高级工程数学

2122(1)

沈超敏

计算机科学与技术学院

cmshen@cs.ecnu.edu.cn

Topic 2: Condition

Three acronyms:

- FONC (First Order Necessary Condition)
- SONC (Second Order Necessary Condition)
- SOSC (Second Order Sufficient Condition)

$\boxed{x^* \text{ is a local minimizer of } f}$

Necessary

Sufficient

interior pt $\nabla f(\boldsymbol{x}^*) = 0, \nabla^2 f(\boldsymbol{x}^*) \geqslant 0 \quad \nabla f(\boldsymbol{x}^*)$

$\nabla f(\boldsymbol{x}^*) = 0, \nabla^2 f(\boldsymbol{x}^*) > 0$

general pt $\nabla f(\boldsymbol{x}^*) \cdot \boldsymbol{d} \geqslant 0 \quad \forall d$

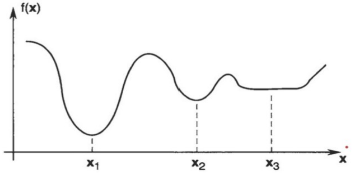
If $\nabla f(\boldsymbol{x}^*) \cdot \boldsymbol{d} = 0$ for some $\boldsymbol{d}$, $\boldsymbol{d}^\top \nabla^2 f(\boldsymbol{x}^*) \boldsymbol{d} \geqslant 0$

Topic 2: Condition

★ Conditions for Local Minimizers

- Global minimizer $\boldsymbol{x}^*$: $f(\boldsymbol{x}) \geqslant f(\boldsymbol{x}^*) \quad \forall \boldsymbol{x} \in \Omega \setminus \{\boldsymbol{x}^*\}$
- Strict global minimizer $\boldsymbol{x}^*$: $f(\boldsymbol{x}) > f(\boldsymbol{x}^*) \quad \forall \boldsymbol{x} \in \Omega \setminus \{\boldsymbol{x}^*\}$
- Mathematically, "near" can be characterized as $\|\boldsymbol{x} - \boldsymbol{x}^*\| < \varepsilon$
- $\boldsymbol{x}^*$ is a local minimizer if $\exists \varepsilon > 0$, s.t.

$$f(\boldsymbol{x}) \geqslant f(\boldsymbol{x}^*) \quad \forall \boldsymbol{x} \in \Omega \setminus \{\boldsymbol{x}^*\} \,\&\, \|\boldsymbol{x} - \boldsymbol{x}^*\| < \varepsilon$$



Topic 2: Condition

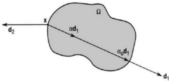
★ First Order Necessary Condition

Theorem. $\boldsymbol{x}^*$ is a local minimizer of f over Ω. Then for any feasible direction $\boldsymbol{d}$ at $\boldsymbol{x}^*$, we have

$$\nabla f\left(\boldsymbol{x}^*\right) \cdot \boldsymbol{d} \geqslant 0$$

Explanation. ① feasible direction $\boldsymbol{d}$ at a point $\boldsymbol{x} \in \Omega$ is a direction so that: starting from $\boldsymbol{x}$ and moving towards $\boldsymbol{d}$ remains in Ω.

Math language: $\exists \alpha_0 > 0$ s.t. $\boldsymbol{x} + \alpha \boldsymbol{d} \in \Omega, \forall \alpha \in [0, \alpha_0]$



② $\nabla f\left(\boldsymbol{x}^*\right) \cdot \boldsymbol{d}$: inner product of two vectors.

Also write as $\boldsymbol{d}^\top \nabla f\left(\boldsymbol{x}^*\right)$ or $\left(\nabla f\left(\boldsymbol{x}^*\right), \boldsymbol{d}\right)$, $\langle \nabla f(\boldsymbol{x}^*), \boldsymbol{d} \rangle$

$\frac{\partial f}{\partial \boldsymbol{d}} \triangleq \nabla f \cdot \boldsymbol{d}$ is the directional derivative when $\|\boldsymbol{d}\| = 1$

- ③ Define $\phi(\alpha) = f\left(\boldsymbol{x}^* + \alpha \boldsymbol{d}\right)$ for $\alpha \in [0, \alpha_0]$, then

$$\phi'(0) = \begin{cases} \lim_{\alpha \to 0^+} \frac{\phi(\alpha) - \phi(0)}{\alpha} = \lim_{\alpha \to 0^+} \frac{f(\boldsymbol{x}^* + \alpha \boldsymbol{d}) - f(\boldsymbol{x}^*)}{\alpha} & \text{Def} \\ \nabla f\left(\boldsymbol{x}^*\right) \cdot \boldsymbol{d} & \text{Chain rule} \end{cases}$$

Topic 2: Condition

- Proof. Let $\boldsymbol{d}$ be any feasible direction at $\boldsymbol{x}^*$. Define $\phi(\alpha) = f(\boldsymbol{x}^* + \alpha\boldsymbol{d})$
  Then $f(\boldsymbol{x}^* + \alpha\boldsymbol{d}) - f(\boldsymbol{x}^*) = \phi(\alpha) - \phi(0) = \phi'(0)\alpha + o(\alpha)$
  $$= \langle \nabla f(\boldsymbol{x}^*) \cdot \boldsymbol{d} \rangle \alpha + o(\alpha)$$
  If $\boldsymbol{x}^*$ is a local minimizer,
  (i.e., $\exists \varepsilon$, s.t. $f(\boldsymbol{x}) \geqslant f(\boldsymbol{x}^*)$, $\forall \boldsymbol{x} \in \Omega \backslash \{\boldsymbol{x}^*\} \,\&\, \|\boldsymbol{x} - \boldsymbol{x}^*\| < \varepsilon$)
  for sufficiently small α (e.g. $\|\alpha\boldsymbol{d}\| < \varepsilon$), $f(\boldsymbol{x}^* + \alpha d) - f(\boldsymbol{x}^*) \geqslant 0$
  then $\phi'(0) = \nabla f(\boldsymbol{x}^*) \cdot \boldsymbol{d} \geqslant 0$
- FONC Two possibilities for a given feasible direction $\boldsymbol{d}$.
  $$\begin{cases} \nabla f(\boldsymbol{x}^*) \cdot \boldsymbol{d} > 0 \text{ then } f(\boldsymbol{x}^* + \alpha\boldsymbol{d}) > f(\boldsymbol{x}^*) \text{ for all sufficienty small } \alpha > 0 \\ \nabla f(\boldsymbol{x}^*) \cdot \boldsymbol{d} = 0 \text{ . check second-order derivative} \end{cases}$$

