f(x)\|_2^2 \le f(x) - \frac{1}{2}t\|\nabla f(x)\|_2^2$$

since $\alpha \in (0, 0.5)$, the backtracking condition can be satisfied by $0 \le t \le 1/M$

GD: Analysis

- with backtracking line search for GD: $t = \beta^l$

$$f\big(x - t\nabla f(x)\big) < f(x) - \alpha t \|\nabla f(x)\|_2^2$$

  - if $l = 0$, then $f\big(x - t\nabla f(x)\big) < f(x) - \alpha \|\nabla f(x)\|_2^2$
  - if $l > 1$, then

$$\left.\begin{aligned} f\big(x - \beta^{-1} t\nabla f(x)\big) &> f(x) - \alpha\beta^{-1} t \|\nabla f(x)\|_2^2 \\ f\big(x - \beta^{-1} t\nabla f(x)\big) &\le f(x) - \beta^{-1} t \|\nabla f(x)\|_2^2 + \frac{M(\beta^{-1}t)^2}{2}\|\nabla f(x)\|_2^2 \end{aligned}\right\}$$

$$-\alpha < -1 + \frac{M}{2}\beta^{-1} t$$

$$\Downarrow$$

$$t > \frac{2(1-\alpha)\beta}{M} > \frac{\beta}{M}$$

- we now have

$$f\big(x - t\nabla f(x)\big) < f(x) - \min\{\alpha, \alpha\beta/M\} \|\nabla f(x)\|_2^2$$

GD: Analysis

- with backtracking line search for GD: $t = \beta^l$

$$f\big(x - t\nabla f(x)\big) < f(x) - \alpha t\|\nabla f(x)\|_2^2$$

  - if $l = 0$, then $f\big(x - t\nabla f(x)\big) < f(x) - \alpha\|\nabla f(x)\|_2^2$
  - if $l > 1$, then

$$\left.\begin{aligned} f\big(x - \beta^{-1}t\nabla f(x)\big) &> f(x) - \alpha\beta^{-1}t\|\nabla f(x)\|_2^2 \\ f\big(x - \beta^{-1}t\nabla f(x)\big) &\le f(x) - \beta^{-1}t\|\nabla f(x)\|_2^2 + \frac{M(\beta^{-1}t)^2}{2}\|\nabla f(x)\|_2^2 \end{aligned}\right\}$$

$$-\alpha < -1 + \frac{M}{2}\beta^{-1}t$$

$$\Downarrow$$

$$t > \frac{2(1-\alpha)\beta}{M} > \frac{\beta}{M}$$

- we now have

$$f\big(x - t\nabla f(x)\big) < f(x) - \min\{\alpha, \alpha\beta/M\}\,\|\nabla f(x)\|_2^2$$

$$f\big(x - t\nabla f(x)\big) \le f(x) - \frac{1}{2M}\|\nabla f(x)\|_2^2$$

- efficient decrease
- following analysis is similar

Gradient Descent Algorithm

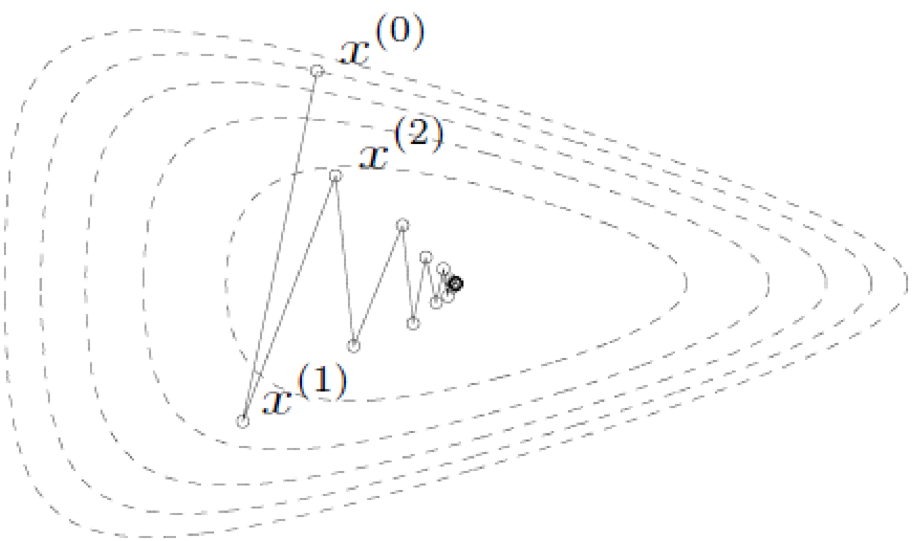
given a starting point $x \in \mathbf{dom}\, f$.
repeat
1. $\Delta x := -\nabla f(x)$.
2. *Line search.* Choose step size t via exact or backtracking line search.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

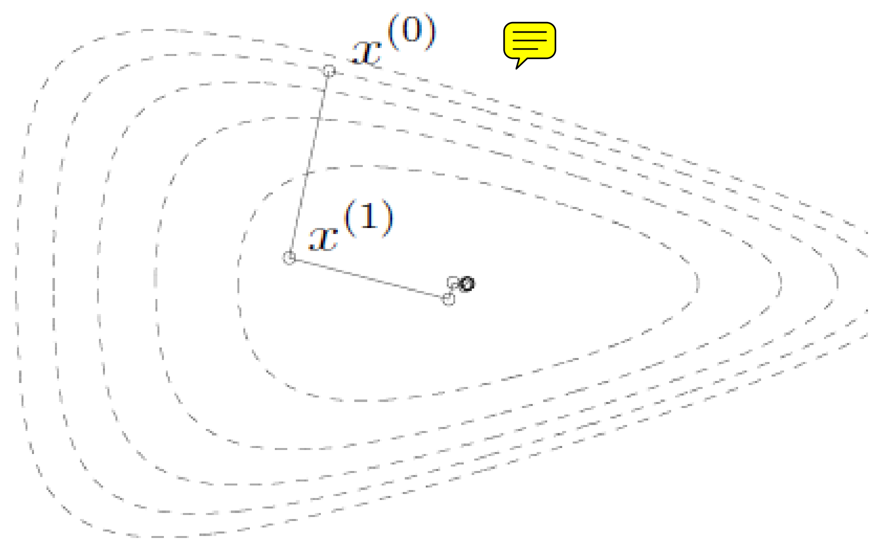
- line search (exact, backtracking)
- solution quality (strong convexity)
- linear convergence
  (strong convexity, boundedness)

Gradient Descent Algorithm

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



backtracking line search



exact line search

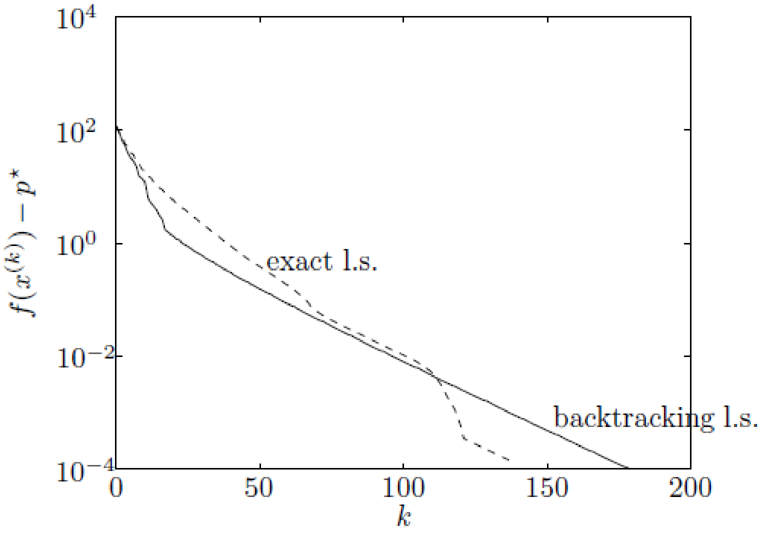
Gradient Descent Algorithm

given a starting point $x \in \mathbf{dom}\, f$.
repeat
1. $\Delta x := -\nabla f(x)$.
2. *Line search.* Choose step size t via exact or backtracking line search.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

$$f(x) = c^T x - \sum_{i=1}^{500} \log(b_i - a_i^T x)$$

- line search (exact, backtracking)
- solution quality (strong convexity)
- linear convergence
  (strong convexity, boundedness)



Gradient Descent Algorithm

given a starting point $x \in \mathbf{dom}\, f$.
repeat
1. $\Delta x := -\nabla f(x)$.
2. *Line search.* Choose step size t via exact or backtracking line search.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

