Machine Learning

Lecture 5: Gradient Descent Method

王浩

信息科学与技术学院

Email: wanghao1@shanghaitech.edu.cn

本节内容

- Gradient Descent Method
- Introduction to Mathematical Optimization
- Convergence of GD Method
- Newton method

梯度下降法
(Gradient Descent Method)

Gradient Descent Method

Batch/Vanilla Gradient Descent Method

$$\min_{\theta_0,\theta_1} \quad J(\theta_0,\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(\theta_0 + \theta_1 x^i - y^i)^2$$

Iteratively, starting with some (θ_0^0, θ_1^0), keep changing (θ_0^0, θ_1^0) to reduce $J(\theta_0,\theta_1)$ (why we can do this?) until we hopefully end up at a minimum.

$$\theta_0^{k+1} \leftarrow \theta_0^k - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0^k, \theta_1^k)$$
$$\theta_1^{k+1} \leftarrow \theta_1^k - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0^k, \theta_1^k)$$

Simultaneously update, using (θ_0^k, θ_1^k) to compute $(\theta_0^{k+1}, \theta_1^{k+1})$

Gradient Descent Method

Least Squared Error Minimization

$$\min_{\boldsymbol{\theta}} \frac{1}{2}\|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 = \frac{1}{2}\sum_{i=1}^{m}((\mathbf{x}^i)^\top \boldsymbol{\theta} - y^i)^2$$

Batch/Vanilla Gradient Descent Method

$$\boldsymbol{\theta}^{k+1} \leftarrow \boldsymbol{\theta}^k - \alpha \nabla J(\boldsymbol{\theta}^k)$$

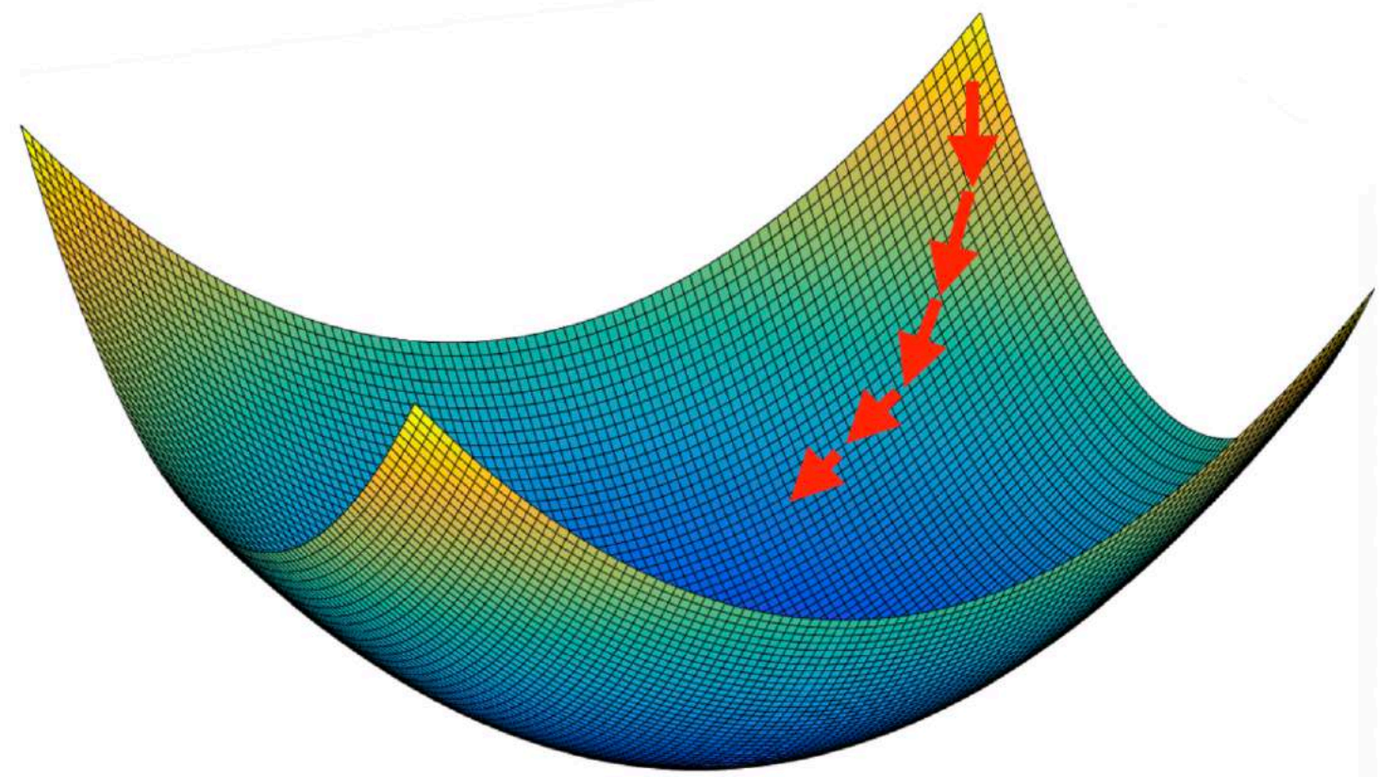
Stepsize/Learning rate: α, may be varying α^k

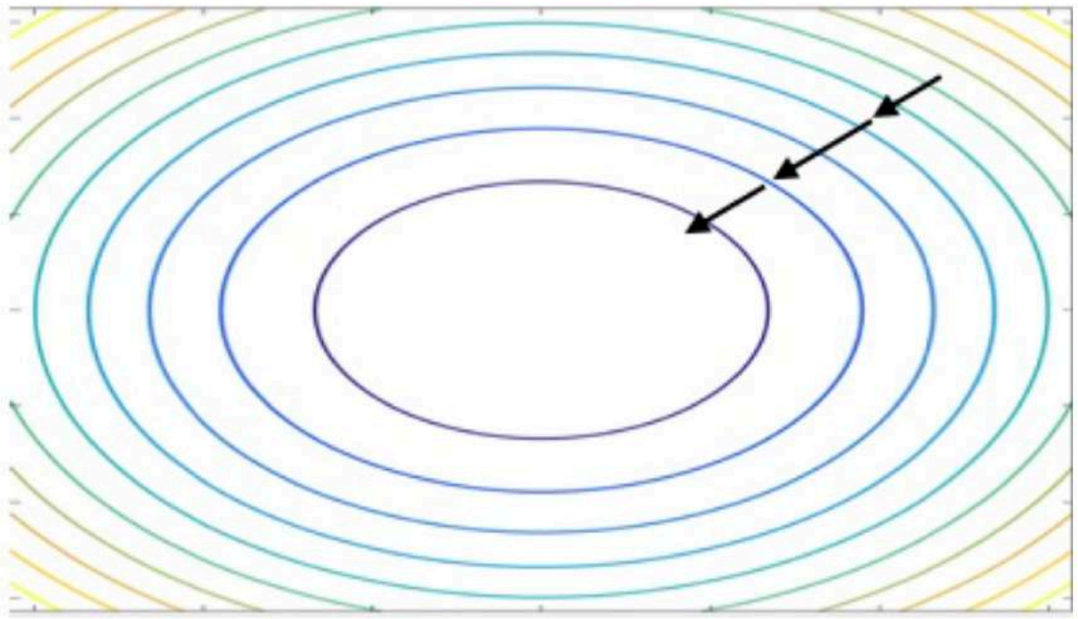
Until "convergence".

Key issue: how to select learning rate? when to terminate? how fast are our algorithms?

Gradient Descent Method

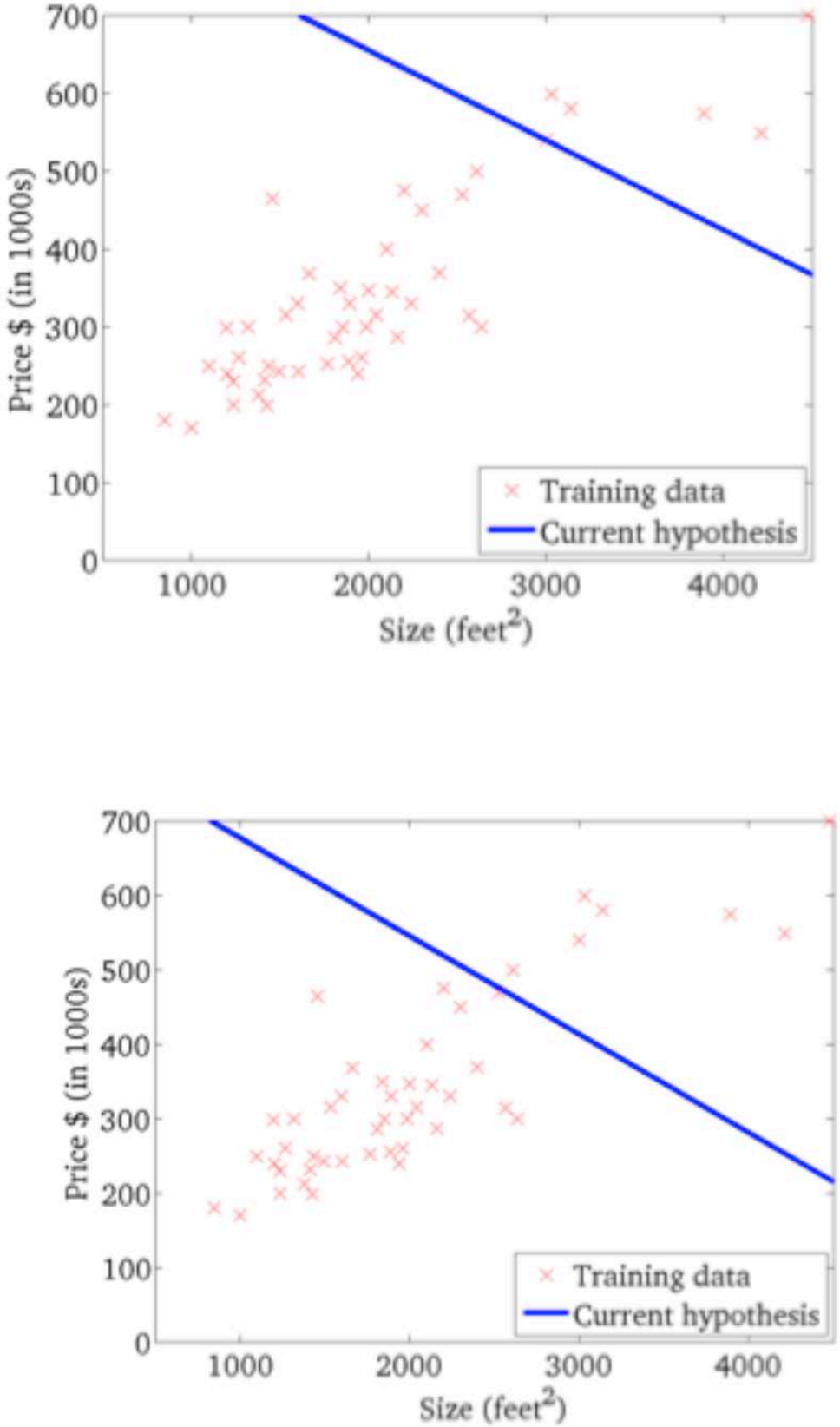
Notice that the least squared error is convex with respect to (θ_0, θ_1)