Topic 2: Condition

★ **Second Order Necessary Condition**

- Theorem If $\boldsymbol{x}^*$ is a local minimizer of f over Ω, and there exists a feasible direction $\boldsymbol{d}$ at $\boldsymbol{x}^*$ s.t. $\nabla f\left(\boldsymbol{x}^*\right) \cdot \boldsymbol{d}=0$, then

$$\boldsymbol{d}^{\top} \nabla^2 f\left(\boldsymbol{x}^*\right) \boldsymbol{d} \geqslant 0$$

- Proof   Consider $\phi(\alpha)=f\left(\boldsymbol{x}^*+\alpha \boldsymbol{d}\right)$ and its Taylor series at $\alpha=0$

$$\phi(\alpha)=\phi(0)+\phi'(0)\alpha+\frac{1}{2}\phi''(0)\alpha^2+o\left(\alpha^2\right).$$

$$\text{as } \phi'(0)=\nabla f\left(\boldsymbol{x}^*\right) \cdot \boldsymbol{d}=0$$

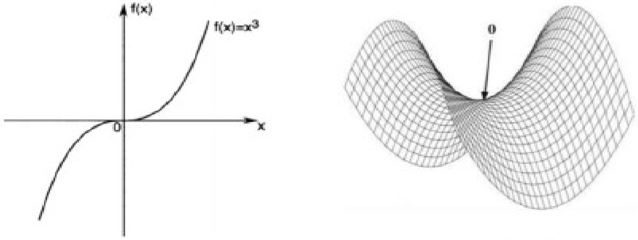
So we have $\phi(\alpha)-\phi(0)=\frac{1}{2}\phi''(0)\alpha^2+o\left(\alpha^2\right)$. Written in terms f is $f\left(\boldsymbol{x}^*+\alpha \boldsymbol{d}\right)-f\left(\boldsymbol{x}^*\right)=\frac{\alpha^2}{2}\boldsymbol{d}^{\top} \nabla^2 f\left(x^*\right) \boldsymbol{d}+o\left(\alpha^2\right)$.
If $\boldsymbol{d}^{\top} \nabla^2 f\left(\boldsymbol{x}^*\right) \boldsymbol{d}<0$, then for sufficiently small α (how small?)
which contradicts that x^* is a local minimizer.
so, $\boldsymbol{d}^{\top} \nabla^2 f\left(\boldsymbol{x}^*\right) \boldsymbol{d} \geq 0$.

Topic 2: Condition

- Corollary $\boldsymbol{x}^*$ is an interior local minimizer of f. Then
- FONC $\nabla f(\boldsymbol{x}^*) = 0$
- SONC $\boldsymbol{d}^\top \nabla^2 f(\boldsymbol{x}^*)\, \boldsymbol{d} \geqslant 0, \quad \forall \boldsymbol{d} \in \mathbb{R}^n$

  Examples. 6.3 (p.86), 6.5 (p.89)

  Necessary conditions are not sufficient



$f(x) = x^3,\ f'(x) = 3x^2,\ f''(x) = 6x$

$f(\boldsymbol{x}) = x_1^2 - x_2^2$

$\mathbf{0}$ is a saddle point: $\nabla f(\mathbf{0}) = \mathbf{0}$ but neither a local minimizer nor maximizer

By SONC, $\mathbf{0}$ is not a local minimizer!

Video 32 结束

Topic 2: Condition

★ Second Order Sufficient Condition

- Th 6.3 (SOSC) $f \in C^2(\Omega), \boldsymbol{x}^* \in \Omega$ is an interior point.

  Suppose that (1) $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$; (2) $\nabla^2 f(\boldsymbol{x}^*) \succ 0$.

  Then $\boldsymbol{x}^*$ is a strict local minimizer of f.

  Pf. $\nabla^2 f(x^*) \succ 0 \Leftrightarrow \lambda_{\min}\left(\nabla^2 f(x^*)\right) > 0$

  (Prove by diagonaligation of $\nabla^2 f(\boldsymbol{x}^*) = Q^\top \wedge Q\Big)$

  For a feasible direction $\boldsymbol{d} \neq 0$, define $\phi(\alpha) = f(\boldsymbol{x}^* + \alpha\boldsymbol{d})$

$$
\begin{aligned}
f(\boldsymbol{x}^* + \alpha\boldsymbol{d}) - f(\boldsymbol{x}^*) = \phi(\alpha) - \phi(0) &= \frac{1}{2}\phi''(0)\alpha^2 + o\left(\alpha^2\right) \\
&= \frac{1}{2}\boldsymbol{d}^\top \nabla^2 f(\boldsymbol{x}^*)\,\boldsymbol{d}\alpha^2 + o\left(\alpha^2\right) \\
&\geqslant \frac{1}{2}\lambda_{\min}\|d\|^2\alpha^2 + o\left(\alpha^2\right) > 0
\end{aligned}
$$

  if α is sufficiently small.

Topic 2: Condition

x^* is a local minimizer of f

Necessary

interior pt $\nabla f\left(\boldsymbol{x}^*\right)=0, \nabla^2 f\left(\boldsymbol{x}^*\right) \geqslant 0 \quad \nabla f\left(\boldsymbol{x}^*\right)$

general pt $\nabla f\left(\boldsymbol{x}^*\right) \cdot \boldsymbol{d} \geqslant 0 \quad \forall d$

If $\nabla f\left(\boldsymbol{x}^*\right) \cdot \boldsymbol{d}=0$ for some $\boldsymbol{d}, \boldsymbol{d}^{\top} \nabla^2 f\left(\boldsymbol{x}^*\right) \boldsymbol{d} \geqslant 0$

Sufficient

$\nabla f\left(\boldsymbol{x}^*\right)=0, \nabla^2 f\left(\boldsymbol{x}^*\right)>0$

Video 33 结束

Ch 7. One-Dimensional Search Methods

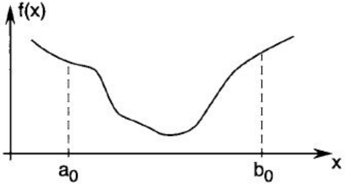


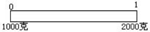
Figure 7.1 Unimodal function.

Ch 7. One-Dimensional Search Methods

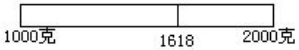
§2 单 因 素

我们知道，钢要用某种化学元素来加强其强度，太少不好，太多也不好.例如，碳太多了成为生铁，碳太少了成为熟铁，都不成钢材，每吨要加多少碳才能达到强度最高？假定已从理论上算出)每吨在1000克到2000克之间.普通的方法是加1001克，1002克，……，做下去，做了一千次以后，才能发现最好的选择，这种方法称为均分法.做一千次实验既浪费时间又浪费原材料.为了迅速找出最优方案，我们建议以下的“折迭纸条法”.