- line search (exact, backtracking)
- solution quality (strong convexity)
- linear convergence
  (strong convexity, boundedness)
- convergence speed relies on condition number $c = \frac{m}{M}$

Steepest Descent Method

$$\min \quad f(x)$$

- to produce a sequence of points $x^{(k)}$ to approach the optimum

$$x^{(k+1)} = x^{(k)} + t^{(k)} \Delta x^{(k)} \quad \text{with} \quad f\left(x^{(k+1)}\right) < f(x^{(k)})$$

$$f\left(x^{(k+1)}\right) \leq f\left(x^{(k)}\right) + \nabla f\left(x^{(k)}\right)^{\top} \Delta x^{(k)}$$

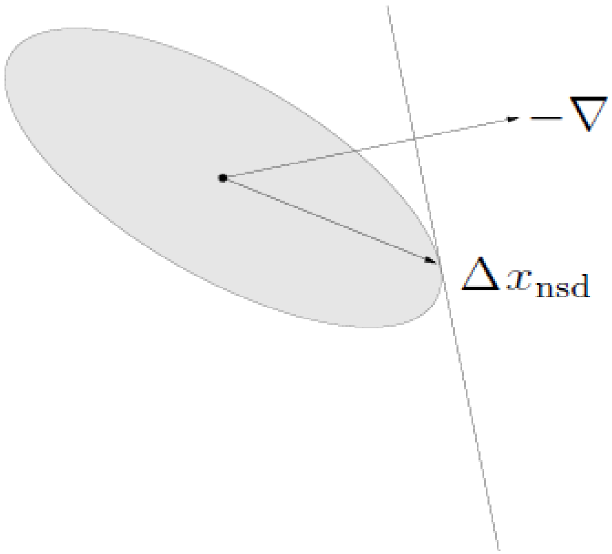
- which direction is the "best"
  - normalized steepest descent direction

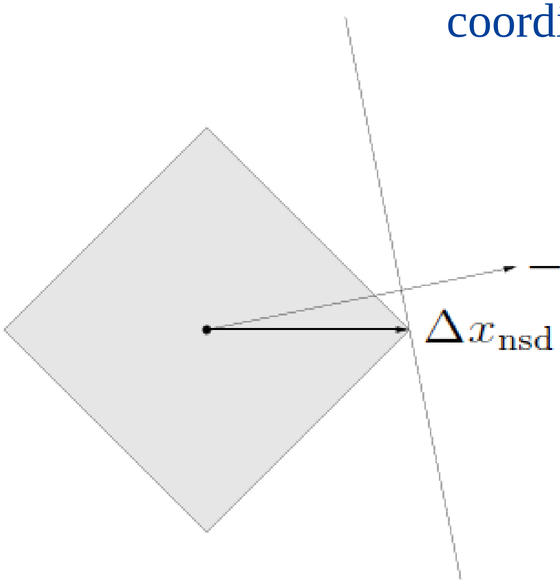
$$\Delta x_{\text{nsd}} = \text{argmin}_v \{\nabla f(x)^{\top} v, \text{s.t.} \|v\| = 1\}$$

- projection of $-\nabla f(x)$ on the unit ball
- different norm corresponds to different directions

Steepest Descent Method

coordinate descent





$$\Delta x_{\mathrm{nsd}} = \mathrm{argmin}_v \{\nabla f(x)^\top v, \mathrm{s.t.} \|v\| = 1\}$$

- projection of $-\nabla f(x)$ on the unit ball
- different norm corresponds to different directions

Sub-gradient for LASSO

$$\min_{x} \quad \gamma \sum_{j=1}^{n} |x_j| + \frac{1}{2} \sum_{i=1}^{m} (x^\top a_i - b_i)^2$$

- sub-gradient

$$\frac{\partial f}{\partial x} \in \lambda \frac{\partial \|x\|_1}{\partial x} + A^\top (Ax - Y)$$

define shrinkage operator:

$$\left(S_\lambda(u)\right)_i = \begin{cases} u_i - \lambda, & u(i) \geq \lambda \\ 0, & |u(i)| < \lambda \\ u_i + \lambda, & u(i) \leq -\lambda \end{cases}$$

- optimality condition

$$0 \in \lambda \frac{\partial \|x\|_1}{\partial x} + A^\top (Ax - Y)$$

$$\Downarrow$$

$$x = S_\lambda(x - A^\top (Ax - Y)\ )$$

Iterative Soft Thresholding Algorithm

$$\min_{x} \quad \gamma \sum_{j=1}^{n} |x_j| + \frac{1}{2} \sum_{i=1}^{m} (x^\top a_i - b_i)^2$$

- optimality condition

$$x = S_\lambda(x - A^\top (Ax - Y))$$

- iterative update

$$x^{k+1} = S_\lambda\left(x^k - A^\top (Ax^k - Y) \right)$$

Convergence discussion for

$$x^{k+1} = T(x^k)$$

- a fixed point satisfies optimality condition
- the operator is non-expansive

Iterative Soft Thresholding Algorithm

$$x^{k+1} = S_\lambda\left(x^k - A^\top(Ax^k - Y)\ \right)$$

- a fixed point satisfies the optimality condition

$$x = S_\lambda(x - A^\top(Ax - Y)\ )$$

- non-expansive

$$T(x) \triangleq S_\lambda(x - A^\top(Ax - Y)\ )$$

how to verify?

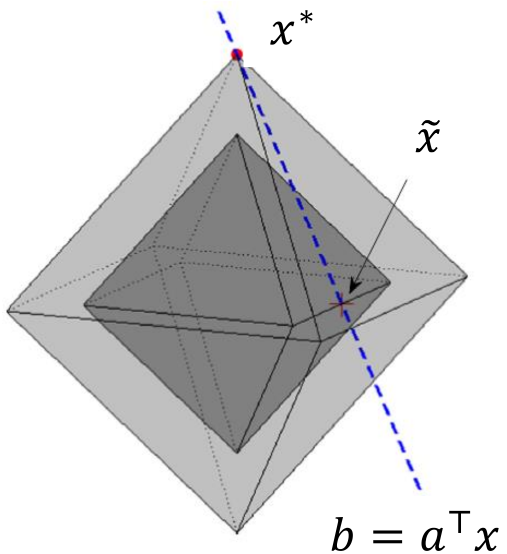
$$\|S_\lambda(u) - S_\lambda(v)\| \leq \|u - v\|$$

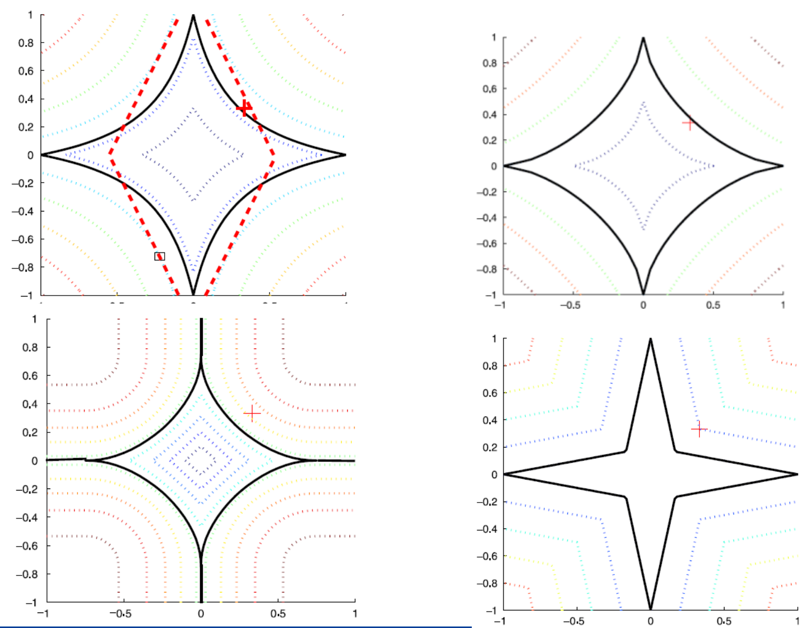
$$\begin{aligned}
\|T(u) - T(v)\| &= \left\|S_\lambda(u - A^\top(Au - Y)\ ) - S_\lambda(v - A^\top(Av - Y)\ )\right\| \\
&\leq \left\|u - A^\top(Au - Y) - v + A^\top(Av - Y)\right\| \\
&= \left\|(I - A^\top A)u - (I - A^\top A)v\right\| \\
&\leq \left\|I - A^\top A\right\| \|u - v\|
\end{aligned}$$

convergence condition:
$\left\|I - A^\top A\right\| < 1$

Off the convexity

- l1-norm is the best approximation among convex functions
- non-convex functions could enhance the sparsity