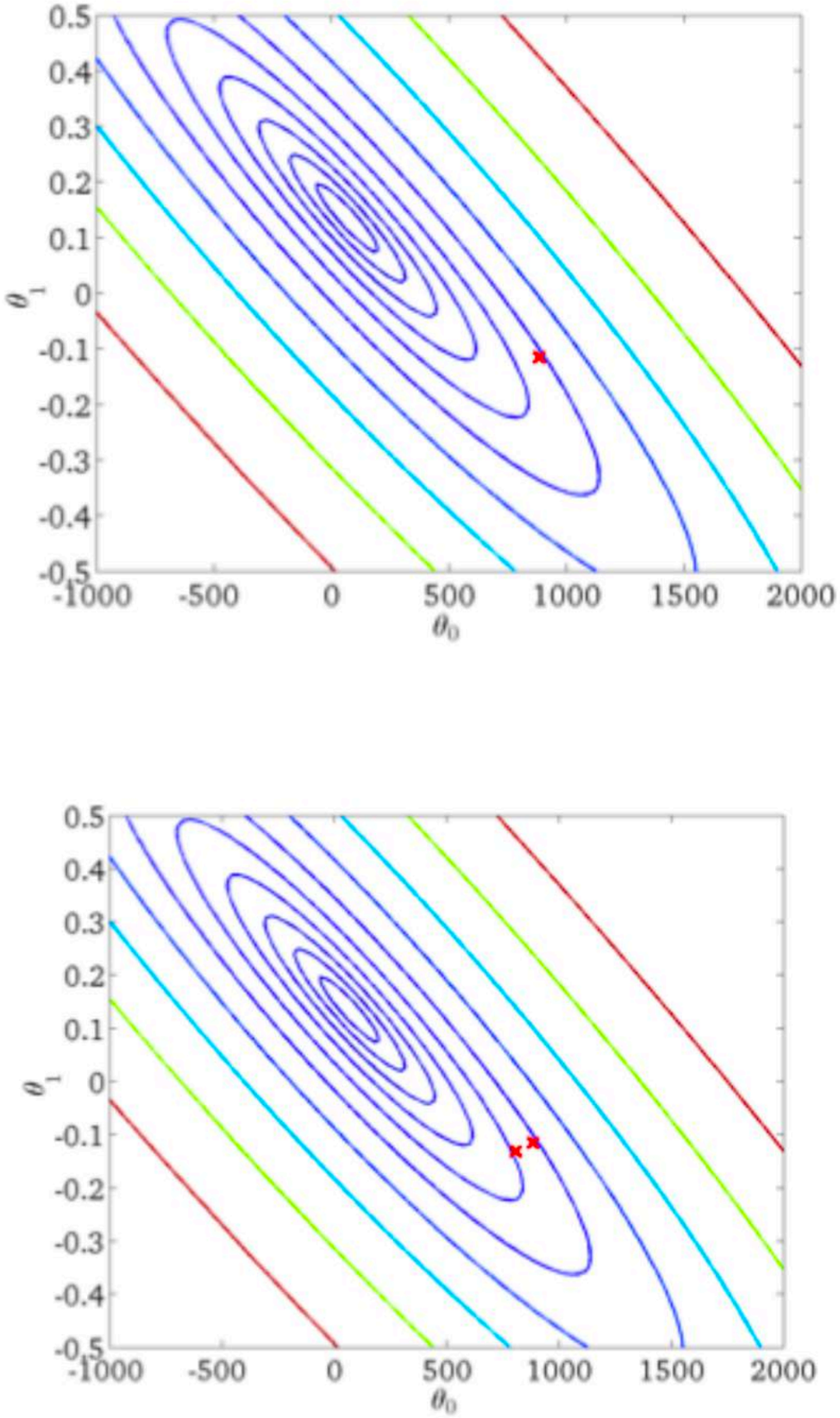
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



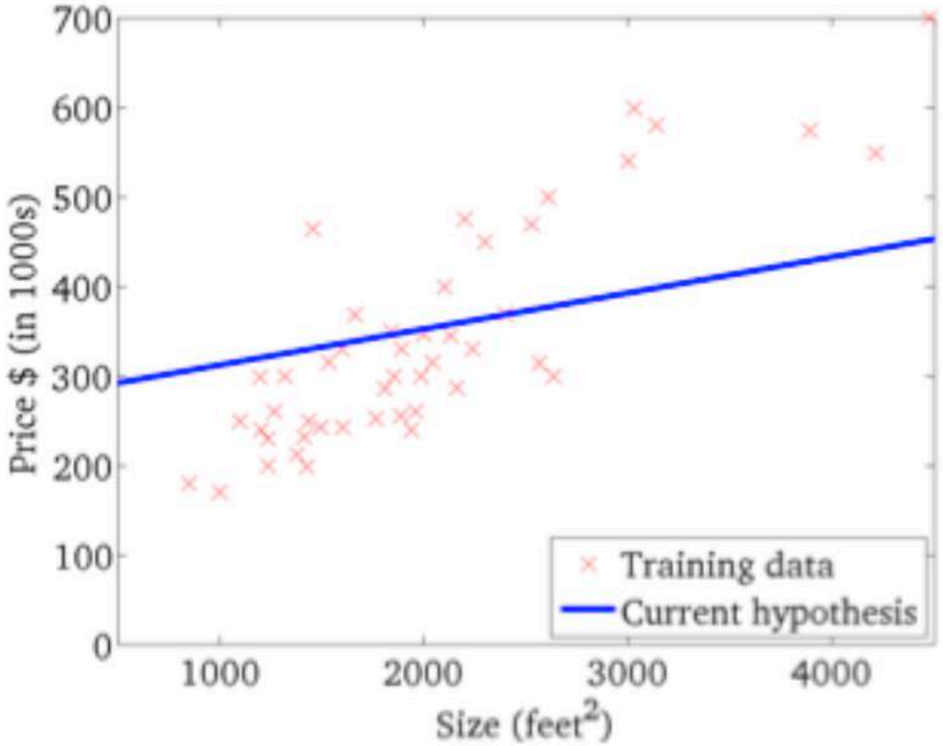
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



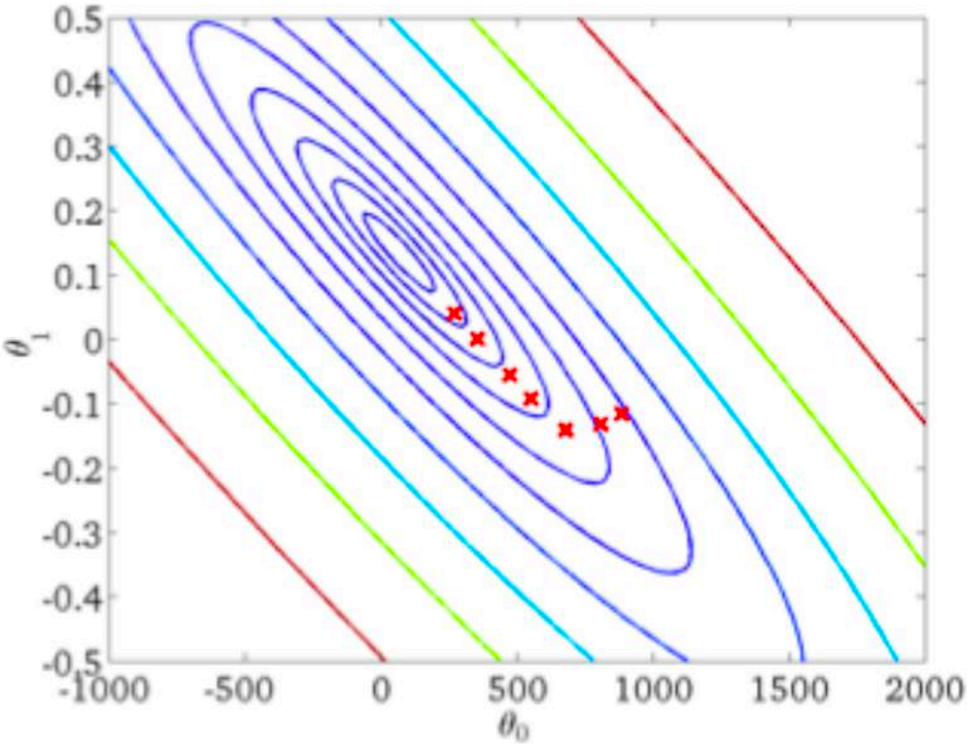
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



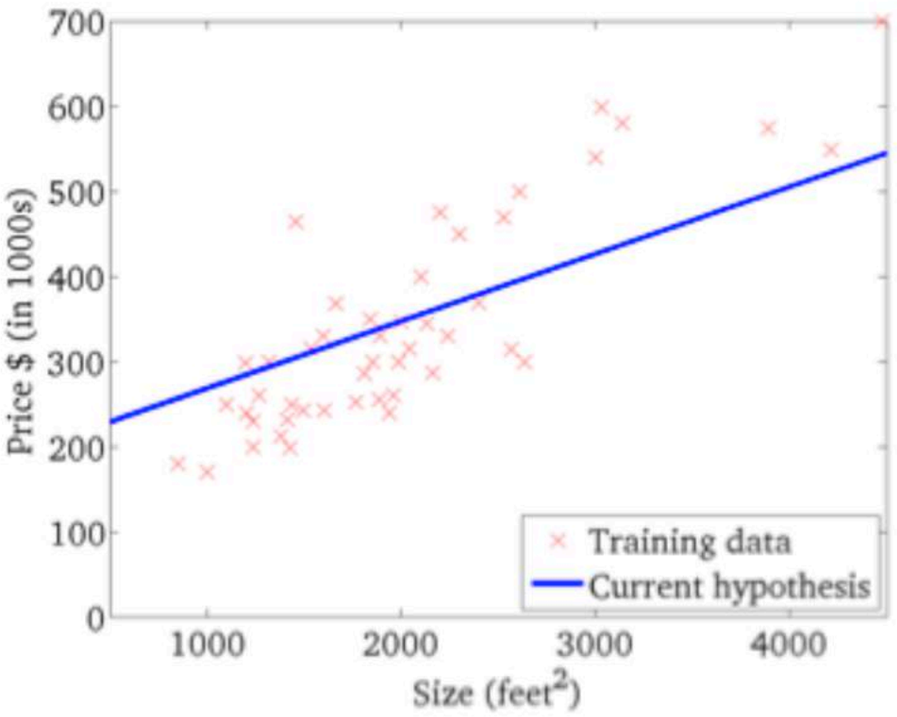
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



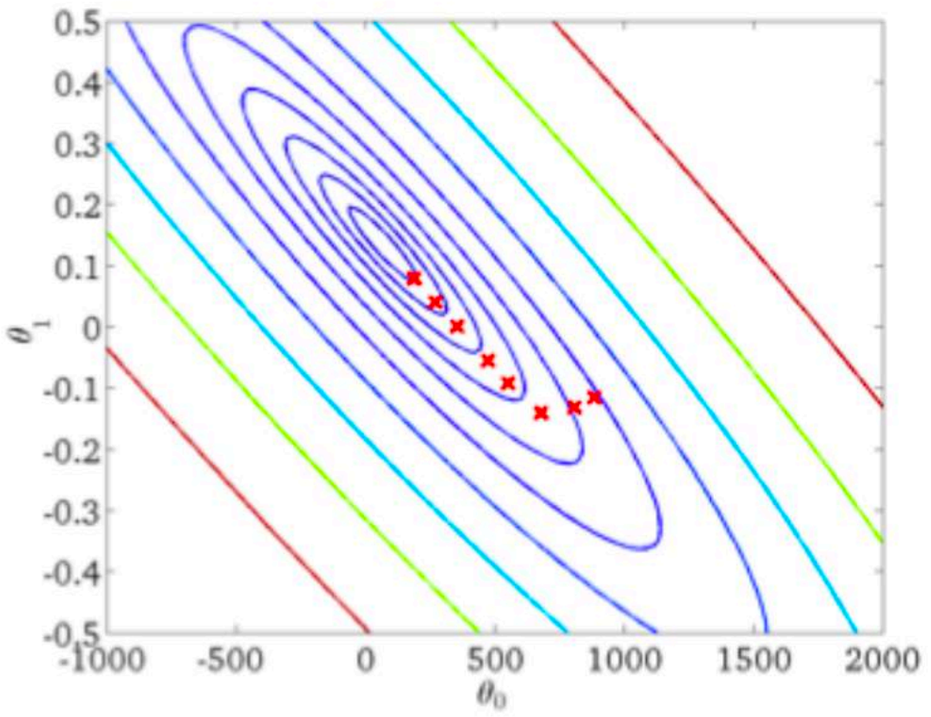
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



数学规划基础知识

Mathematical Optimization

Consider the general form of an optimization problem

$$\min_{x \in \mathbb{R}^n} \quad f(x) \quad \text{s.t. } x \in C$$

目标，约束，可行集合

In this lecture we consider the unconstrained minimization of a function f that satisfy the following requirements:

- f admits a minimizer x^* on $\mathbb{R}^n$.
- f is continuously differentiable and convex on $\mathbb{R}^n$.
- f is smooth in the sense that the gradient mapping ∇f is L-Lipschitz: for any $x, y \in \mathbb{R}^n$, one has $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$.
- C is convex

Read: Chapter 2, Numerical Optimization, Jorge Nocedal, Stephen Wright

重中之重： 泰勒定理

Theorem (Taylor's Theorem) 三种形式都很重要，1、3最常见。

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ *is continuously differentiable and that* $p \in \mathbb{R}^n$. *Then we have*

$$f(x+p) = f(x) + \nabla f(x+tp)^\top p$$

for some $t \in (0,1)$. *Moreover, if* f *is twice continuously differentiable, we have that*

$$\nabla f(x+p) = \nabla f(x) + \int_0^1 \nabla^2 f(x+tp)p \; dt$$

and that

$$f(x+p) = f(x) + \nabla f(x)^\top p + \frac{1}{2} p^\top \nabla^2 f(x+tp)p$$

for some $t \in (0,1)$.