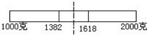
请牢记一个数0.618.



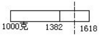
用一个有刻度的纸条表达1000～2000克，在这纸条长度的0.618的地方划一条线，在这条线所指示的刻度做一次实验，也就是按1618克做一次实验.



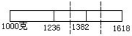
然后把纸条对中迭起，前一线落在另一层上的地方，再划一条线，这条线在1382克处，再按1382克做一次实验.

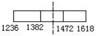


两次实验进行比较，如果1382克的好一些，我们在1618处把纸条的右边一段剪掉，得：



(如果1618克比较好，则在1382克处剪掉左边一段).再依中对折起来，又可划出一条线在1236克处：





依1236克做实验，再和1382克的结果比较.如果，仍然是1382克好，则在1236处剪掉左边：

再依中对折，找出一个试点是1472，按1472克做实验，做出后再剪掉一段，等等.注意每次留下的纸条的长度是上次长度的0.618(留下的纸条长=0.618×上次长).

就这样，实验、分析、再实验、再分析，矛盾的解决和又出现的过程中，一次比一次地更加接近所需要的加入量，直到所能达到的精度.

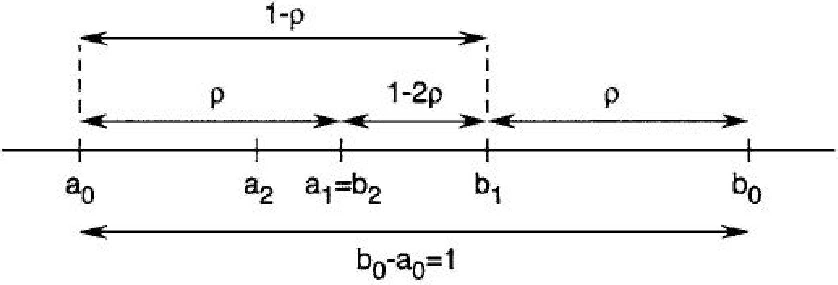


Figure 7.4 Finding value of ρ resulting in only one new evaluation of

Video 34 结束

7.3 Fibonacci Method

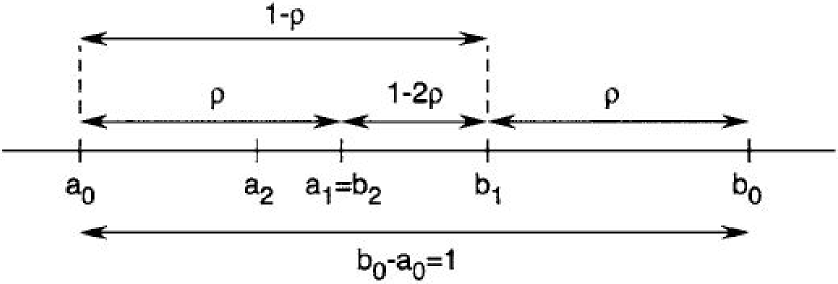
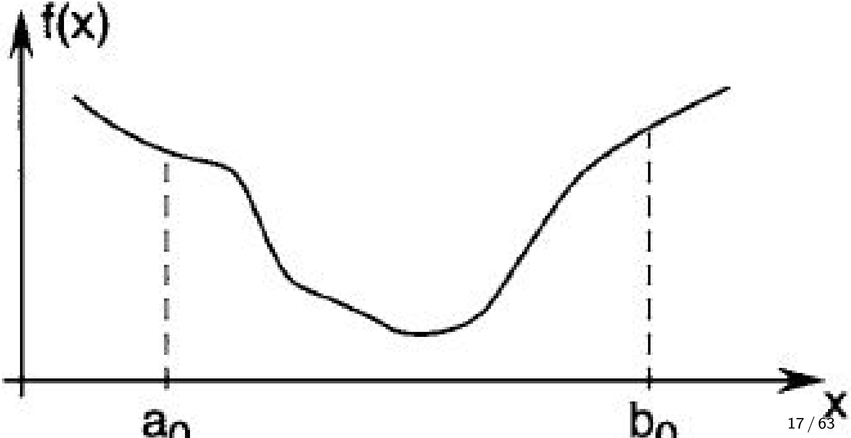


Figure 7.4 Finding value of ρ resulting in only one new evaluation of

Ch 7. One-Dimensional Search Methods

7.4 Bisection Method



Video 35 结束

Topic 3: Ch 8. Gradient Methods

$\nabla f(x)$ is the direction of maximum rate of increase of f at x.

$-\nabla f(x)$ is the direction of maximum rate of decrease of f at x.

Lemma: (Cauchy-Schwarz inequality) For $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$

$$\boldsymbol{a} \cdot \boldsymbol{b} = (\boldsymbol{a}, \boldsymbol{b}) = \sum_{i=1}^{n} a_i b_i \leqslant \left(\sum_{i=1}^{n} a_i^2\right)^{\frac{1}{2}} \left(\sum_{i=1}^{n} b_i^2\right)^{\frac{1}{2}} = \|\boldsymbol{a}\| \cdot \|\boldsymbol{b}\|$$

Apply to $(\nabla f, \boldsymbol{d}) \leqslant \|\nabla f\| \cdot \|\boldsymbol{d}\| = \|\nabla f\|$ if $\|\boldsymbol{d}\| = 1$

Equality holds $\Leftrightarrow \boldsymbol{d} = \frac{\nabla f}{\|\nabla f\|}$

Therefore $-\nabla f(\boldsymbol{x})$ is the max-rate descending direction. When $\nabla f(\boldsymbol{x}) \neq \mathbf{0}$, for α sufficiently small, $f(\boldsymbol{x} - \alpha \nabla f(\boldsymbol{x})) < f(\boldsymbol{x})$.

Topic 3: Gradient Methods

Gradient Descent Algorithm:

Start from $\boldsymbol{x}^0$ for $k = 0, 1, 2, \cdots$ till converge

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha_k \nabla f\left(\boldsymbol{x}^k\right)$$

Ideal condition: $\left\|\nabla f\left(\boldsymbol{x}^k\right)\right\| = 0$

Practical conditions:

gradient condition $\left\|\nabla f\left(\boldsymbol{x}^k\right)\right\| < \varepsilon$

success objective condition $\frac{\left|f\left(\boldsymbol{x}^{k+1}\right) - f\left(\boldsymbol{x}^k\right)\right|}{\left|f\left(\boldsymbol{x}^k\right)\right|} < \varepsilon$

successive point difference $\frac{\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\right\|}{\left\|\boldsymbol{x}^k\right\|} < \varepsilon$

Replace denominator by $\max\left\{1, |f\left(\boldsymbol{x}^k\right)|\right\}$ or $\max\{1, \|\boldsymbol{x}^k\|\}$ to avoid division by tiny numbers.