Off the convexity

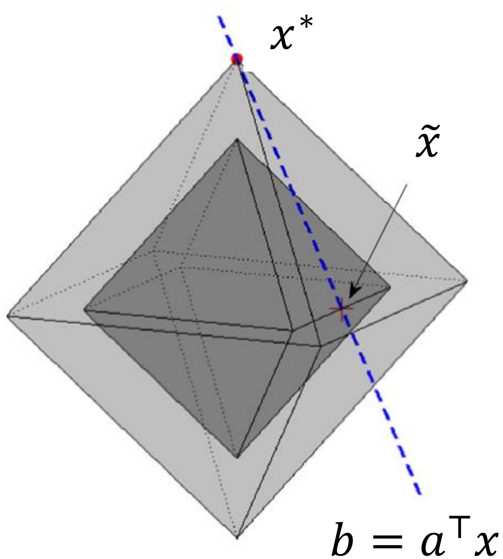
$$\min_x \quad \gamma \sum_{j=1}^{n} |x_j| + \frac{1}{2} \sum_{i=1}^{m} (x^\top a_i - b_i)^2$$

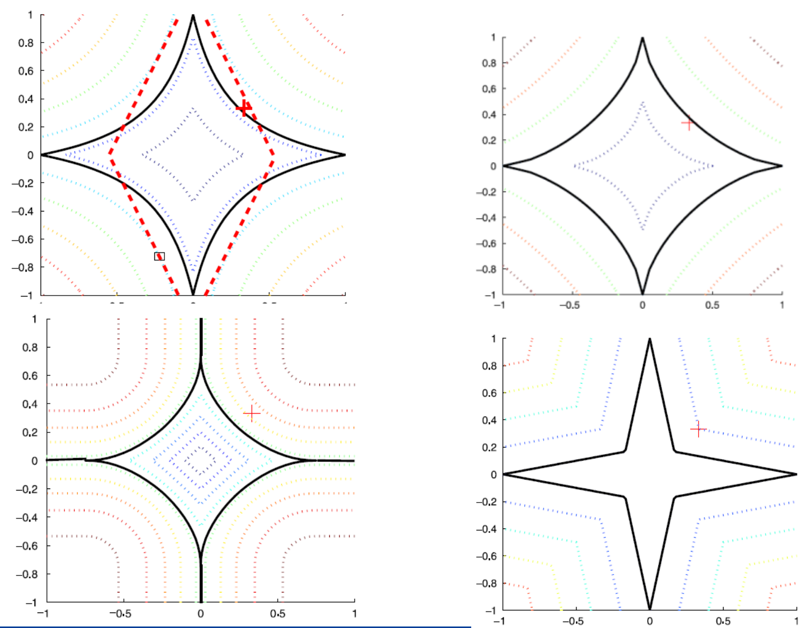
- in the view of iterative reweighted

$$x^{k+1} = \operatorname*{argmin}_x \quad \gamma \sum_{j=1}^{n} p_j^k |x_j| + \frac{1}{2} \sum_{i=1}^{m} (x^\top a_i - b_i)^2$$

How to design the weights?

p_j^k is inverse proportional to $|x_j^k|$





Off the convexity

$$\min_x \quad \gamma \sum_{j=1}^{n} |x_j| + \frac{1}{2} \sum_{i=1}^{m} (x^\top a_i - b_i)^2$$

- in the view of iterative reweighted

$$x^{k+1} = \operatorname*{argmin}_x \quad \gamma \sum_{j=1}^{n} p_j^k |x_j| + \frac{1}{2} \sum_{i=1}^{m} (x^\top a_i - b_i)^2$$

$$\left\| S_{\lambda,p}(u) - S_{\lambda,p}(v) \right\| \ ? \ \|u - v\|$$

- ISTA for non-convex penalty:

$$x^{k+1} = S_{\lambda,p^k}\left(x^k - A^\top (Ax^k - Y)\right)$$

$$\left(S_{\lambda,p}(u)\right)_i = \begin{cases} u_i - p_i\lambda, & u_i \geq p_i\lambda \\ 0, & |u(i)| < p_i\lambda \\ u_i + \lambda, & u(i) \leq -p_i\lambda \end{cases}$$

$u > v > 0$

$$u - p_i(u)\lambda$$
$$\wedge$$
$$v - p_i(v)\lambda$$

$$u - v >_{\text{possible}} u - p_i(u)\lambda - (v - p_i(v)\lambda)$$

目录 Contents

1 Optimality Condition

2 Gradient Descent Algorithm

3 **Newton Method**

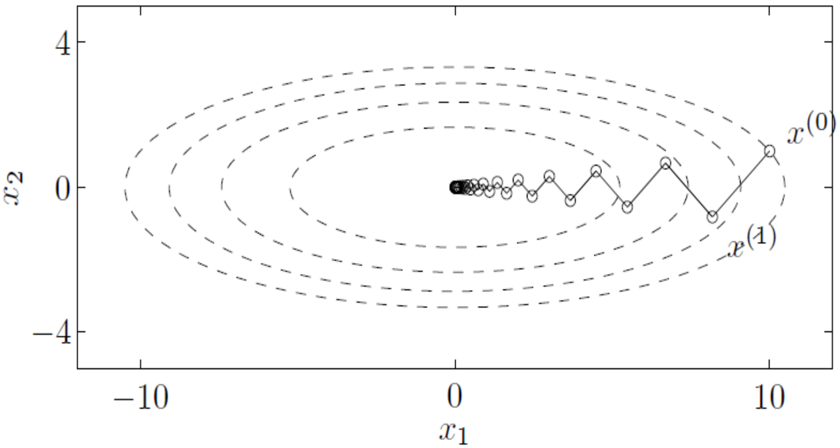
4 Nesterov Acceleration

GD: Staircase

given a starting point $x \in \mathbf{dom}\, f$.
repeat
1. $\Delta x := -\nabla f(x)$.
2. *Line search.* Choose step size t via exact or backtracking line search.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

- gradient descent with exact line search results in "staircase"
  - the reason
  - when there is no
  - when it is worse

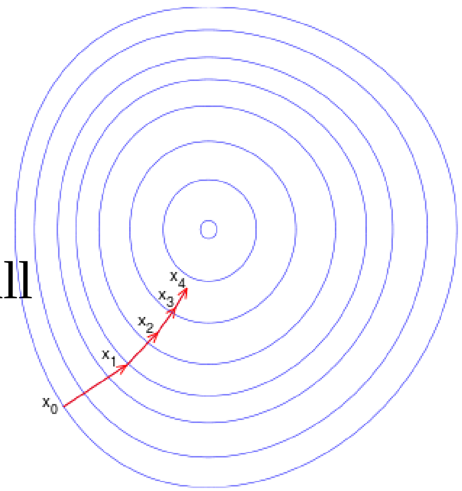


Adjustment

$$f\left(x^{(k)}\right) - p^* \le (1 - c)^k \left(f\left(x^{(0)}\right) - p^*\right)$$

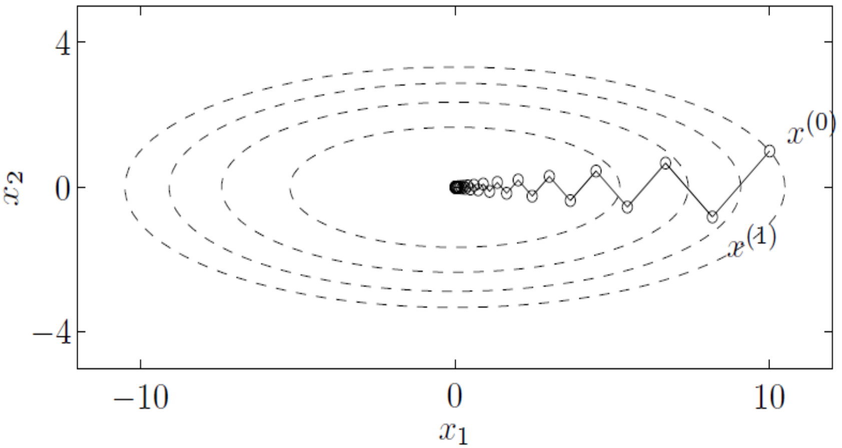
condition number

$$c = m/M$$



- when there is no staircase: $c = 1$, i.e., the contour is a ball
- if not, we should adjust the descent direction, related to the condition number

$$\Delta x_{\mathrm{nt}} = -\nabla^2 f(x)^{-1} \nabla f(x)$$



Newton Step

$$\Delta x_{\text{nt}} = -\nabla^2 f(x)^{-1} \nabla f(x)$$

how to prove?