Lipschitz Continuity

Definition (L-Lipschitz Continuity) 几何含义、Lipschtiz常数

In particular, a real-valued function $f : \mathbb{R}^n \to \mathbb{R}$ is called Lipschitz continuous (locally Lipschitz continuous on $\mathcal{B} \subset \mathbb{R}^n$) if there exists a positive real constant L such that, for all x_1 and $x_2 \in \mathbb{R}^n$ $(\in \mathcal{B})$,

$$|f(x_1) - f(x_2)| \leq L\|x_1 - x_2\|.$$

- (Lipschitz $\implies$ $\mathcal{C}^0$) Every locally Lipschitz-continuous function is continuous.
- ($\mathcal{C}^1$ $\implies$ Lipschitz) Every continuously differentiable function is locally Lipschitz.
- For $f \in \mathcal{C}^1$, Lipschitz continuity $\iff$ $\|\nabla f(x)\| \leq L$.

Examples:

L-smoothness

Definition (L-smoothness)

和L-Lipschtiz的关系

f is L-smooth (locally L-smooth on $\mathcal{B} \subset \mathbb{R}^n$) in the sense that the gradient mapping ∇f is L-Lipschitz: for any $x, y \in \mathbb{R}^n$ ($\in \mathcal{B}$), one has $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$.

Important property:

Lemma

Let f be a L-smooth function on $\mathbb{R}^n$. Then for any $x, y \in \mathbb{R}^n$, one has

$$|f(x) - f(y) - \nabla f(y)^\top (x - y)| \leq \frac{L}{2}\|x - y\|^2.$$

Proof.

Applying smoothness, and Cauchy-Schwarz, which improves the constant and also get rid of the convexity condition, to represent the quantity $f(x) - f(y)$ as an integral:

$$
\begin{aligned}
& |f(x) - f(y) - \nabla f(y)^\top (x - y)| \\
= & \left| \int_0^1 \nabla f(y + t(x - y))^\top (x - y)\, dt - \nabla f(y)^\top (x - y) \right| \\
= & \left| \int_0^1 [\nabla f(y + t(x - y))^\top - \nabla f(y)]^\top (x - y)\, dt \right| \\
\leq & \left| \int_0^1 L \| t(x - y) \| \| x - y \|\, dt \right| \\
\leq & \int_0^1 Lt \| x - y \|^2\, dt = \frac{L}{2} \| x - y \|^2 .
\end{aligned}
$$

□

- For $f \in \mathcal{C}^2$, Lipschitz continuity $\iff$ $\| \nabla^2 f(x) \| \leq L$.

Convexity “凸”为什么在优化分析里很重要

- Convex sets are central to optimization theory

- It is desirable to have a convex feasible region

- However, even if a feasible region is not convex, elements of convex analysis arise in our notions of optimality conditions

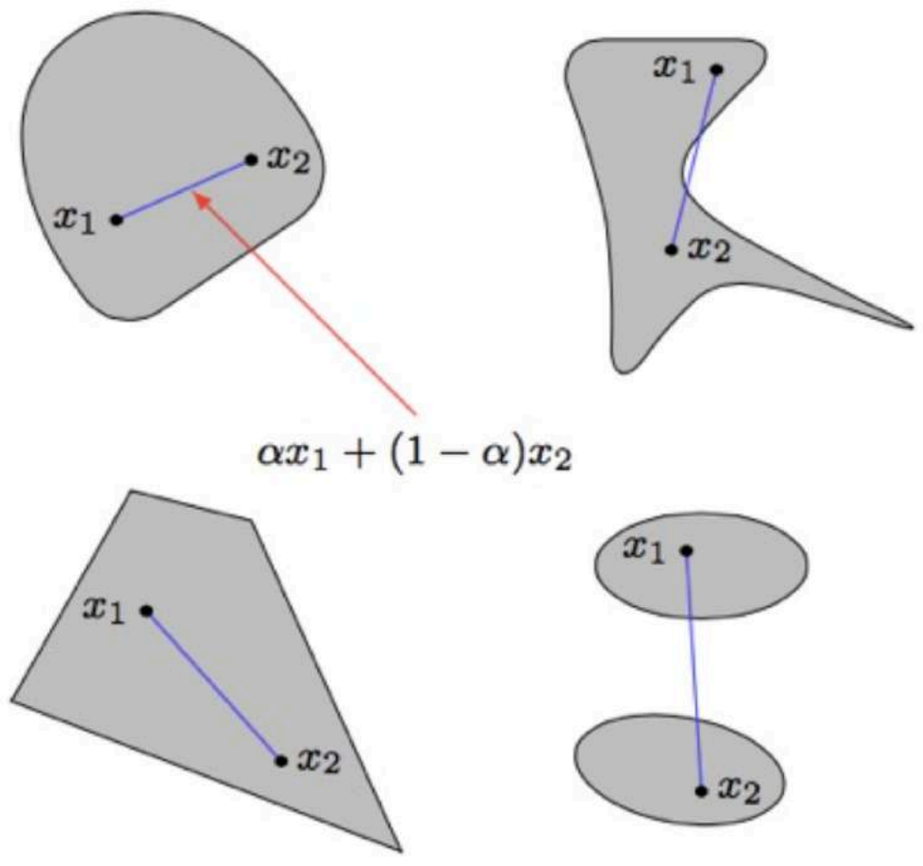
- In addition, numerous algorithms are based on topics that arise in convex analysis, including projections and separating hyperplanes

Convex Sets 凸集合的基本性质

Definition (Convex set)

A set $\mathcal{X}$ is convex if for any $\{x_1, x_2\} \in \mathcal{X}$ and $\alpha \in [0, 1]$

$$\alpha x_1 + (1 - \alpha) x_2 \in \mathcal{X}.$$



Sets that are convex (left) and not convex (right)

Convex/Concave functions 凸函数的性质、判定

Definition (Convex function)

A function $f: \mathbb{R}^n \to \mathbb{R}$ is convex if for all $\{x_1, x_2\} \in \mathcal{X}$ and $\alpha \in [0, 1]$, we have

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2).$$

- More generally, convexity of a function presumes convexity of its domain
- A function f is strictly convex if for $x_1 \neq x_2$ inequality above holds strictly
- Equivalent definition if f is differentiable and for $\forall \{x_1, x_2\} \in \mathcal{X}$

$$f(x_1) \geq f(x_2) + \nabla f(x_2)^\top (x_1 - x_2)$$

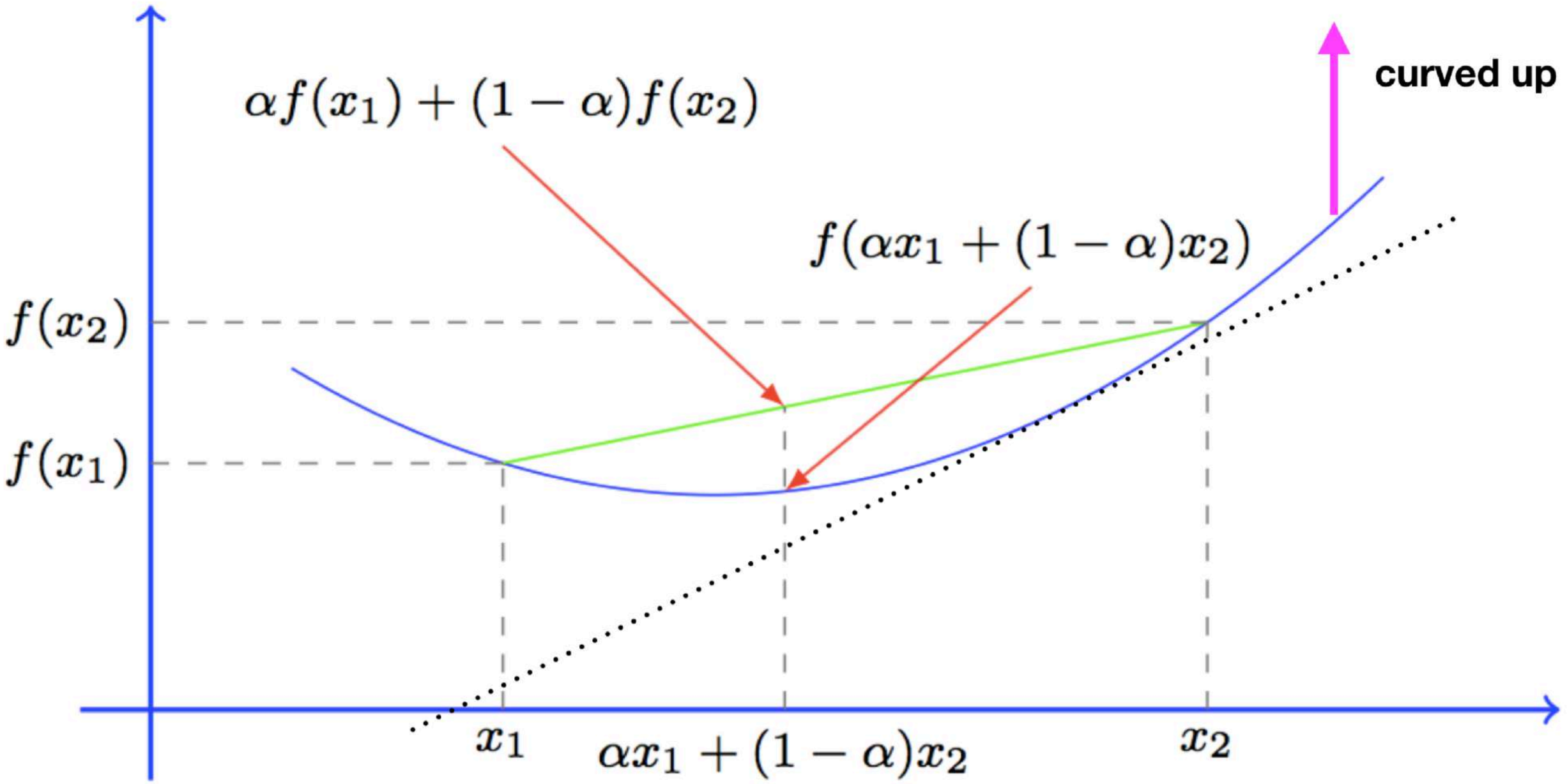
- Equivalent definition if f is twice differentiable and $\forall x \in \mathcal{X}$

$$\nabla^2 f(x) \succeq 0$$

- f is concave if $-f$ is convex

凸函数的几何意义

Convex/Concave functions



Strongly Convexity 强凸函数的性质、几何意义

Definition (Strongly Convex)

A function $f: \mathbb{R}^n \to \mathbb{R}$ is strongly convex with constant $\mu > 0$ if for all $\{x_1, x_2\} \in \mathcal{X}$ and $\alpha \in [0, 1]$, we have

$$f(\alpha x_1 + (1-\alpha)x_2) + \tfrac{1}{2}\mu\alpha(1-\alpha)\|x_1 - x_2\|_2^2 \leq \alpha f(x_1) + (1-\alpha)f(x_2).$$

- Equivalent definition if f is differentiable and for $\{x_1, x_2\} \in \mathcal{X}$, it holds

$$f(x_1) \geq f(x_2) + \nabla f(x_2)^\top (x_1 - x_2) + \tfrac{1}{2}\mu\|x_1 - x_2\|_2^2.$$