Topic 3: Gradient Methods

Step size:

1. Exact line search. Expensive and not worth

2. Fixed estimate value.

3. Line search (Ch. 7).

具体地

1. Exact or “best” for $\phi_k(\alpha) = f\left(x_k - \alpha \nabla f\left(x_k\right)\right)$

$$\alpha_k = \underset{\alpha \geqslant 0}{\arg\min}\, \phi_k(\alpha)$$

This is called Steepest Descent method.

Topic 3: Gradient Methods

2. Based on properties of f, choose a fixed value

small: converges slow

large: may diverge faster

convergence efficiency

3. Several practical line search algorithms

• Golden section • Newton's method

• Fibonacci • Secant method

• Bisection • Bracketing

Topic 3: Gradient Methods

Quadratic Programming

For a symmetric and positive definite (SPD) matrix Q, i.e., $Q > 0$,

we can define a new norm $\|\boldsymbol{x}\|_Q = \left(\boldsymbol{x}^\top Q\boldsymbol{x}\right)^{\frac{1}{2}}$ and a new inner product

$$(\boldsymbol{x}, \boldsymbol{y})_Q \triangleq (Q\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}, Q\boldsymbol{y}) = \boldsymbol{y}^T Q\boldsymbol{x}$$

Let $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_Q^2 - (\boldsymbol{b}, \boldsymbol{x})$. Consider non-constrained, convex, and smooth optimization problem:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$$

$$\nabla f(\boldsymbol{x}) = Q\boldsymbol{x} - \boldsymbol{b}, \quad \nabla^2 f(\boldsymbol{x}) = Q > 0$$

Topic 3: Quadratic Programming

As f is strictly convex, the global minimum pt is $\nabla f(x) = \mathbf{0}$

Solve $Q\boldsymbol{x} - \boldsymbol{b} = \mathbf{0}$ to get the solution $\boldsymbol{x} = Q^{-1}\boldsymbol{b}$.

Why not computing $Q^{-1}\boldsymbol{b}$ directly?

1. Q^{-1} is expensive, $O(n^3)$ complexity.

2. Want a method for more general problems.

Steepest descent for quadratic programming

$$\phi_k(\alpha) = f\left(\boldsymbol{x}_k - \alpha \nabla f\left(\boldsymbol{x}_k\right)\right), \quad \alpha_k = \underset{\alpha}{\arg\min} \phi_k(\alpha) \quad \phi_k'\left(\alpha_k\right) = 0$$

$$\begin{aligned} \phi_k'(\alpha) &= \langle \nabla f(\boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k)), -\nabla f\left(\boldsymbol{x}_k\right) \rangle \\ &= \langle Q\left(\boldsymbol{x}_k - \alpha \nabla f\left(\boldsymbol{x}_k\right)\right) - \boldsymbol{b}, -\nabla f\left(\boldsymbol{x}_k\right) \rangle \\ &= \alpha \|\nabla f(\boldsymbol{x}_k)\|_Q^2 - \langle Q\boldsymbol{x}_k - \boldsymbol{b}, \nabla f\left(\boldsymbol{x}_k\right) \rangle = \alpha \left\|\nabla f\left(\boldsymbol{x}_k\right)\right\|_Q^2 - \left\|\nabla f\left(\boldsymbol{x}_k\right)\right\|^2 \end{aligned}$$

Video 36 结束

Topic 3: Quadratic Programming

If we denote by $\boldsymbol{g}_k = \nabla f\left(\boldsymbol{x}_k\right)$, then we can write as

$$\alpha_k = \left\|\boldsymbol{g}_k\right\|^2 / \left\|\boldsymbol{g}_k\right\|_Q^2$$

As $\lambda_{\min}\|\boldsymbol{v}\|^2 \leqslant \boldsymbol{v}^\top Q \boldsymbol{v} \leqslant \lambda_{\max}\|\boldsymbol{v}\|^2$

(which can be proved first for diagonal matrix and then $Q = U^T \Lambda U$)

so $\frac{1}{\lambda_{\max}} \leqslant \alpha_k \leqslant \frac{1}{\lambda_{\min}}$

Video 37 结束

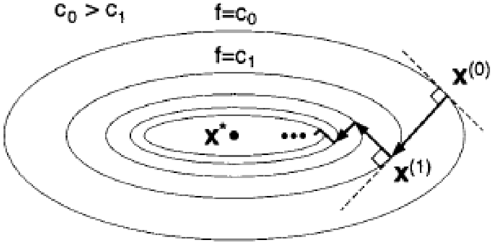


Figure 8.7 Steepest descent method in search for minimizer in a narrow valley.

Topic 4: Newton's method

Consider $$\min_{x \in \mathbb{R}^n} f(x)$$

- $n = 1$ $\quad x_{k+1} = x_k - \left(f''(x_k)\right)^{-1} f'(x_k)$
- $n > 1$ $\quad x_{k+1} = x_k - \left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k)$

Another form ① solve $\quad \nabla^2 f(x_k)\, d_k = -\nabla f(x_k)$

② update $\quad x_{k+1} = x_k + d_k$

Remark. Do not require $\nabla^2 f(x_k) \succ 0$ only needs non-singular (invertible).

Namely, Newton's method also works for non-convex optimization problems.

but may not find local min.

Topic 4: Newton's method

Pro. 1. Convergences super-fast (quadratic rate).

2. Affine invariant.

Con. 1. Local convergence. Require $\|x_0 - x^*\|$ is small enough.

2. Computational cost.

Form Hessian matrix $O(n^2)$. Compute $(\nabla^2 f)^{-1}$: $O(n^3)$.

Topic 4: Newton's method

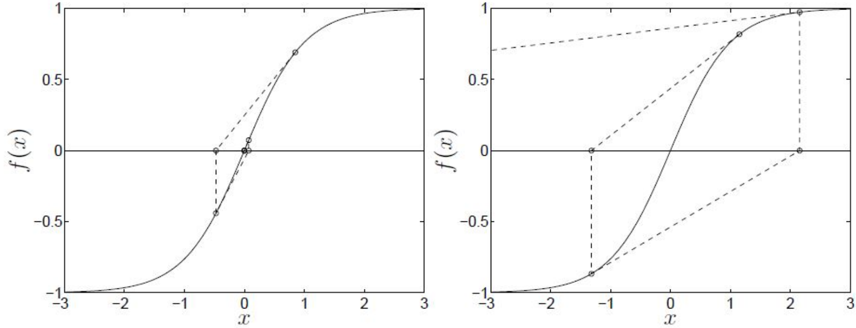


Figure 12.3 The solid line in the left plot is $f(x) = (e^x - e^{-x})/(e^x + e^{-x})$. The dashed line and the circles indicate the iterates in Newton's method for solving $f(x) = 0$, starting at $x^{(0)} = 0.85$ (left) and $x^{(0)} = 1.15$ (right). In the first case the method converges rapidly to $x^\star = 0$. In the second case it does not converge.