- Newton step is a descent direction
- iteratively minimize the second approximation

$$f(x+v) \approx f(x) + \nabla f(x)^\top v + \frac{1}{2} v^\top \nabla^2 f(x) v$$

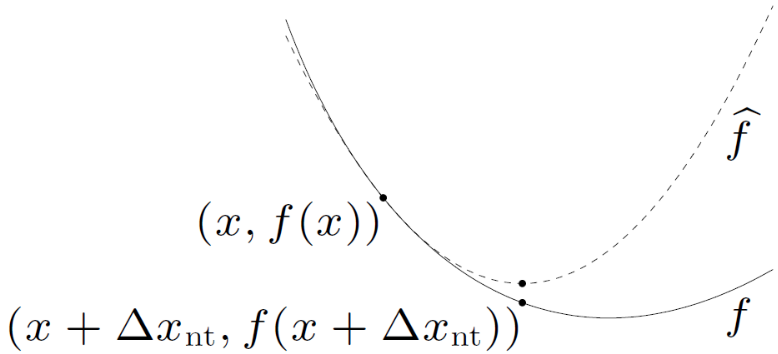
$$\Downarrow$$

$$\nabla f(x) + \nabla^2 f(x) v = 0$$

$$\Downarrow$$

$$v^* = -\nabla^2 f(x)^{-1} \nabla f(x)$$

link to IRLS

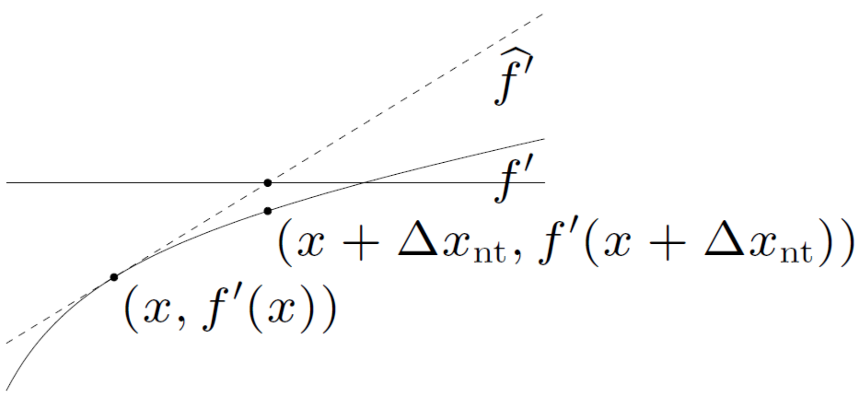


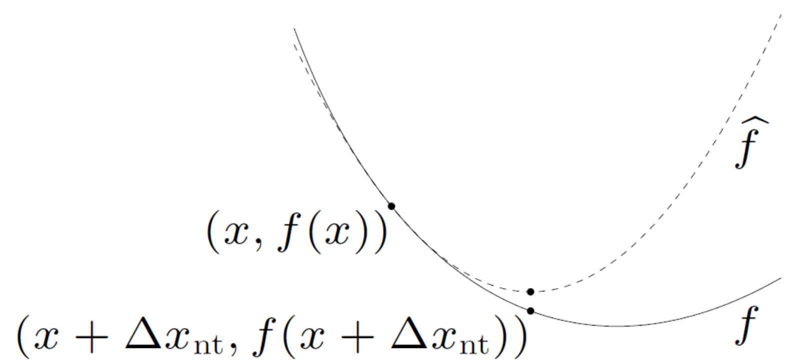
Newton Step

$$\Delta x_{\mathrm{nt}} = -\nabla^2 f(x)^{-1} \nabla f(x)$$

- Newton step is a descent direction
- iteratively solve the linearized optimality condition

$$\nabla f(x+v) \approx \nabla f(x)^{\top} + \nabla^2 f(x) v = 0$$





Newton Step

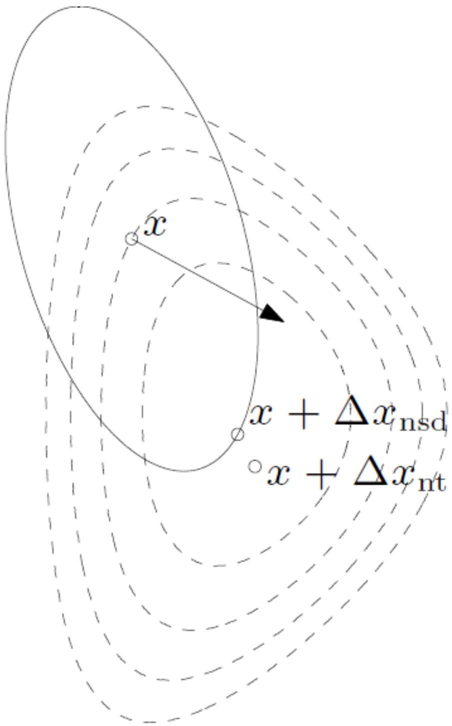
$$\Delta x_{\text{nt}} = -\nabla^2 f(x)^{-1} \nabla f(x)$$

- Newton step is a descent direction
- steepest descent direction in local Hessian norm

$$\Delta x_{\text{nt}} = \operatorname{argmin}_v \left\{ \nabla f(x)^\top v, \text{s.t.} \|v\|_{\nabla^2 f(x)} = 1 \right\}$$

$$\|v\|_{\nabla^2 f(x)} = \sqrt{v^\top \nabla^2 f(x) v}$$

- modified Euclid distance



Newton Decrement

- consider the quadratic approximation

$$f(x+v) \approx f(x) + \nabla f(x)^\top v + \frac{1}{2} v^\top \nabla^2 f(x) v \triangleq f(y)$$

- the gap to the approximated optimum

$$\boxed{f(x) - \inf_y f(y) = \frac{1}{2} \nabla f(x)^\top \nabla^2 f(x)^{-1} \nabla f(x) \triangleq \frac{1}{2} \lambda(x)^2}$$

Newton decrement

- Newton decrement is an estimation of $f(x) - f^*$
- affine invariant: for $g(y) \triangleq f(Tx)$, they have the same Newton decrement

$$\lambda_g(y) = \lambda_f(x)$$

  - independent of linear changes of coordinates

(Damped) Newton's method

- Gradient descent in Newton step with backtracking line search

given a starting point $x \in \mathbf{dom}\, f$, tolerance $\epsilon > 0$.
repeat
1. *Compute the Newton step and decrement.*
$$\Delta x_{\mathrm{nt}} := -\nabla^2 f(x)^{-1} \nabla f(x); \quad \lambda^2 := \nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x).$$
2. *Stopping criterion.* **quit** if $\lambda^2/2 \le \epsilon$.
3. *Line search.* Choose step size t by backtracking line search.
4. *Update.* $x := x + t\Delta x_{\mathrm{nt}}$.

- affine invariant: for $g(y) \triangleq f(Tx)$, they have the same Newton decrement
  - independent of linear changes of coordinates

$$y^0 = Tx^0 \Longrightarrow y^k = Tx^k$$

  - how about the gradient descent?

Convergence analysis: Assumption

- update

$$x^{k+1} = x^k + t^k d^k \qquad d^k = \nabla^2 f\left(x^k\right)^{-1} \nabla f(x^k)$$

- strongly convexity and boundedness assumption

$$m\mathrm{I} \leq \nabla^2 f(x) \leq M\mathrm{I}, \qquad \forall x \in S$$

- Lipschitz continuous on the second order gradient

$$\left\|\nabla^2 f(y) - \nabla^2 f(x)\right\| \leq L\|y - x\|, \forall x, y \in S$$

  - to bound the gap between $f(x)$ and its second-order approximation along d

$$f(x+d) = f(x) + \nabla f(x)^\top d + \frac{1}{2} d^\top \nabla^2 f(x + \xi d) d, \quad \xi \in [0,1]$$

$$\left| f(x+d) - \left(f(x) + \nabla f(x)^\top d + \frac{1}{2} d^\top \nabla^2 f(x) d\right) \right| = \left| d^\top \left(\nabla^2 f(x+\xi d) - \nabla^2 f(x)\right) d \right|$$

$$\leq \frac{1}{2}\left\|\nabla^2 f(x+\xi d) - \nabla^2 f(x)\right\| \|d\|^2 \leq \frac{L}{2}|\xi| \|d\|^3 \leq \frac{L}{2}\|d\|^3$$