- Equivalent definition if f is differentiable and for $\{x_1, x_2\} \in \mathcal{X}$, it holds

$$(\nabla f(x_1) - \nabla f(x_2))^\top (x_1 - x_2) \geq \mu\|x_1 - x_2\|_2^2.$$

- Equivalent definition if f is twice differentiable and $\forall x \in \mathcal{X}$, it holds

$$\nabla^2 f(x) - \mu I \succeq 0.$$

Convex and L-Smooth f

From previous page, for convex L-smooth f, we have

$$0 \leq f(x) - f(y) - \nabla f(y)^\top (x - y) \leq \frac{L}{2}\|x - y\|^2.$$

Lemma

Let f be a convex and L-smooth function on $\mathbb{R}^n$. Then for any $x, y \in \mathbb{R}^n$, one has

$$f(x) - f(y) - \nabla f(x)^\top (x - y) \leq -\frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2.$$

Lemma

Let f be a convex and L-smooth function on $\mathbb{R}^n$. Then for any $x, y \in \mathbb{R}^n$, one has

$$[\nabla f(x) - \nabla f(x)]^\top (x - y) \geq \frac{1}{L}\|\nabla f(x) - \nabla f(y)\|^2.$$

Proof. 推导建议仔细阅读掌握凸函数、L-smooth函数的基本性质

Let $\phi(y) = f(y) - \nabla f(x)^\top y$. Note that ϕ is convex and L-smooth. Remark also that x is the minimizer of ϕ since $\phi(y) - \phi(x) \geq \phi(x)^\top (y - x) = 0$. Thus:

L-smooth

x is optimal

$$f(x) - f(y) - \nabla f(x)^\top (x - y) = \phi(x) - \phi(y) \leq \phi\left(y - \frac{1}{L}\nabla\phi(y)\right) - \phi(y)$$

$$\leq \nabla\phi(y)^\top \left(y - \frac{1}{L}\nabla\phi(y) - y\right) + \frac{L}{2}\left\|y - \frac{1}{L}\nabla\phi(y) - y\right\|^2$$

$$= -\frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2,$$

which concludes the proof. □

Proof.

Summing up two inequalities

$$f(x) - f(y) - \nabla f(x)^\top (x - y) \leq -\frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2$$

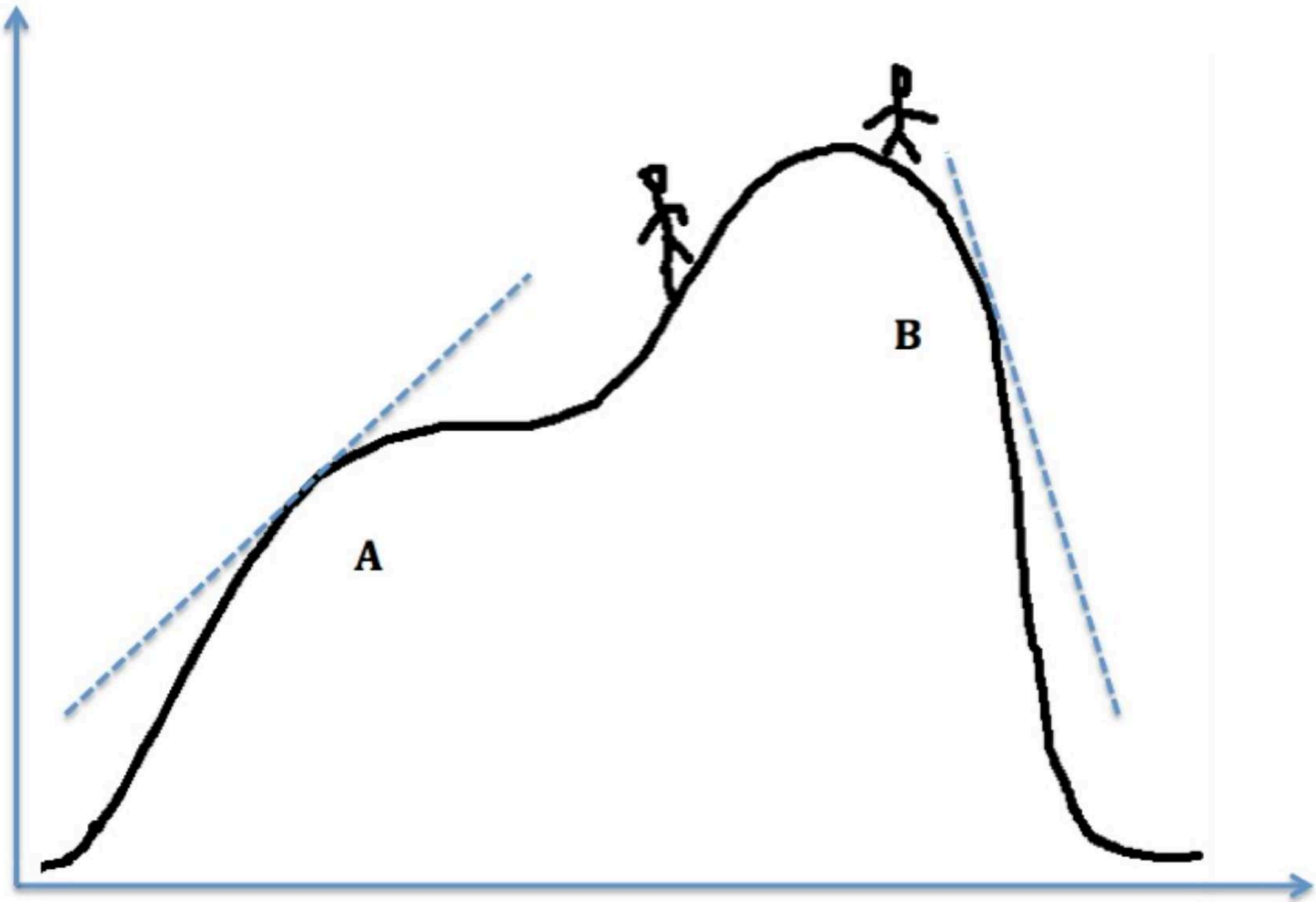
$$f(y) - f(x) - \nabla f(y)^\top (y - x) \leq -\frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2.$$

we have

$$[\nabla f(x) - \nabla f(x)]^\top (x - y) \geq \frac{1}{L}\|\nabla f(x) - \nabla f(y)\|^2.$$

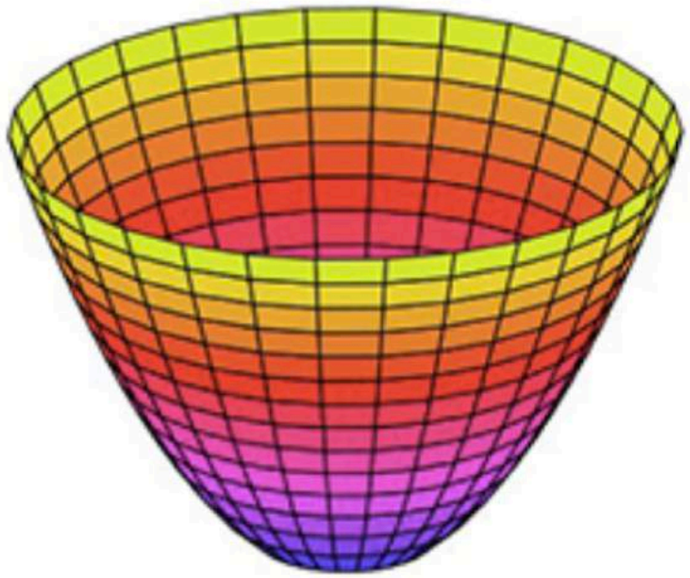
Hiking down a hill

导数所反应的信息是什么?



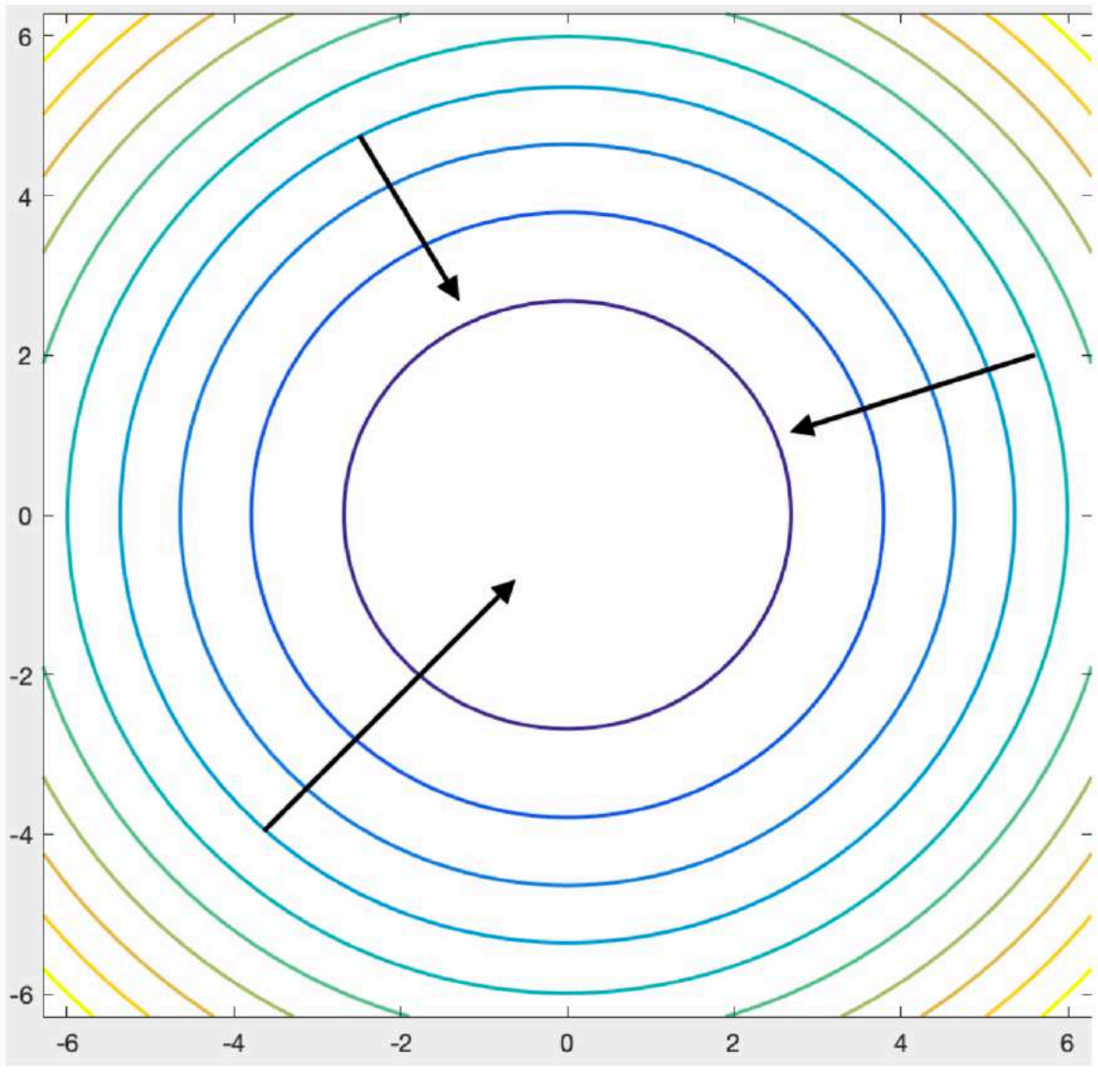
Slope of the tangent: $f'(A) = 1$, $f'(B) = -3$.