Video 38 结束

Topic 4: Newton's method

Derivation. Given current approximate x_k, approximates f by its quadratic Taylor series

$$f(x) \approx f_q(x; x_k) := f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2}\langle \nabla^2 f(x_k)(x - x_k), x - x_k \rangle$$

$$\min_{x \in \mathbb{R}^n} f(x) \rightsquigarrow \min_{x \in \mathbb{R}^n} f_q(x; x_k) \rightsquigarrow \nabla f_q(x_{k+1}; x_k) = 0.$$

$$\Big\downarrow$$

$$\nabla f_q(x; x_k) = \nabla f(x_k) + \nabla^2 f(x_k)(x - x_k)$$

Newton's method

Topic 4: Newton's method

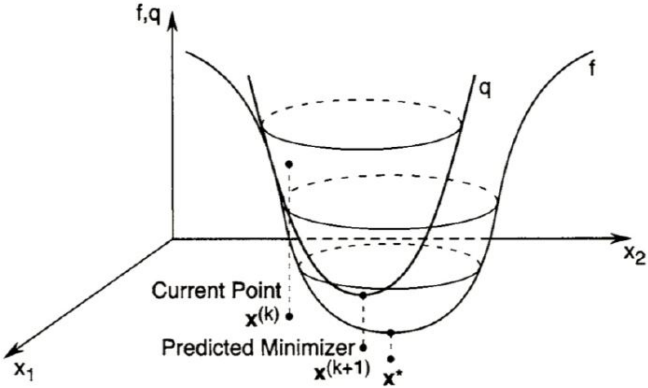


Figure 9.1 Quadratic approximation to the objective function using first and second derivatives.

Video 39 结束

Topic 4: Newton's method

Convergence Analysis.

Theorem. Suppose $f \in C^3$. x^* is a critical pt, i.e. $\nabla f(x^*) = 0$, and $\nabla^2 f(x^*)$ is invertible. Then for all x_0 sufficiently close to x^*, Newton's method is well defined for all k, and $\|x_{k+1} - x^*\| \leqslant c \|x_k - x^*\|^2 \quad \forall k = 0, 1, 2, \cdots$

Proof. Denote by $F(x) = \nabla^2 f(x)$. Then $\det F(x) \in C^1$. As $\det F(x^*) \neq 0$, for sufficiently small ε, $\det F(x) \neq 0$, $\forall \|x - x^*\| < \varepsilon$. So $F(x)$ is invertible.

Topic 4: Newton's method

Furthermore $\|F^{-1}(x)\| \leqslant C$, $\forall \|x - x^*\| < \varepsilon$.

Assume x_k satisfies $\|x_k - x^*\| < \varepsilon$, then $F^{-1}(x_k)$ exists and $\|F^{-1}(x_k)\| \leqslant C$.

Then $x_{k+1} - x^* = x_k - x^* - (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$

$$= (\nabla^2 f(x_k))^{-1} [\nabla^2 f(x_k)(x_k - x^*) - \nabla f(x_k)].$$

We apply first order Taylor expansion to $\nabla f(x^*)$ at x_k to get

$$\nabla f(x^*) = \nabla f(x_k) + \nabla^2 f(x_k)(x^* - x_k) + O(\|x_k - x^*\|^2)$$

Note that $\nabla f(x^*) = 0$ and the sign change, we have

$$\nabla^2 f(x_k)(x_k - x^*) - \nabla f(x_k) = O(\|x_k - x^*\|^2).$$

Therefore $\|x_{k+1} - x^*\| \leqslant C \|(\nabla^2 f(x_k))^{-1}\| \, \|x_k - x^*\|^2$

$$\leqslant C_1 \|x_k - x^*\|^2$$

Again by choosing ε sufficiently small s.t. $C_1\varepsilon^2 < \varepsilon$, we conclude

$$\|x_{k+1} - x^*\| < \varepsilon \text{ and } F(x_{k+1})^{-1} \text{ exists and } \|F(x_{k+1})^{-1}\| \leqslant C.$$

So if ε is small enough and $\|x_0 - x^*\| < \varepsilon$, all $\|x_k - x^*\| < \varepsilon$ and

$$\|x_{k+1} - x^*\| \leqslant C_1 \|x_k - x^*\|^2 \quad \forall\, k = 0, 1, 2, \cdots$$

which implies the local quadratic convergence. #.

Topic 4: Newton's method

Modification of Newton's method.

Newton's method may not be a descent method, i.e. $f(x_{k+1}) > f(x_k)$ is possible (e.g. x^* is a local maximum). Have to restrict to stricktly convex functions.

Lemma. Assume $\nabla^2 f(x) > 0$, $\forall x$. If $\nabla f(x_k) \neq 0$, then Newton's direction $d_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$ is a descent direction in the sense that $f(x_k + \alpha d_k) < f(x_k)$ for sufficiently small α.

Proof. Let $\phi(\alpha) = f(x_k + \alpha d_k)$. Then $\phi'(\alpha) = (\nabla f(x_k + \alpha d_k), d_k)$ and

$$\phi'(0) = -(\nabla f(x_k), (\nabla^2 f(x_k))^{-1} \nabla f(x_k)) = -(g_k, g_k)_Q < 0 \quad \text{where}$$

$$g_k = \nabla f(x_k), \quad Q = (\nabla^2 f(x_k))^{-1} > 0.$$

Then for sufficiently small α, $f(x_k + \alpha d_k) = \phi(\alpha) < \phi(0) = f(x_k)$. #.

Topic 4: Newton's method

For convex functions, we can use the following modification

1. Compute d_k by solving $\nabla^2 f(x_k)\, d_k = -\nabla f(x_k)$.
2. Find $\alpha_k = \arg\min f(x_k + \alpha d_k)$ by line search.
3. Update $x_{k+1} = x_k + \alpha_k d_k$.