Convergence analysis: Assumption

- update

$$x^{k+1} = x^k + t^k d^k \qquad d^k = \nabla^2 f\left(x^k\right)^{-1} \nabla f(x^k)$$

- strongly convexity and boundedness assumption

$$m\mathrm{I} \leq \nabla^2 f(x) \leq M\mathrm{I}, \qquad \forall x \in S$$

- Lipschitz continuous on the second order gradient

$$\left\|\nabla^2 f(y) - \nabla^2 f(x)\right\| \leq L\|y - x\|, \forall x, y \in S$$

  - to bound the norm of the gradient

$$\left.\begin{aligned} &\frac{d\nabla f(x+td)}{dt} = \nabla^2 f(x+td)d \\ &\nabla f(x) = \nabla^2 f(x) d \end{aligned}\right\} \int_0^1 \left(\nabla^2 f(x+td)d - \nabla^2 f(x)d\right) dt = \nabla f(x+td)$$

$$\|\nabla f(x+td)\| \leq \int_0^1 Lt\|d\|^2\, dt = \frac{L}{2}\|d\|^2 = \frac{L}{2}\left\|\nabla^2 f(x)^{-1} \nabla f(x)\right\|^2 \leq \frac{L}{2m^2}\|\nabla f(x)\|^2$$

Convergence analysis: when backtracking stops

$$f(x+td) \le f(x) + \alpha \nabla f(x)^\top d$$

- from Lipschitz condition

$$f(x) + \nabla f(x)^\top d - \frac{1}{2}\nabla f(x)^\top d + \frac{L}{2}\|d\|^3 \le f(x+td) \le f(x) + \alpha \nabla f(x)^\top d$$

$$\frac{1}{2}\nabla f(x)^\top d + \frac{L}{2}\|d\|^3 \le \alpha \nabla f(x)^\top d$$

$$\frac{\|d\|^3}{|\nabla f(x)^\top d|} \le \frac{1-2\alpha}{L} \Rightarrow \|\nabla f(x)\| \le \frac{(1-2\alpha)m^2}{L}$$

$$\|\nabla f(x+td)\| \le \frac{L}{2m^2}\|\nabla f(x)\|^2 \qquad \|\nabla f(x+td)\| \le \frac{(1-\alpha)^2 m^2}{2L} \le \frac{(1-\alpha)m^2}{L}$$

- if for $\hat{k}$,

$$\left\|\nabla f(x^{\hat{k}})\right\| \le \frac{(1-2\alpha)m^2}{L} \Rightarrow \|\nabla f(x^k)\| \le \frac{(1-2\alpha)m^2}{L}, \forall k > \hat{k}$$

Convergence analysis: speed

- when $\|\nabla f(x^k)\| \leq \frac{(1-2\alpha)m^2}{L}$ $\xrightarrow{0<\alpha<1}$ $\frac{L}{m^2}\|\nabla f(x^k)\| \leq 1$

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

- for any $k > \hat{k}$

$$\frac{L}{2m^2}\|\nabla f(x^k)\| \leq \left(\frac{L}{2m^2}\|\nabla f(x^k)\|^2\right)^{2^{k-\hat{k}}} \leq \left(\frac{1}{2}\right)^{2^{k-\hat{k}}}$$

- following the analysis in GD

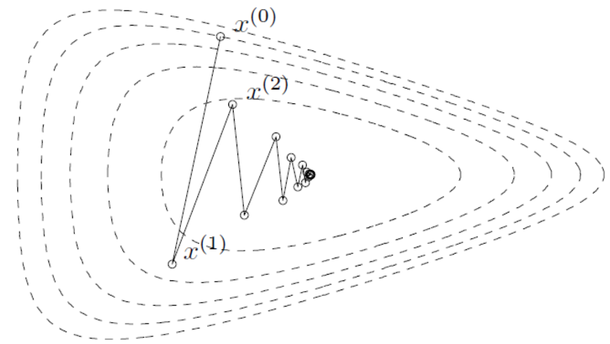
$$f(x^k) - f^* \leq \frac{4m^4}{C_1L^2}\left(\frac{1}{2}\right)^{2^{k-\hat{k}+1}}$$

$$6 + \frac{M^2L^2/m^5}{\alpha\beta\min\{1, 9(1-2\alpha)^2\}}(f(x^{(0)}) - p^\star).$$

$2^{2^6} \approx 10^{19}$

Generally, we only needs no more than 6 iteration in this phase

Convergence analysis: speed



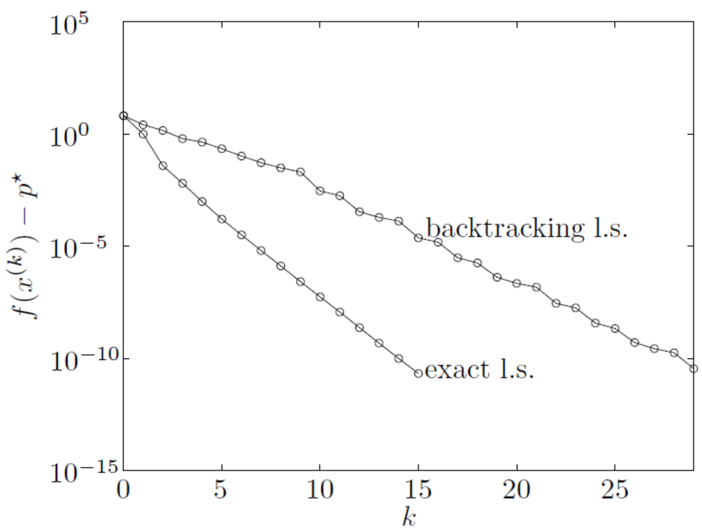
- when $\left\|\nabla f(x^{\hat{k}})\right\| > \frac{(1-2\alpha)m^2}{L} \triangleq \eta$

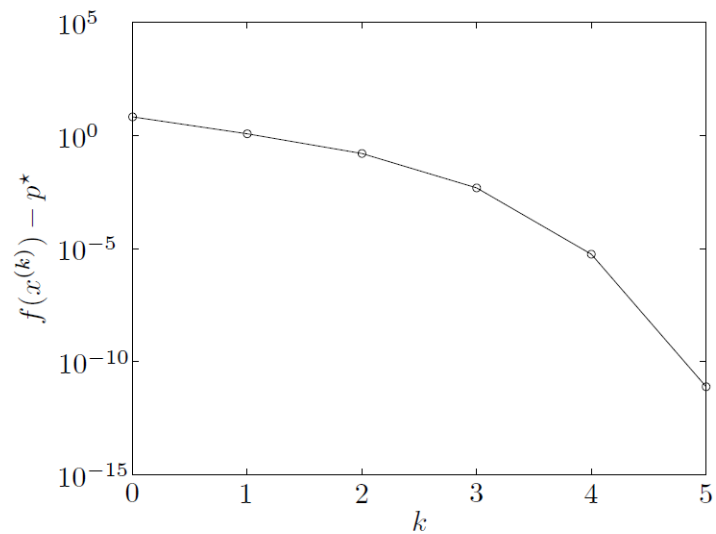
  following the discussion in GD, we will have

$$f\left(x^{k}\right) - f\left(x^{k+1}\right) \geq C\eta^2$$

before: damped Newton phase

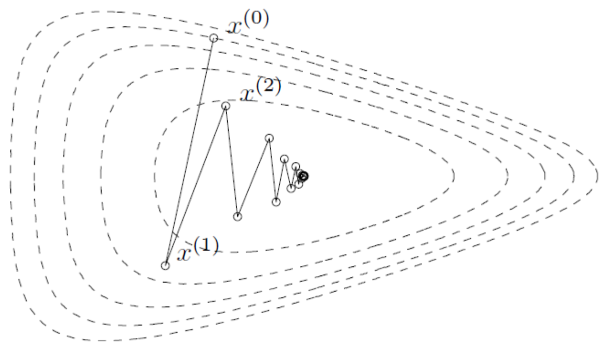
- quadratically convergent phase





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

Convergence analysis: speed



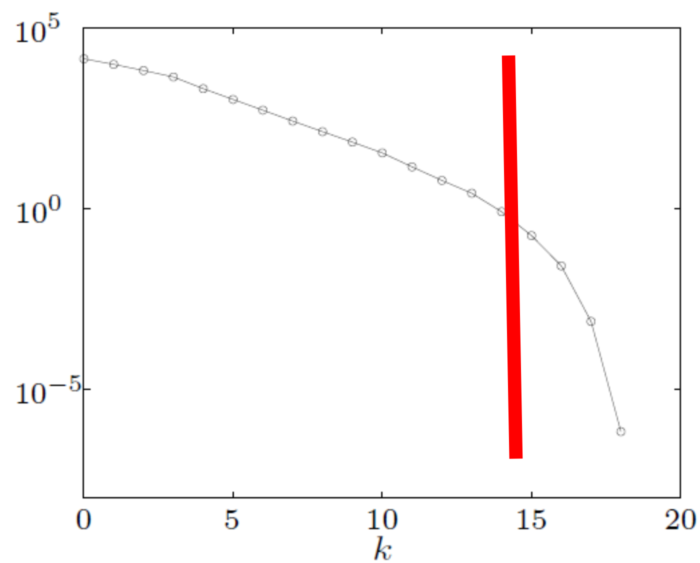
- when $\left\|\nabla f(x^{\hat{k}})\right\| > \dfrac{(1-2\alpha)m^2}{L} \triangleq \eta$