Gradient and contour 多元变量函数的梯度、等值线（等高线）



$$f(x_1, x_2) = 0.5x_1^2 + 0.5x_2^2$$

$$\nabla f = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



Directional derivatives 方向导数的定义和几何意义

Definition (Directional derivative)

Given proper $f\colon \mathbb{R}^n \to \bar{\mathbb{R}}$, a point $x \in \operatorname{dom}(f)$, and a direction $d \in \mathbb{R}^n$, the directional derivative of f at x in the direction d (if it exists) is

$$f'(d;x) = \lim_{\alpha \to 0^+} \frac{f(x+\alpha d) - f(x)}{\alpha}.$$

- Note that this definition does not require f to be differentiable
- If f is convex, then for every $x \in \operatorname{dom}(f)$ and $d \in \mathbb{R}^n$, the limit exists
- If f is convex and $x \in \operatorname{int}(\operatorname{dom}(f))$, then $f'(d;x)$ is finite
- If f is differentiable, then $f'(d;x)$ exists and

$$f'(d;x) = \nabla f(x)^\top d.$$

Descent directions

重中之重：下降方向

Consider a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$

- At a point $x \in \mathbb{R}^n$, a descent direction d is one for which we have

$$\boxed{\nabla f(x)^\top d < 0}$$

- We can decrease f by moving (a small distance) along such a direction d. Why?

为什么下降方向可以
带来目标下降？

The steepest descent direction: the negative gradient

最速下降方向的含义

- The steepest descent direction is the solution of the optimization problem

$$\min_{d\in\mathbb{R}^n} f'(d;x) \quad \text{s.t. } \|d\|_2 \le 1$$

- Solve for the optimal solution

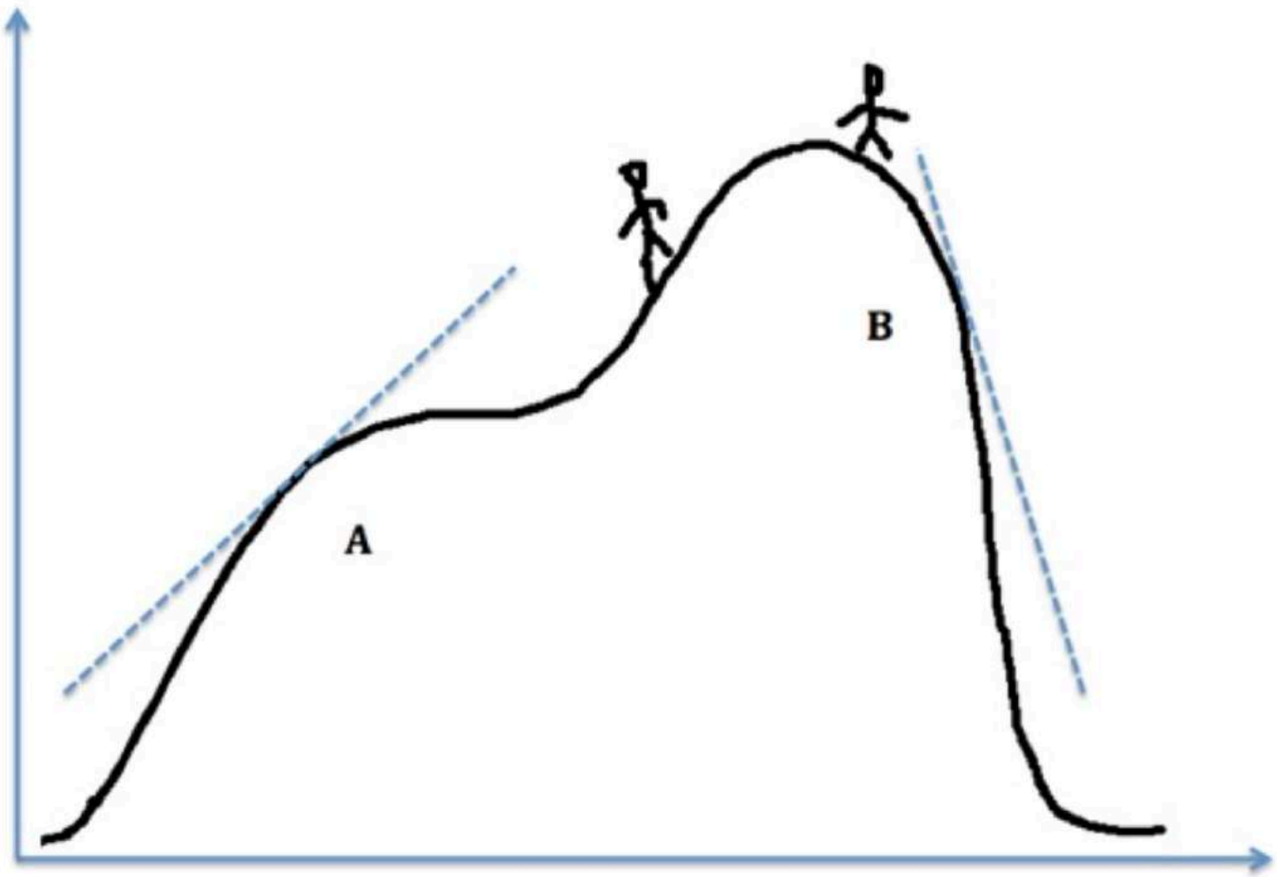
$$\min_{\|d\|_2\le 1} f'(d;x) = \min_{\|d\|_2\le 1} g^\top d$$

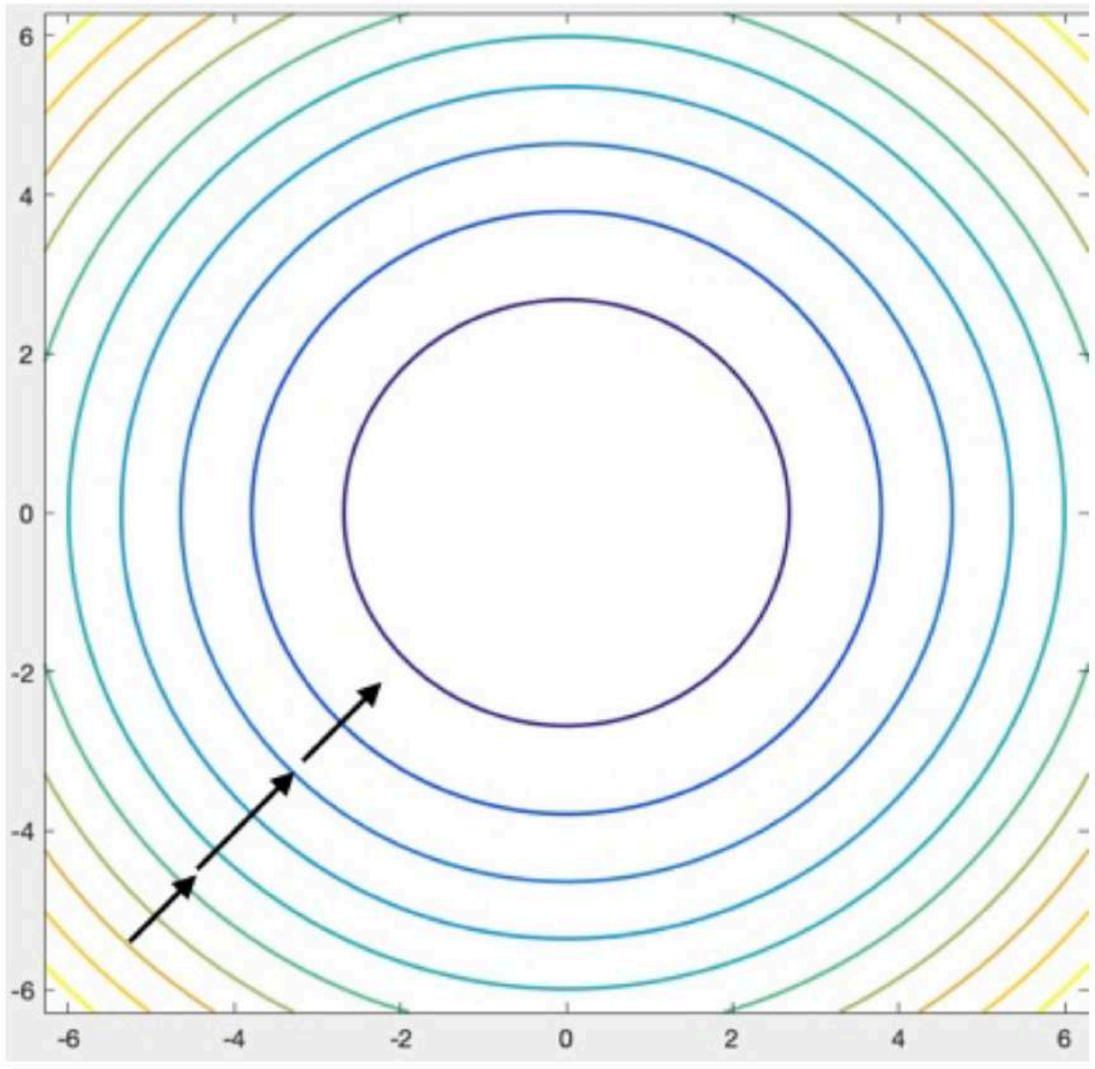
- The steepest descent direction

$$d = -g/\|g\|_2 \quad (g \neq 0)$$

Moving along a descent direction

何时最优?





Question is when to stop? or, how do you know you are (nearly) at an optimal solution.

Global and local minima 最优解的定义：全局极小和局部极小

Ideal minima are those that minimize a function globally over its domain

Definition (Global minimum)

A vector x^* is a (strictly) global minimum (minimizer) of f is

$$f(x^*) \leq f(x), \forall x \in \mathbb{R}^n \quad (f(x^*) < f(x), \forall x \in \mathbb{R}^n)$$

Commonly, however, we are satisfied with a weaker form of minimizer.

Definition (Local minimum)

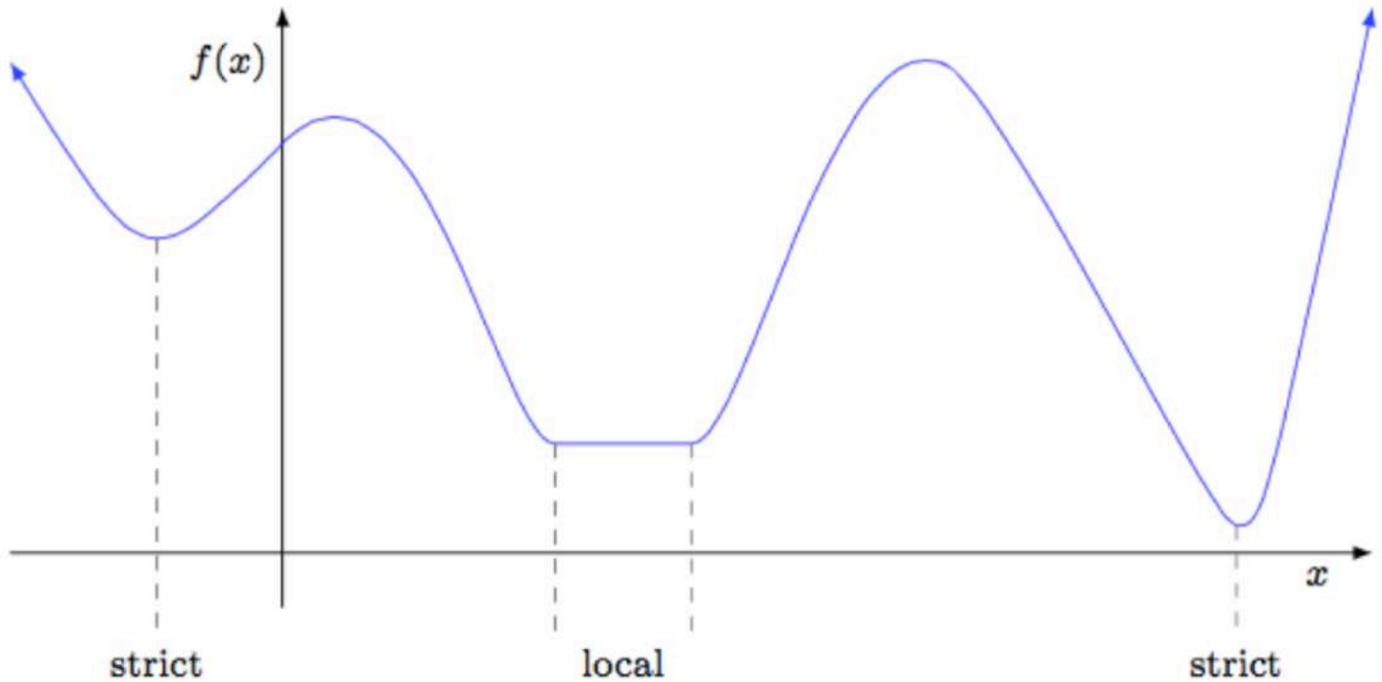
A vector x^* is a (strictly) local minimum (minimizer) of f if there exists $\epsilon > 0$ such that