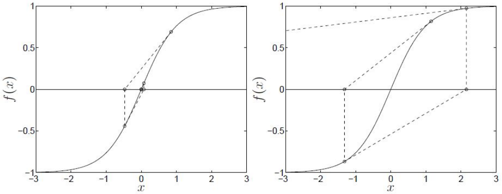


Figure 12.3 The solid line in the left plot is $f(x) = (e^x - e^{-x})/(e^x + e^{-x})$. The dashed line and the circles indicate the iterates in Newton's method for solving $f(x) = 0$, starting at $x^{(0)} = 0.85$ (left) and $x^{(0)} = 1.15$ (right). In the first case the method converges rapidly to $x^\star = 0$. In the second case it does not converge.

Topic 4: Newton's method

What if $\nabla^2 f$ is not SPD? Note that for non-convex functions, the gradient method $x_{k+1} = x_k - \alpha_k I \nabla f(x_k)$ is always a descent method. This motivates the Levenberg-Marquardt modification

$$\boxed{x_{k+1} = x_k - \alpha_k \left(\nabla^2 f(x_k) + \mu_k I \right)^{-1} \nabla f(x_k),}$$

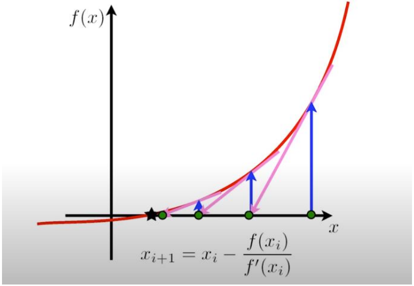
where $\mu_k > 0$ is chosen s.t. $\nabla^2 f(x_k) + \mu_k I > 0$ and $\alpha_k > 0$ is a step size.

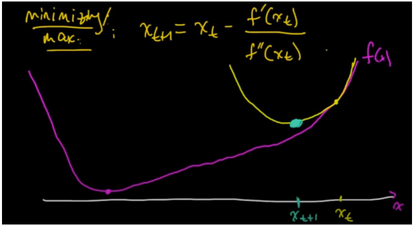
It is a mixture of Newton and gradient methods:

- $\mu_k = 0$. Newton's method.
- $\mu_k \to +\infty$. Gradient method.

Video 40 结束

Topic 5: Quasi-Newton's method





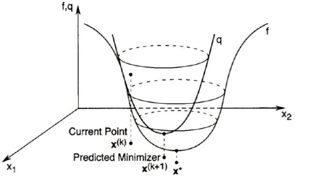
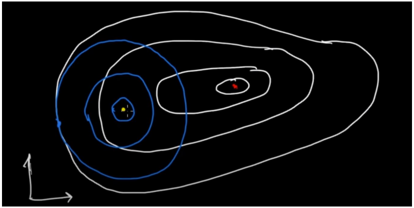


Figure 9.1 Quadratic approximation to the objective function using first and second derivatives.



Topic 5: Quasi-Newton's method

Search directions

- Gradient method $\qquad d_k = -g_k$
- Conjugate gradient method $\qquad d_k = -g_k + \beta d_{k-1}$
- Newton's method $\qquad d_k = -(\nabla^2 f(x_k))^{-1} g_k$
- Quasi-Newton method $\qquad d_k = -\cancel{H_k}\, g_k$

B_K

Topic 5: Quasi-Newton's method

Algorithm

Start with x_0, g_0, H_0.

for $k = 0, 1, 2, \cdots$

$$d_k = -\cancel{H}_k^{B} g_k ;$$

$$\alpha_k = \arg\min_{\alpha} f(x_k + \alpha d_k) ;$$

$$x_{k+1} = x_k + \alpha_k d_k ;$$

$$g_{k+1} = \nabla f(x_{k+1}) ;$$

★ $\boxed{\text{Update } \cancel{H}_{k+1}}$ ★ many choices

B

end

Topic 5: Quasi-Newton's method

Construction of $\overset{B}{\cancel{H}}_k$

Two consideration ① $\overset{B}{\cancel{H}}_k \approx (\nabla^2 f(x_k))^{-1}$ approximation to inverse of Hessian

② $\underset{B}{\cancel{H}}_k g_k$ is efficient to compute.

Approximation. Recall for 1-d line search,

$$f''(x_k) \approx \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$$

the quoient only suitable for scalars

$$\nabla f(x_k) - \nabla f(x_{k-1}) = \nabla^2 f(\xi)\,(x_k - x_{k-1}) \quad \text{by mean value theorem}$$

$$\underbrace{(\nabla^2 f(\xi))^{-1}}_{\underset{B}{\cancel{H}}_k}\underbrace{(\nabla f(x_k) - \nabla f(x_{k-1}))}_{\Delta g_{k-1}} = \underbrace{x_k - x_{k-1}}_{\Delta x_{k-1}}$$

$$\boxed{B\,\cancel{H}_{k+1}\,\Delta g_k = \Delta x_k} \qquad (*)$$

Topic 5: Quasi-Newton's method

Rank One Correction $\quad B\,\cancel{H}_{k+1} = \overset{B}{\cancel{H}_k} + z_k z_k^T$

$z_k z_k^T$ is a $n \times n$ matrix with rank 1.

$$(z_k z_k^T) v = z_k \underbrace{(z_k^T v)}_{\text{number}} = (z_k, v)\, z_k \quad \text{is easy to compute.}$$

$$\underset{B}{\cancel{H}_{k+1}}\, g_{k+1} = \underset{\text{simple diagonal}}{\overset{B}{\cancel{H}_0}}\, g_{k+1} + \underbrace{\sum_{i=0}^{k} z_i (z_i, g_{k+1})}_{k \text{ vector products}}$$

When k is large, we can restart the iteration by choosing $\overset{B}{\cancel{H}_{k+1}} = \overset{B}{\cancel{H}_0}$.

Next we use (*) to determine z_k.

Topic 5: Quasi-Newton's method

$$\underset{B}{\cancel{H}}_{k+1}\Delta g_k = \boxed{\underset{B}{\cancel{H}}_k \Delta g_k + z_k (z_k, \Delta g_k) = \Delta x_k} \qquad (1)$$

So $$z_k = \frac{\Delta x_k - \overset{B}{\cancel{H}}_k \Delta g_k}{\underbrace{(z_k, \Delta g_k)}_{\text{number}}}$$ still unknown.

Multiply with Δg_k to (1) to get

$$(\underset{B}{\cancel{H}}_k \Delta g_k, \Delta g_k) + (z_k, \Delta g_k)^2 = (\Delta x_k, \Delta g_k)$$

which implies $$(z_k, \Delta g_k)^2 = (\Delta x_k, \Delta g_k) - (\overset{B}{\cancel{H}}_k \Delta g_k, \Delta g_k)$$

$$= (\Delta x_k - \underset{B}{\cancel{H}}_k \Delta g_k, \Delta g_k)$$

Topic 5: Quasi-Newton's method

Question: RHS may not be positive! That's why we add a_k.