  following the discussion in GD, we will have

$$f\left(x^k\right) - f\left(x^{k+1}\right) \geq C\eta^2$$

before: damped Newton phase

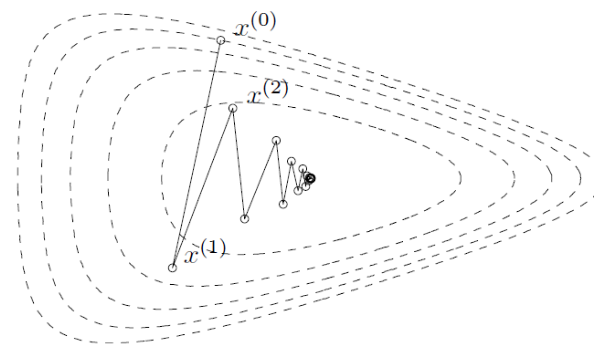
- quadratically convergent phase



- Convergence of Newton's method is rapid in general, and quadratic near $x^\star$. Once the quadratic convergence phase is reached, at most six or so iterations are required to produce a solution of very high accuracy.
- Newton's method is affine invariant. It is insensitive to the choice of coordinates, or the condition number of the sublevel sets of the objective.
- Newton's method scales well with problem size. Its performance on problems in $\mathbf{R}^{10000}$ is similar to its performance on problems in $\mathbf{R}^{10}$, with only a modest increase in the number of steps required.
- The good performance of Newton's method is not dependent on the choice of algorithm parameters. In contrast, the choice of norm for steepest descent plays a critical role in its performance.

$$f(x) = -\sum_{i=1}^{10000} \log(1 - x_i^2) - \sum_{i=1}^{100000} \log(b_i - a_i^T x)$$

Convergence analysis: speed



- when $\left\|\nabla f(x^{\hat{k}})\right\| > \dfrac{(1-2\alpha)m^2}{L} \triangleq \eta$

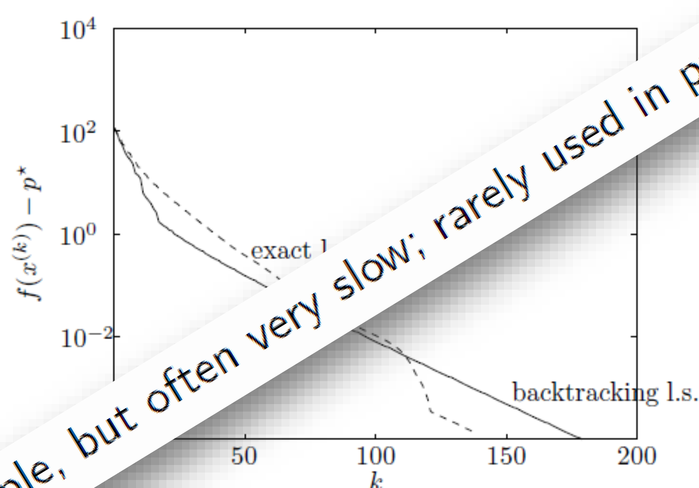
  following the discussion in GD, we will have

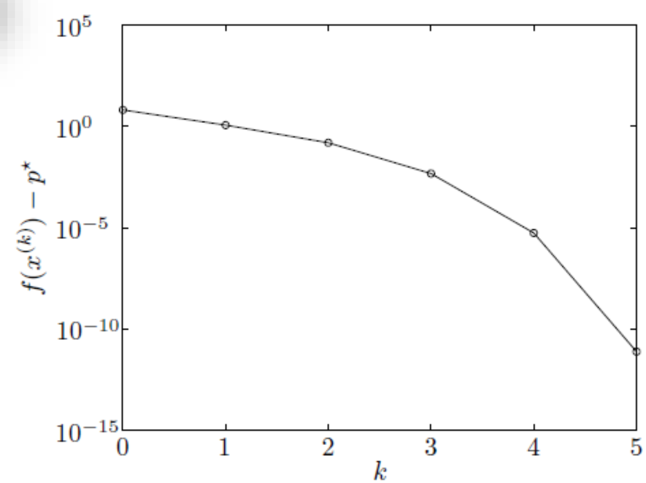
$$f\left(x^k\right) - f\left(x^{k+1}\right) \geq C\eta^2$$

before: damped Newton phase

- quadratically convergent phase

very simple, but often very slow; rarely used in practice





$$f(x) = c^T x - \sum_{i=1}^{500} \log(b_i - a_i^T x)$$

目录 Contents

1 Optimality Condition

2 Gradient Descent Algorithm

3 Newton Method

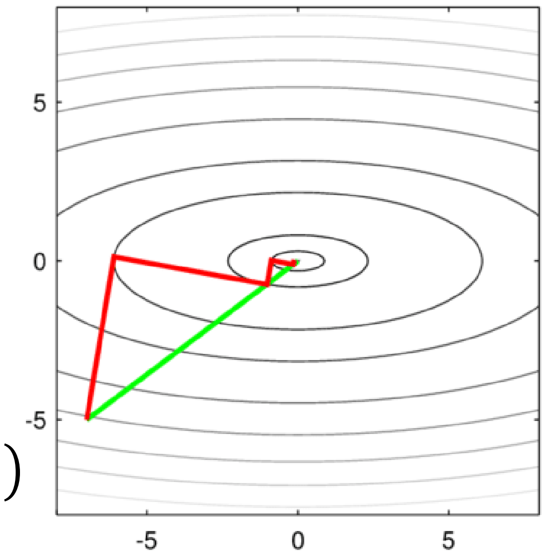
4 Nesterov Acceleration

Comparison: GD and Newton's Method

- gradient descent $$f\left(x^k\right) - f^* \leq (1-c)^k\left(f\left(x^{(0)}\right) - f^*\right)$$
  - linear convergence: stairwise
  - affine variant: line search is helpful
  - cheap for calculation
- Newton's method $$f\left(x^k\right) - f^* \leq \frac{4m^4}{C_1 L^2}\left(\frac{1}{2}\right)^{2^{k-\hat{k}+1}}$$
  - two phases
  - affine invariant: full step could be used
  - expensive: the inverse of the Hessian
- can we have something between gradient descent and Newton's method
  - acceptable additional computation to estimate second-order information

Momentum could help

- estimate the second-order information by several gradient
  - Hessian: the change (gradient) of gradient
  - $\nabla f(x^k) \approx \nabla f(x^{k-1}) + t\nabla^2 f(x^{k-1})$ we can assume it is unchanged
- gradient descent may have stairwise phenomena
  - calculating the current direction needs to consider about earlier steps
  - $\nabla f(x^{k+1})$ is different to $\nabla f(x^k)$ and similar to $\nabla f(x^{k-1})$
- using momentum information to lookahead gradient



$$z_{k+1} = x_k - t_k \nabla f(x_k)$$

$$x_{k+1} = x_k + \delta_k \underline{(x_k - x_{k-1})}$$

$$\delta_k \in [0,1)$$

Polyak's momentum

- using momentum information to lookahead gradient
- Polyak's momentum algorithm

$$x_{k+1} = x_k - t_k \nabla f(x_k) + \delta_k (x_k - x_{k-1})$$

$$f(x) = \frac{1}{2} x^\top S x \qquad \nabla f(x) = Sx$$

$$x_{k+1} = x_k - t_k z_k \qquad z_k = \nabla f(x_k) + \beta_k z_{k-1}$$

$$\Rightarrow \quad x_{k+1} = x_k - t_k z_k \qquad z_{k+1} - S x_{k+1} = \beta_k z_k$$