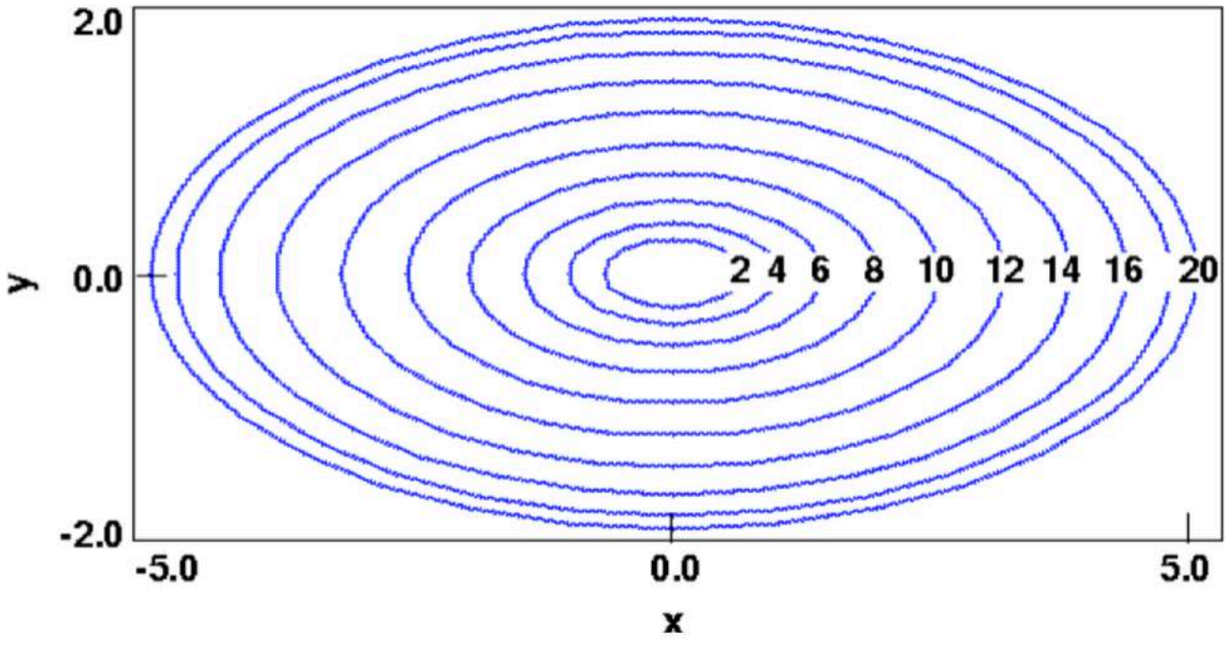
$$f(x^*) \leq f(x), \forall x \in \mathbb{B}(x^*, \epsilon) = \{x \in \mathbb{R}^n : \|x - x^*\|_2 \leq \epsilon\}$$

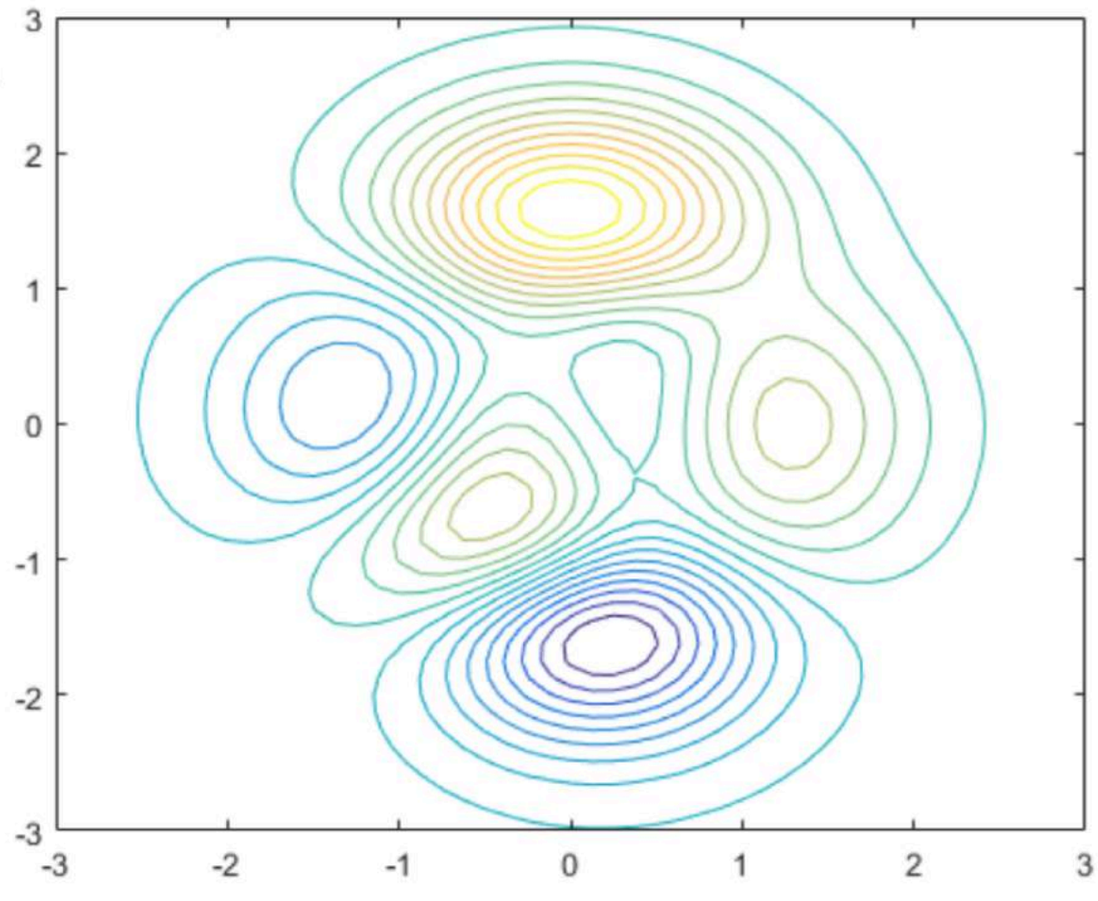
$$(f(x^*) < f(x), \forall x \in \mathbb{B}(x^*, \epsilon) = \{x \in \mathbb{R}^n : \|x - x^*\|_2 \leq \epsilon\})$$

Global and local minima

全局和局部极小的几何含义
以及各自的难点







Global and local minima 重中之重：凸则局部极小=全局极小

$$\text{Local minimum} \overset{\text{convex}}{\Longrightarrow} \text{Global minimum}$$

A special fact in convex optimization is that all local minima are global minima

Theorem 很重要的定理

If $f : \mathbb{R}^n \to \mathbb{R}$ is convex, then a local minimum of f is a global minimum of f. If f is strictly convex, then there exists at most one global minimum of f.

Proof.

To derive a contradiction, suppose that x^* is a local minimum of f that is not a global minimum. Then, there exists $\bar{x}$ such that $f(\bar{x}) < f(x^*)$. By convexity of f , we have for all $\alpha \in (0, 1)$ that

$$f(\alpha x^* + (1 - \alpha)\bar{x}) \leq \alpha f(x^*) + (1 - \alpha) f(\bar{x}) < f(x^*).$$

This means that f has a value strictly lower that $f(x^*)$ at every point on the line segment $(x^*, \bar{x}]$, which violates the local minimality of x^*.

□

Global and local minima 非凸优化问题怎么办?

- Unfortunately, for nonconvex optimization, the conditions in the definitions of global and local minima are not entirely useful.

- Unless we can verify strict quasiconvexity, we rarely have global information about f, and so have no way to verify if a point is a global minimizer.
  即使对于获取局部解，也不容易

- Thus, in nonconvex optimization, we often focus on finding a local minimizer.

- Using calculus, we can derive local optimality conditions that aid in determining if a point is a local minimizer.

- In this manner, we rarely (if ever) use the aforementioned definitions directly.

First-order necessary condition 如何去刻画局部极小点

Theorem (First-order necessary condition)

If $f \in \mathcal{C}$ and x^* is a local minimizer of f, then $\nabla f(x^*) = 0$.

Proof.

Hint: think about the descent directions··· □

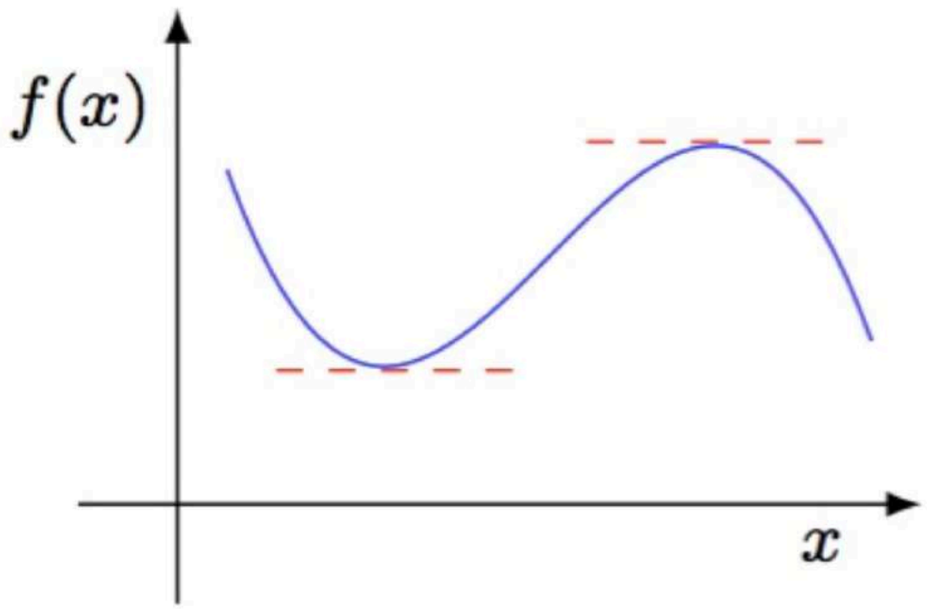
重要定理：但我认为你可以独立证明出来。

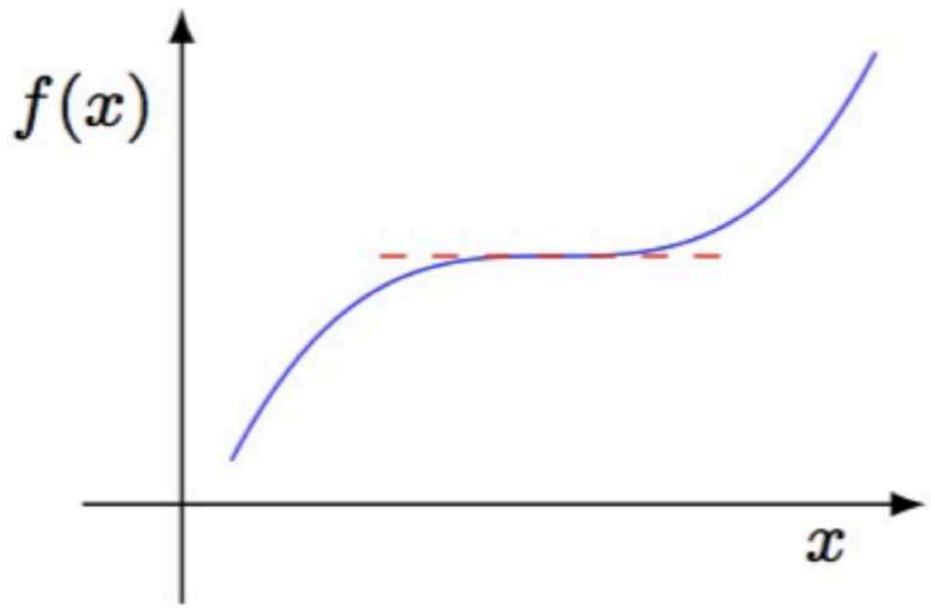
Stationary points (first-order optimal solutions)

一阶条件和最优性的联系。驻点定义。

- We can limit our search to points where $\nabla f(x^*) = 0$.