We can skip a_k and skip the sqrt to compute $(z_k, \Delta g_k)$. Continue to

$$\boxed{z_k z_k^T = \frac{(\Delta x_k - \overset{B}{\cancel{H}}_k \Delta g_k)(\Delta x_k - \overset{B}{\cancel{H}}_k \Delta g_k)^T}{(\Delta x_k - \underset{B}{\cancel{H}}_k \Delta g_k, \Delta g_k)}}$$

Notational simplification $\Delta e_k = \Delta x_k - \underset{B}{\cancel{H}}_k \Delta g_k$.

$u v^T = u \otimes v$: $\otimes$ tensor product. $u \otimes v = (v \otimes u)^T$. $u \otimes u$ is symmetric

$$\boxed{\underset{B}{\cancel{H}}_{k+1} = \underset{B}{\cancel{H}}_k + \frac{\Delta e_k \otimes \Delta e_k}{(\Delta e_k, \Delta g_k)}}$$

Video 41 结束

Topic 5: Quasi-Newton's method

Example 11.1 Let

$$f(x_1, x_2) = x_1^2 + \frac{1}{2}x_2^2 + 3.$$

Apply the rank one correction algorithm to minimize f. Use $\boldsymbol{x}^{(0)} = [1, 2]^\top$ and $\boldsymbol{H}_0 = \boldsymbol{I}_2$ (2×2 identity matrix).

We can represent f as

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{x} + 3.$$

Thus,

$$\boldsymbol{g}^{(k)} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}^{(k)}.$$

Because $\boldsymbol{H}_0 = \boldsymbol{I}_2$,

$$\boldsymbol{d}^{(0)} = -\boldsymbol{g}^{(0)} = [-2, -2]^\top.$$

The objective function is quadratic, and hence

$$\begin{aligned} \alpha_0 = \arg\min_{\alpha \geq 0} f(\boldsymbol{x}^{(0)} + \alpha \boldsymbol{d}^{(0)}) &= -\frac{\boldsymbol{g}^{(0)\top}\boldsymbol{d}^{(0)}}{\boldsymbol{d}^{(0)\top}\boldsymbol{Q}\boldsymbol{d}^{(0)}} \\ &= \frac{[2, 2] \begin{bmatrix} 2 \\ 2 \end{bmatrix}}{[2, 2] \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix}} = \frac{2}{3}, \end{aligned}$$

and thus

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} + \alpha_0 \boldsymbol{d}^{(0)} = \left[-\frac{1}{3}, \frac{2}{3}\right]^\top.$$

Topic 5: Quasi-Newton's method

We then compute

$$\Delta \boldsymbol{x}^{(0)} = \alpha_0 \boldsymbol{d}^{(0)} = \left[-\frac{4}{3}, -\frac{4}{3}\right]^\top,$$
$$\boldsymbol{g}^{(1)} = \boldsymbol{Q}\boldsymbol{x}^{(1)} = \left[-\frac{2}{3}, \frac{2}{3}\right]^\top,$$
$$\Delta \boldsymbol{g}^{(0)} = \boldsymbol{g}^{(1)} - \boldsymbol{g}^{(0)} = \left[-\frac{8}{3}, -\frac{4}{3}\right]^\top.$$

Because

$$\Delta \boldsymbol{g}^{(0)\top}(\Delta \boldsymbol{x}^{(0)} - \boldsymbol{H}_0 \Delta \boldsymbol{g}^{(0)}) = \left[-\frac{8}{3}, -\frac{4}{3}\right] \begin{bmatrix} \frac{4}{3} \\ 0 \end{bmatrix} = -\frac{32}{9},$$

we obtain

$$\boldsymbol{H}_1 = \boldsymbol{H}_0 + \frac{(\Delta \boldsymbol{x}^{(0)} - \boldsymbol{H}_0 \Delta \boldsymbol{g}^{(0)})(\Delta \boldsymbol{x}^{(0)} - \boldsymbol{H}_0 \Delta \boldsymbol{g}^{(0)})^\top}{\Delta \boldsymbol{g}^{(0)\top}(\Delta \boldsymbol{x}^{(0)} - \boldsymbol{H}_0 \Delta \boldsymbol{g}^{(0)})} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix}.$$

Therefore,

$$\boldsymbol{d}^{(1)} = -\boldsymbol{H}_1 \boldsymbol{g}^{(1)} = \left[\frac{1}{3}, -\frac{2}{3}\right]^\top$$

and

$$\alpha_1 = -\frac{\boldsymbol{g}^{(1)\top}\boldsymbol{d}^{(1)}}{\boldsymbol{d}^{(1)\top}\boldsymbol{Q}\boldsymbol{d}^{(1)}} = 1.$$

We now compute

$$\boldsymbol{x}^{(2)} = \boldsymbol{x}^{(1)} + \alpha_1 \boldsymbol{d}^{(1)} = [0, 0]^\top.$$

Topic 5: Quasi-Newton's method

Unfortunately, the rank one correction algorithm is not very satisfactory, for several reasons. First, the matrix $\boldsymbol{H}_{k+1}$ that the rank one algorithm generates may not be positive definite (see Example 11.2 below) and thus $\boldsymbol{d}^{(k+1)}$ may not be a descent direction. This happens even in the quadratic case (see Example 11.10). Furthermore, if

$$\Delta \boldsymbol{g}^{(k)}(\Delta \boldsymbol{x}^{(k)} - \boldsymbol{H}_k \Delta \boldsymbol{g}^{(k)})$$

is close to zero, then there may be numerical problems in evaluating $\boldsymbol{H}_{k+1}$.

Video 42 结束

Least Square Problems

Problem $Ax = b$, A: $m \times n$ matrix, $b \in \mathbb{R}^m$ is given, find $x \in \mathbb{R}^n$
$m \geq n$

Since # eqns $\geq$ # unknowns, no solution in general.

Instead, consider optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) := \frac{1}{2} \| Ax - b \|^2.$$

Topic 6: Solving Linear Equations

Solution. $\nabla f(x) = A^T A x - A^T b$. $\nabla^2 f(x) = A^T A \succeq 0$.

Lemma. When A is full rank, i.e. $\text{rank}(A) = n$, then $\nabla^2 f = A^T A \succ 0$.

Proof. Let $A = \begin{pmatrix} | & | & & | \\ a_1 & a_2 & \cdots & a_n \\ | & | & & | \end{pmatrix}$, where a_i is the i-th column vector.

Interpret $Au = \sum_{i=1}^{n} u_i a_i$ as linear combination of column vectors.