$$\Rightarrow \quad \begin{bmatrix} 1 & 0 \\ -S & 1 \end{bmatrix} \begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & -\boldsymbol{t_k} \\ 0 & \boldsymbol{\beta_k} \end{bmatrix} \begin{bmatrix} x_k \\ z_k \end{bmatrix}$$

$$Sq = \lambda q \qquad \nabla f(x) = S x_k = c_k \lambda q \qquad x_k = c_k q \qquad z_k = d_k q$$

$$\begin{bmatrix} 1 & 0 \\ -\lambda & 1 \end{bmatrix} \begin{bmatrix} c_{k+1} \\ d_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & -t_k \\ 0 & \beta_k \end{bmatrix} \begin{bmatrix} c_k \\ d_k \end{bmatrix}$$

$$\Leftarrow \quad \begin{bmatrix} c_{k+1} \\ d_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & -t_k \\ \lambda & \beta_k - \lambda \end{bmatrix}}_{R} \begin{bmatrix} c_k \\ d_k \end{bmatrix}$$

we can choose t_k and β_k to make R small

Polyak's momentum

- choose t_k and β_k to make R, the eigenvalue e_1, e_2, small
- $R = \begin{bmatrix} 1 & -t_k \\ \lambda & \beta_k - \lambda \end{bmatrix}$ is dependent on λ: $m \le \lambda \le M$

$$f\left(x^{(k)}\right) - p^* \le (1 - m/M)^k \left(f\left(x^{(0)}\right) - p^*\right)$$

gradient descent

$$\min_{\beta,t} \sup_{m \le \lambda \le M} \max\{e_1(\lambda,\beta,t), e_2(\lambda,\beta,t)\}$$

$$\Downarrow$$

$$t^* = \left(\frac{2}{\sqrt{M} + \sqrt{m}}\right)^2 \qquad \beta^* = \left(\frac{\sqrt{M} - \sqrt{m}}{\sqrt{M} + \sqrt{m}}\right)^2$$

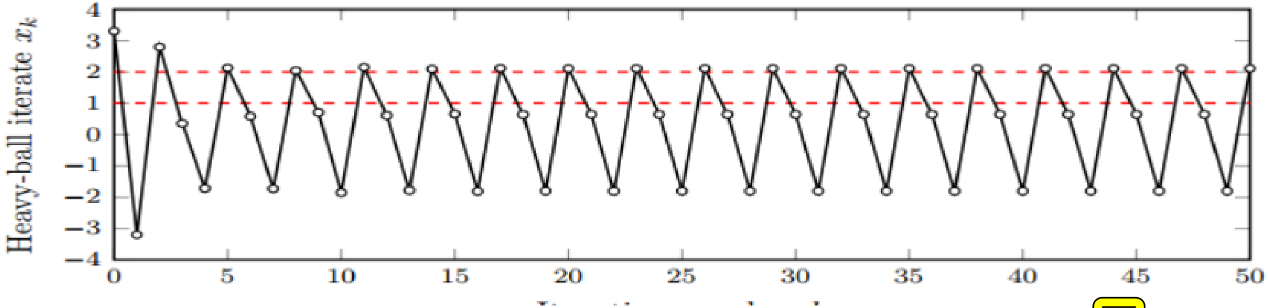
"miracles do not happen so much in math"
—— Gilbert Strang

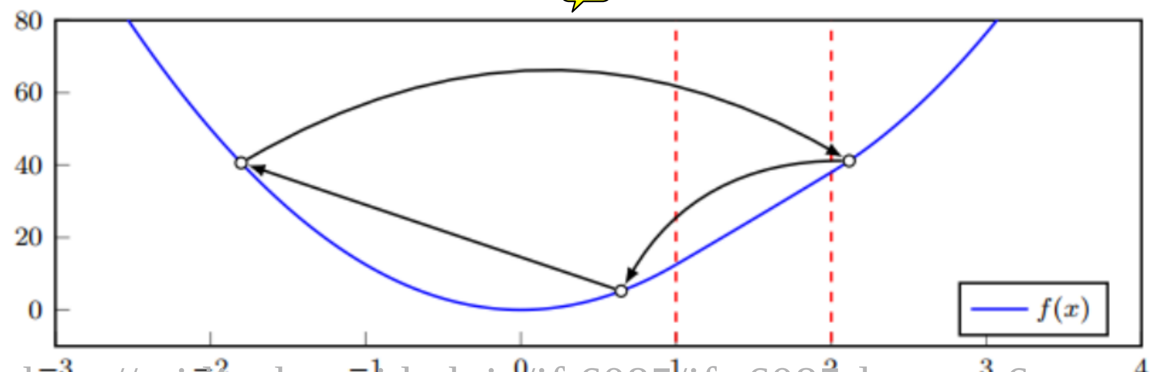
Polyak's momentum

- using momentum information to lookahead gradient
- Polyak's momentum algorithm

$$x_{k+1} = x_k - t_k \nabla f(x_k) + \delta_k (x_k - x_{k-1})$$

$$\nabla f(x) = \begin{cases} 25x, & \text{if } x < 1 \\ x + 24, & \text{if } 1 \le x < 2 \\ 25x - 24, & \text{otherwise} \end{cases}$$





http://mitliagkas.github.io/ift6085/ift-6085-lecture-6-notes.pdf

Nesterov Acceleration

- Polyak's momentum algorithm $x_{k+1} = x_k - t_k \nabla f(x_k) + \delta_k (x_k - x_{k-1})$

$$z_{k+1} = x_k - t_k \nabla f(x_k)$$

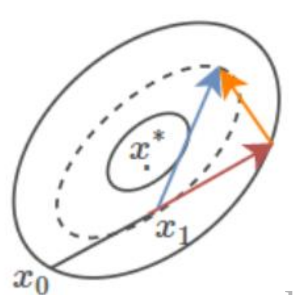
$$x_{k+1} = x_k + \delta_k (x_k - x_{k-1}) \qquad \delta_k \in [0,1)$$

- Nesterov acceleration $x_{k+1} \approx x_k - t_k \nabla f(x_k + \delta_k (x_k - x_{k-1})) + \delta_k (x^k - x^{k-1})$

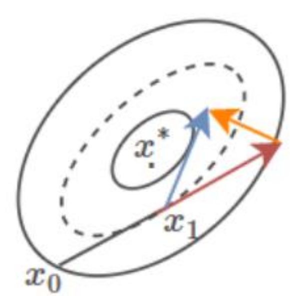
$$z_{k+1} = x_k - t_k \nabla f(x_k)$$

$$x_{k+1} = z_{k+1} + \delta_k (z_{k+1} - z_k) \qquad \delta_k \in [0,1)$$

Polyak's Momentum



Nesterov Momentum



evaluates the gradient after applying momentum

http://mitliagkas.github.io/ift6085/ift-6085-lecture-6-notes.pdf

Nesterov Acceleration

- Polyak's momentum algorithm $x_{k+1} = x_k - t_k \nabla f(x_k) + \delta_k (x_k - x_{k-1})$

$$z_{k+1} = x_k - t_k \nabla f(x_k)$$

$$x_{k+1} = x_k + \delta_k (x_k - x_{k-1}) \qquad \delta_k \in [0,1)$$

- Nesterov acceleration $x_{k+1} \approx x_k - t_k \nabla f(x_k + \delta_k (x_k - x_{k-1})) + \delta_k (x^k - x^{k-1})$

$$z_{k+1} = x_k - t_k \nabla f(x_k)$$

$$x_{k+1} = z_{k+1} + \delta_k (z_{k+1} - z_k) \qquad \delta_k \in [0,1)$$

- accelerated gradient descent can be viewed as:
  - a linear coupling of gradient descent and mirror descent (primal-dual)
  - a discretization of a certain second-order ODE https://arxiv.org/abs/1407.1537

https://arxiv.org/abs/1503.01243

Nesterov Acceleration

- Nesterov acceleration

$$y^{k+1} = x^k - \gamma^k \nabla f(x^k)$$

$$x^{k+1} = y^{k+1} + \mu^k (y^{k+1} - y^k) \quad \mu^k \in [0,1)$$

$$t^{k+1} = \frac{1+\sqrt{1+4t_k}}{2}, \qquad \mu^k = \frac{t^k - 1}{t^{k+1}}$$

$$f(x^k) - f(x^*) \le \frac{C\|x^0 - x^*\|^2}{k^2}$$

Table 1: Convergence rate for Gradient Descent & Nesterov Accelerated Gradient

Class of Function	GD	NAG
Smooth	$O(1/T)$	$O(1/T^2)$
Smooth & Strongly-Convex	$O\left(exp\left(-\frac{T}{\kappa}\right)\right)$	$O\left(exp\left(-\frac{T}{\sqrt{\kappa}}\right)\right)$

http://mitliagkas.github.io/ift6085/ift-6085-lecture-6-notes.pdf

SIAM J. IMAGING SCIENCES
Vol. 2, No. 1, pp. 183–202

A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems*

Amir Beck† and Marc Teboulle‡

Fast ISTA

$$\min_{x} \quad \lambda\|x\|_1 + \|Ax - B\|_2^2$$

- **ISTA**

$$x^{k+1} = S_\lambda\left(x^k - A^\top(Ax^k - B)\right)$$

$$S_\lambda = \operatorname*{argmin}_{\mathbf{x}} \left\{ g(\mathbf{x}) + \frac{L}{2}\left\| \mathbf{x} - \left(\mathbf{y} - \frac{1}{L}\nabla f(\mathbf{y})\right)\right\|^2 \right\}$$