- However, $\nabla f(x^*) = 0$ does not imply that we have a local minimizer!





- At least we know that if $\nabla f(x^*) \neq 0$, then x^* is not a local minimizer.

Definition (Stationary point)

A point $x \in \mathbb{R}^n$ is a stationary point for $f \in \mathcal{C}$ if $\nabla f(x) = 0$.

Why convexity is pretty? 凸问题为何简单?

If f is convex, then we have the following stronger result.

Theorem (First-order necessary and sufficient condition)

If $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is convex and $\nabla f(x^*) = 0$, then x^* is a global minimizer of f.

The following statements are true about the convex minimization problem.

- If a local minimum exists, then it is a global minimum.
- The set of all (global) minima is convex.
- For each strictly convex function, if the function has minimum, then the minimum is unique.

Nonsmooth convex cases: subgradient

凸、但非光滑的情况　　次梯度、次微分不是一回事！

A subgradient c of a convex function f at a point x^0 is a vector such that

$$f(x) \geq f(x^0) + c^\top (x - x^0).$$

The set of all subgradient c is called the subdifferential of f at x^0, denoted as $\partial f(x^0)$, and it is convex.

Theorem (First-order condition)

If $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is convex and $0 \in \partial f(x^*)$, then x^* is a global minimizer of f.

Termination 终止条件是要有最优性条件来保证有意义的！

When to terminate?

By definition: $\nabla f(\theta^*) = 0 \iff \theta^*$ is stationary

By convexity: $\nabla f(\theta^*) = 0 \iff \theta^*$ is global minimizer

Termination criterion: $\|\nabla f(\theta^k)\|_2^2 \leq \epsilon$

More practically, $\|\nabla f(\theta^k)\|_{\max} \leq \epsilon$

How to select learning rate? This relates to the convergence analysis of BGD...

Convergent sequences 何为收敛?

Recall the following definition

Definition (Convergent sequence)

A sequence $\{x^k\}$ is said to converge to x^* if

$$\lim_{k\to\infty} \|x^k - x^*\|_2 = 0$$

Theorem (Bolzano-Weierstrauss Theorem)

Every bounded sequence in $\mathbb{R}^n$ has a convergent subsequence.

Global convergence of an algorithm （全局收敛性）

全局收敛≠收敛到全局解

Definition (Globally convergent algorithm)

If, from any initial iterate x^0 (and under presumed conditions), the iterate sequence $\{x^k\}$ generated by an algorithm for solving a problem converges to a solution of the problem, then the algorithm is said to be globally convergent (under the presumed conditions).

Global convergence of an algorithm

全局收敛≠收敛到全局解

- Note that, in the context of optimization, this definition does NOT presume or require convergence to a globally optimal solution. (Even optimization experts are often confused by this!)

- This definition is not saying that a given iterate sequence is "globally" convergent. After all, if such an algorithm produces a sequence $\{x^k\}$, the sequence itself is simply convergent.

- Rather, this definition refers to a property of an algorithm and the fact that it produces a convergent sequence from any starting point.

- We may also say that an algorithm is globally convergent if, e.g., a stationarity measure vanishes, even if the iterate sequence itself does not converge; e.g., if $\nabla f(x^k) \to 0$.

Local convergence of an algorithm （局部收敛性）

局部收敛和全局收敛的区别

This refers to the local behavior of an algorithm. By saying local, we mean "near" the limit point.

Definition (Locally convergent algorithm)

If, from any initial x^0 in a neighborhood of a solution x^* (and under presumed conditions), the iterate sequence $\{x^k\}$ generated by an algorithm converges to x^*, then the algorithm is said to be locally convergent to x^* (under the presumed conditions)

Local convergence of an algorithm

局部收敛和全局收敛的区别

- A locally convergent algorithm is NOT necessarily globally convergent. e.g. Newton method (we will see later in this class).

- A globally convergent algorithm is locally convergent.

- However, an important characterization of local convergence is the corresponding rate of convergence. This distinguishes "how quickly" the iterates converge; e.g., 0.9999999999999^k converges to 0, but I don't want to have to wait for it to get very close to 0...

- (One can also talk about rates of global convergence, as is commonly done in convex optimization. Illustrate later....)

Convergence Rate

如何刻画算法的效率？

Suppose $\|x_k - x_*\|_2 \to 0$; i.e., the sequence $\{x_k\}$ converges to x_*.

Definition (Q-linear convergence)

If there exists a constant $c \in [0, 1)$ and $\hat{k} \geq 0$ such that

$$\|x_{k+1} - x_*\|_2 \leq c\|x_k - x_*\|_2 \text{ for all } k \geq \hat{k},$$

then $\{x_k\}$ converges Q-linearly to x_*.

Definition (Q-superlinear convergence)

If there exists a sequence $\{c_k\} \to 0$ such that

$$\|x_{k+1} - x_*\|_2 \leq c_k\|x_k - x_*\|_2,$$

then $\{x_k\}$ converges Q-superlinearly to x_*.

Definition (Q-quadratic convergence)

If there exists a constant $c \geq 0$ and $\hat{k} \geq 0$ such that

$$\|x_{k+1} - x_*\|_2 \leq c\|x_k - x_*\|_2^2 \text{ for all } k \geq \hat{k},$$

then $\{x_k\}$ converges Q-quadratically to x_*.

There is also "R-" convergence, but we'll skip that; so we'll drop the "Q-".

Convergence Rate

```
>> x = newton('example2',1.391);
================================
  k      ||F(x)||        ||d||
================================
   0   9.4749e-001   2.7808e+000
   1   9.4708e-001   2.7763e+000
   2   9.4598e-001   2.7647e+000
   3   9.4308e-001   2.7342e+000
   4   9.3539e-001   2.6555e+000
   5   9.1489e-001   2.4597e+000
   6   8.5946e-001   2.0165e+000
   7   7.0810e-001   1.2272e+000
   8   3.5527e-001   4.0417e-001
   9   3.3148e-002   3.3184e-002
  10   2.4303e-005   2.4303e-005
  11   9.5692e-015   -----------
================================
>> x

x =

 -9.5692e-015
```

```
>> x = newton('example',[-1;1]);
================================
  k      ||F(x)||        ||d||
================================
   0   6.3212e-001   6.5353e-001
   1   4.6100e-002   4.1159e-002
   2   2.4495e-004   2.2103e-004
   3   6.9278e-009   -----------
================================
>> x

x =

   -0.5671
    0.5671
```

Oracle complexity 如何刻画算法的效率?

Definition (ϵ-optimality)

For $\epsilon > 0$, x is said to be ϵ-optimality if

$$f(x) - f(x^*) < \epsilon.$$

Alternatively, one can define it by gradient such as $\|f(x)\| < \epsilon$.

Worst-case complexity studies how many iterations at most for an algorithm to reach ϵ-optimality: $O(1/\epsilon)$, equivalent to saying

$$f(x) - f(x^*) = O(1/k).$$

Note: criterion may vary according to different optimality residual

How do we evaluate the efficiency of an optimization algorithm?

"efficiency"是个很模糊的定义。

算法的有效性可以有多方面标准。

$$\min_{x \in C} \quad f(x)$$

- (Global) convergence
  - global algorithm
  - local algorithm
- Convergence speed
  - (local) convergence rate
  - oracle complexity analysis
- Computational cost per iteration
  - function/gradient evaluations
  - subproblem solution
- Storage
- Many other aspects... (parallel, distributed, ...)

Convergence Analysis of Batch Gradient Descent

Interpretation of Gradient Descent

（最速）梯度下降法的直观含义是什么?

General nonlinear problem (may be nonconvex)

$$\min_{x \in C} \quad f(x)$$

Gradient descent:

$$x^{k+1} \leftarrow x^k - \alpha \nabla f(x^k)$$

At each iteration, consider the expansion

$$f(y) \approx f(x) + \nabla f(x)^\top (y - x) + \frac{1}{2\alpha} \|y - x\|^2$$

Quadratic approximation, replacing usual $\nabla^2 f(x)$ by $\frac{1}{\alpha} I$

$f(x) + \nabla f(x)^\top (y - x)$ linear approximation to f

$\frac{1}{2t} \|y - x\|^2$ proximity term to x, with weight $1/(2\alpha)$

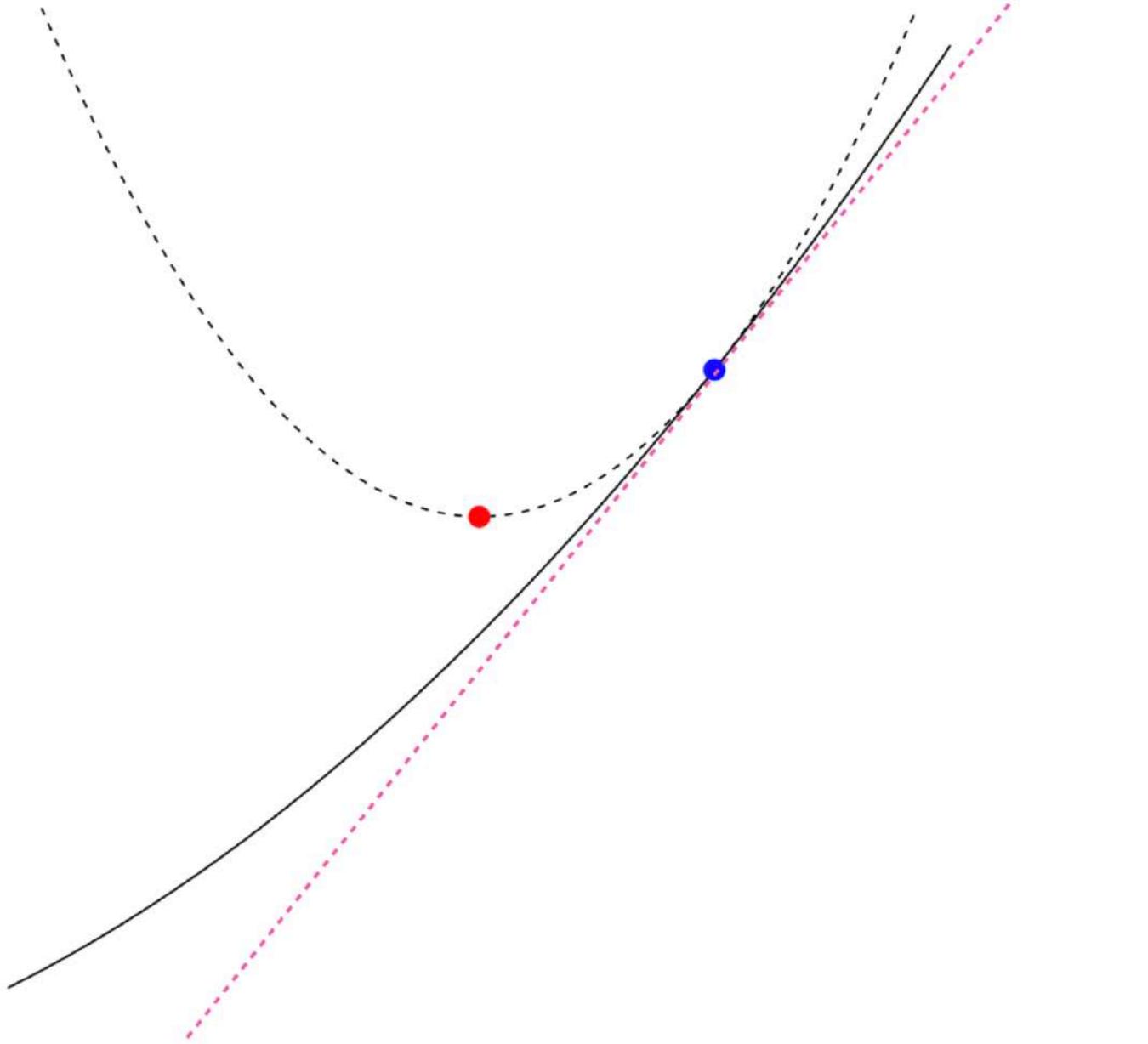
Choose next point $y = x^+$ to minimize quadratic approximation

$$x^+ = x - \alpha \nabla f(x)$$

Gradient descent revisited Geoff Gordon & Ryan Tibshirani Optimization 10-725 / 36-725