$\text{rank}(A) = n$ means $\{a_1, \cdots, a_n\}$ are linearly independent. Therefore $Au = 0$ implies $u = 0$. Consequently $(A^T A u, u) = (Au, Au) > 0$, $\forall u \in \mathbb{R}^n, u \neq 0$.

#

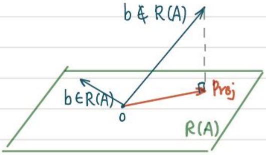
As a quadratic and stricktly convex function, the global minimum of f is the solution to $\nabla f(x) = 0$, $x^* = \underbrace{(A^T A)^{-1}}_{\text{inverse is not easy to compute!}} A^T b$.

Topic 6: Solving Linear Equations

Geometry. Define the column space $C(A) = \text{span}\{a_1, a_2, \cdots, a_n\}$

and the range of A, $R(A) = \{Ay \mid y \in \mathbb{R}^n\}$.

By the formulae, $Ay = \Sigma y_i a_i$, we know $C(A) = R(A)$.



Case 1. $b \in R(A)$.

Then there exists a x s.t. $Ax = b$.

When $\text{rank}(A) = n$, the solution is unique.

Case 2. $b \notin R(A)$.

Then no x s.t. $Ax = b$.

Find the projection to the subspace $R(A)$, i.e. $Ax^* = \text{Proj}_{R(A)} b$.

By definition, $(Ax^*, v) = (b, v) \quad \forall v \in R(A) \quad (1)$.

Note that v is in the form $v = Ay$, $y \in R(A)$.

So (1) is equivalent to $(Ax^*, Ay) = (b, Ay) \quad \forall y \in \mathbb{R}^n$.

$$(A^T A x^*, y) = (A^T b, y) \quad \forall y \in \mathbb{R}^n.$$

We get the formuale $x^* = (A^T A)^{-1} A^T b$.

By the property of projection, $\|b - Ax^*\| = \min_{y \in \mathbb{R}^n} \|b - Ay\|$.

Topic 6: Solving Linear Equations

Projector.

$R^{\perp}(A) = \{c \in \mathbb{R}^m, (c, Ay) = 0, \forall y \in \mathbb{R}^n\}$

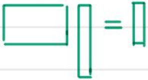
$N(A^T) = \{c \in \mathbb{R}^m, A^T c = 0\}$

- $\mathrm{Proj}_{R(A)} = A(A^TA)^{-1}A^T$,
- $\mathrm{Proj}_{R^{\perp}(A)} = \mathrm{Proj}_{N(A^T)} = I - A(A^TA)^{-1}A^T$

Verify $N(A^T) = R^{\perp}(A)$

Topic 6: Solving Linear Equations

$A_{m\times n}$, $m \le n$. $\text{rank}(A) = m$.



more unknowns than equations.

There may exists infinite many solutions. Which one to pick?

Consider constrained optimization problem

$$\min_{\{x\in\mathbb{R}^n,\ Ax=b\}} \|x\|^2 . \qquad (1)$$

Theorem. The solution to (1) is $x^* = A^T(AA^T)^{-1}b$.

Proof. Write $x = x^* + e$. $Ax = b \Leftrightarrow Ae = 0$.

$\|x\|^2 = \|x^*+e\|^2 = \|x^*\|^2 + \|e\|^2 + 2\underbrace{\cancel{(x^*,e)}}_{0}$ since $(x^*,e) = (A^T(AA^T)^{-1}b, e)$

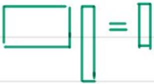
$= ((AA^T)^{-1}b, Ae) = 0.$

$\min_{Ax=b} \|x\|^2 = \|x^*\|^2 + \cancel{\min_{Ae=0} \|e\|^2} = \|x^*\|^2.$ #.

Topic 6: Solving Linear Equations

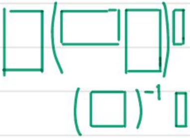
Remark.

$Ax = b$



$Ax = b$

$x^* = A^T(AA^T)^{-1}b$



$x^* = (A^TA)^{-1}A^Tb$

Geometry. $S = \{x \in \mathbb{R}^n : Ax = b\}$ is an affine space not a linear space

$x_1, x_2 \in S$ but $x_1 - x_2 \notin S$ as $A(x_1 - x_2) = b - b = 0$.

Topic 6: Solving Linear Equations

Kaczmarz算法

$$A = \begin{bmatrix} \boldsymbol{a}_1^{\mathrm{T}} \\ \boldsymbol{a}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{a}_m^{\mathrm{T}} \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

初始点 $\boldsymbol{x}_0$

$i = 0$ $\qquad 0 < \mu < 2$

for j in $1 \ldots m$

$$\boldsymbol{x}_{im+j} = \boldsymbol{x}_{im+j-1} + \mu(b_j - \boldsymbol{a}_j^{\mathrm{T}} \boldsymbol{x}_{im+j-1})\left(\frac{\boldsymbol{a}_j}{\boldsymbol{a}_j^{\mathrm{T}} \boldsymbol{a}_j}\right)$$

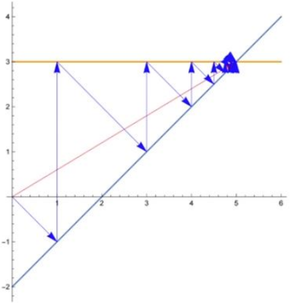
$i = i + 1$

Kaczmarz算法.pdf

```
x, p =
Kaczmarz([1 -1;
          0  1],
         [2, 3],
         [0, 0])
```

$$A = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$



Topic 6: Solving Linear Equations

Kaczmarz 算法

Kaczmarz算法.pdf

```
function Kaczmarz(A,
                  b,
                  x;
                  μ = 1,
                  ϵ = 0.001)
  #Check if the dimensions match
  if !(ndims(A) == 2 && ndims(b) == 1) # MUST be double '&'
     error("A should be a matrix (dims = 2) and b should be a vector (dim = 1)")
  end
  r, c = size(A)
  l = length(b)
  if l != r
     error("# of rows in A should match with # of elements in b")
  end
  if r > c
     error("Kaczmarz algorithm is not applicable!")
  end
  xk = x
  xn = x
  i = 0
  A0 = mapslices(r -> r .* (1 /(r'*r)), A, dims=[2])
  pts = []
  push!(pts, xn)
  while i == 0 || norm(xn .- xk) > ϵ
      i  = i + 1
      for j in 1:r
         xk = xn
         xn = xn .+ μ*(b[j] - A[j, 1:end]'*xn) .* A0[j,1:end]
         push!(pts, xn)
      end
  end
  return (xn, pts)
```

Video 43 结束