- **FISTA**

$$y^{k+1} = S_\lambda\left(x^k - A^\top(B - Ax^k)\right)$$

$$t^{k+1} = \frac{1 + \sqrt{1 + 4t_k}}{2}$$

$$x^{k+1} = y^{k+1} + \left(\frac{t^k - 1}{t^{k+1}}\right)(y^{k+1} - y^k)$$


SIAM J. IMAGING SCIENCES
Vol. 2, No. 1, pp. 183–202

A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems*

Amir Beck† and Marc Teboulle‡

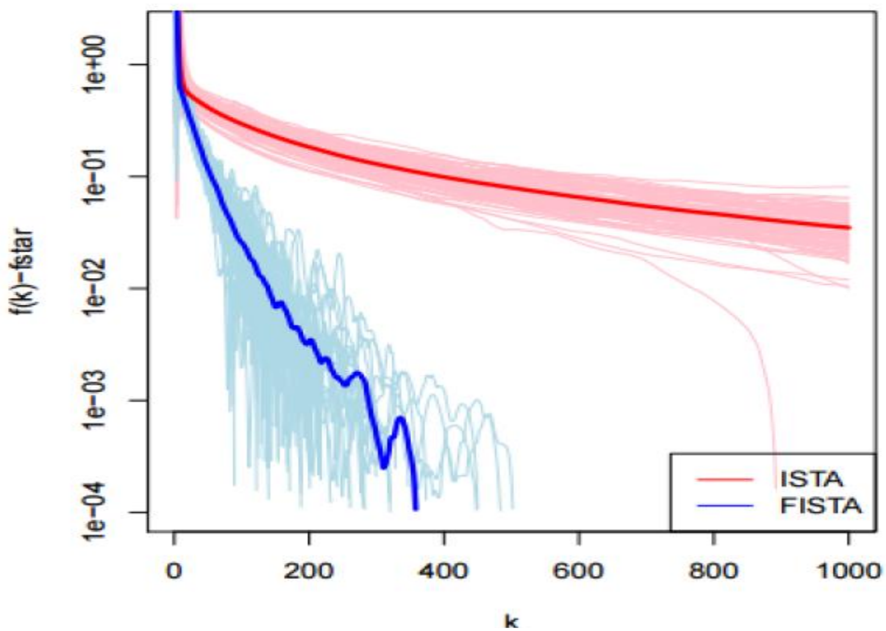
Fast ISTA

$$\min_{x} \quad \lambda\|x\|_1 + \|Ax - B\|_2^2$$

- **ISTA**

$$x^{k+1} = S_\lambda\left(x^k - A^\top(Ax^k - B)\ \right)$$

- **FISTA**



Home Work

- If your problem is a non-constrained problem
  - implement gradient descent and accelerate it by Nestrov method. Submit the code and draw the convergence curves: (1) $f(x^k)$ v. s. k and (a) $f(x^k)$ v. s. time
- If your problem is a constrained one
  - 9.10 (please submit your code as well)

THANKS

Lemmas

$$
\begin{aligned}
f\big(x - t\nabla f(x)\big) - f(y) &\le f\big(x - t\nabla f(x)\big) - f(x) + \nabla f(x)^\top (x-y) \\
&\le \nabla f(x)^\top (x - t\nabla f(x) - x) + \frac{1}{2t}\|x - t\nabla f(x) - x\|^2 + \nabla f(x)^\top (x-y) \\
&= -\frac{t}{2}\|\nabla f(x)\|^2 + \nabla f(x)^\top (x-y)
\end{aligned}
$$

$$y^{k+1} = x^k - \gamma^k \nabla f(x^k)$$

$$
\begin{aligned}
f(y^{k+1}) - f(y^k) = f\left(x^k - t\nabla f(x^k)\right) - f(y^k) &\le -\frac{t}{2}\left\|\nabla f(x^k)\right\|^2 + \nabla f(x^k)^\top (x^k - y^k) \\
&= -\frac{1}{2t}\left\|y^{k+1} - x^k\right\|^2 - \frac{1}{t}(y^{k+1} - x^k)^\top (x^k - y^k)
\end{aligned}
$$

$$f(y^{k+1}) - f(x^*) \le -\frac{1}{2t}\left\|y^{k+1} - x^k\right\|^2 - \frac{1}{t}(y^{k+1} - x^k)^\top (x^k - x^*)$$

Lemmas

$$f\big(x - t\nabla f(x)\big) - f(y) \le f\big(x - t\nabla f(x)\big) - f(x) + \nabla f(x)^\top (x-y)$$
$$\le \nabla f(x)^\top (x - t\nabla f(x) - x) + \frac{1}{2t}\|x - t\nabla f(x) - x\|^2 + \nabla f(x)^\top (x-y)$$
$$= -\frac{t}{2}\|\nabla f(x)\|^2 + \nabla f(x)^\top (x-y)$$

$$\Downarrow$$

$$y^{k+1} = x^k - \gamma^k \nabla f(x^k)$$

$$\frac{t}{2}\left\|\nabla f\left(x^k\right)\right\|^2 - \nabla f\left(x^k\right)^\top (x^k - y^k) = \frac{\gamma^k}{2}\left(\left\|\nabla f\left(x^k\right)\right\|^2 - \frac{2}{\gamma^k}\nabla f\left(x^k\right)^\top (x^k - y^k)\right)$$
$$= \frac{\gamma^k}{2}\left(\left\|\nabla f\left(x^k\right) - \frac{1}{\gamma^k}\nabla f\left(x^k\right)^\top (x^k - y^k)\right\|^2 - \frac{1}{{\gamma^k}^2}\left\|x^k - y^k\right\|^2\right)$$
$$= \frac{\gamma^k}{2}\left(\frac{1}{{\gamma^k}^2}\left\|y^k - (x^k - \gamma^k \nabla f\left(x^k\right)\right\|^2 - \frac{1}{{\gamma^k}^2}\left\|x^k - y^k\right\|^2\right)$$

$$f\left(y^{k+1}\right) - f\left(y^k\right) \le \frac{1}{2t}\left(\|y - (x - \alpha\nabla f(x)\|^2 - \left\|x^k - y^k\right\|^2\right)$$

Convergence Analysis

$$y_{k+1} = x_k - t_k \nabla f(x_k) \qquad x_{k+1} = z_{k+1} + \delta_k (y_{k+1} - y_k) \qquad \delta_k \in [0,1)$$

$$\begin{aligned} f\big(x - t\nabla f(x)\big) - f(y) &\le f\big(x - t\nabla f(x)\big) - f(x) + \nabla f(x)^\top (x-y) \\ &\le \nabla f(x)^\top (x - t\nabla f(x) - x) + \frac{1}{2t}\|x - t\nabla f(x) - x\|^2 + \nabla f(x)^\top (x-y) \\ &= -\frac{t}{2}\|\nabla f(x)\|^2 + \nabla f(x)^\top (x-y) \end{aligned}$$

$$\begin{aligned} f(y^{k+1}) - f(y^k) = f\left(x^k - t\nabla f(x^k)\right) - f(y^k) &\le -\frac{t}{2}\left\|\nabla f(x^k)\right\|^2 + \nabla f(x^k)^\top (x^k - y^k) \\ &= -\frac{1}{2t}\left\|y^{k+1} - y^k\right\|^2 - \frac{1}{t}(y^{k+1} - y^k)^\top (x^k - y^k) \end{aligned}$$

$$f(y^{k+1}) - f(x^*) \le -\frac{1}{2t}\left\|y^{k+1} - y^k\right\|^2 - \frac{1}{t}(y^{k+1} - y^k)^\top (x^k - x^*)$$