Interpretation of Gradient Descent

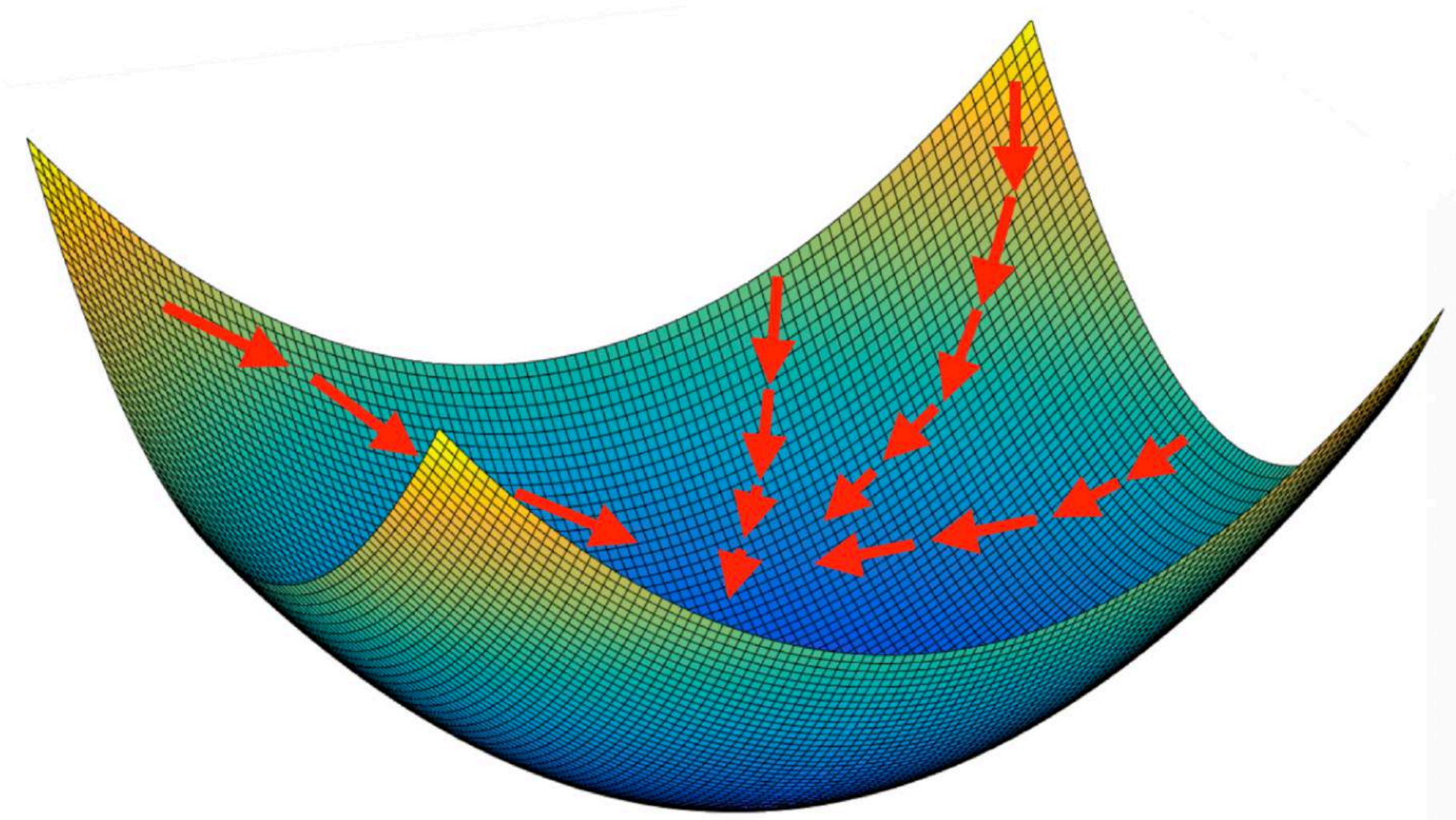
（最速）梯度下降法的几何含义是什么？



Blue point is x, red point is x^+

Starting point (global convergence)

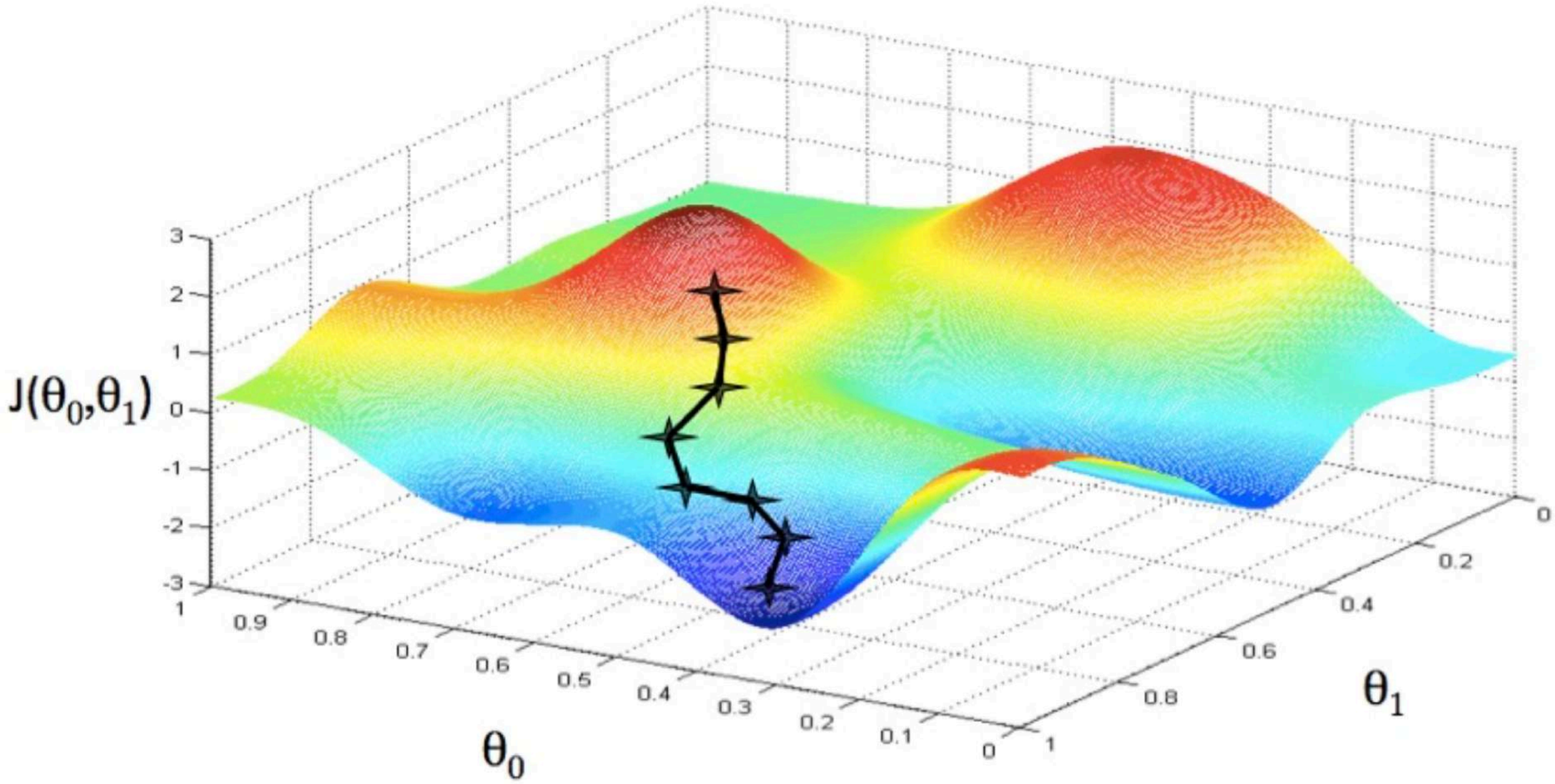
Arbitrary starting point will lead to optimal solution, since least squared error is convex, since any local solution is global



Starting point (global convergence)

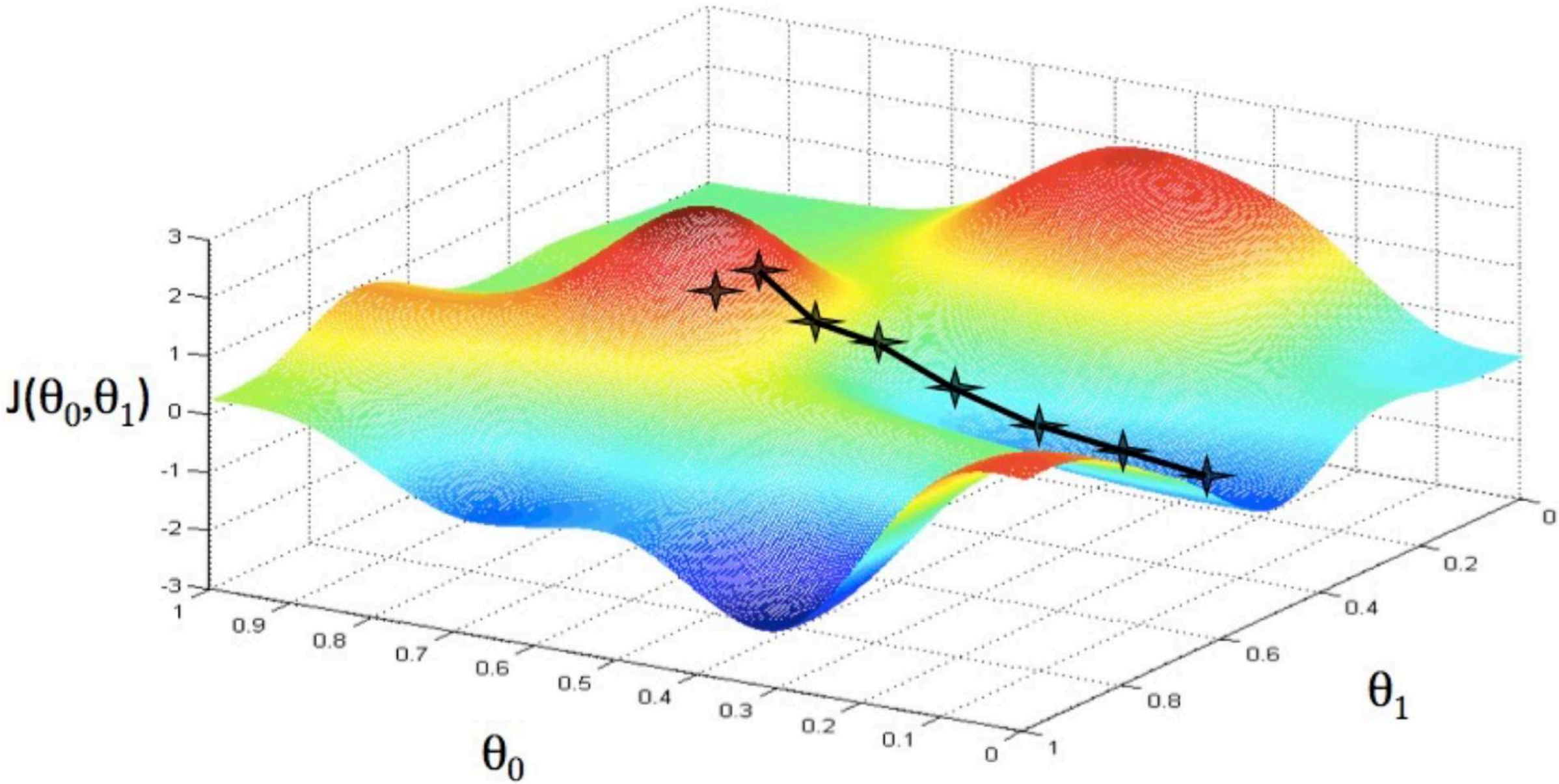
算法虽然全局收敛，但最终解可能不同。

For nonconvex cases, the algorithm converges starting with any initial point. But may lead to different optimal solution.



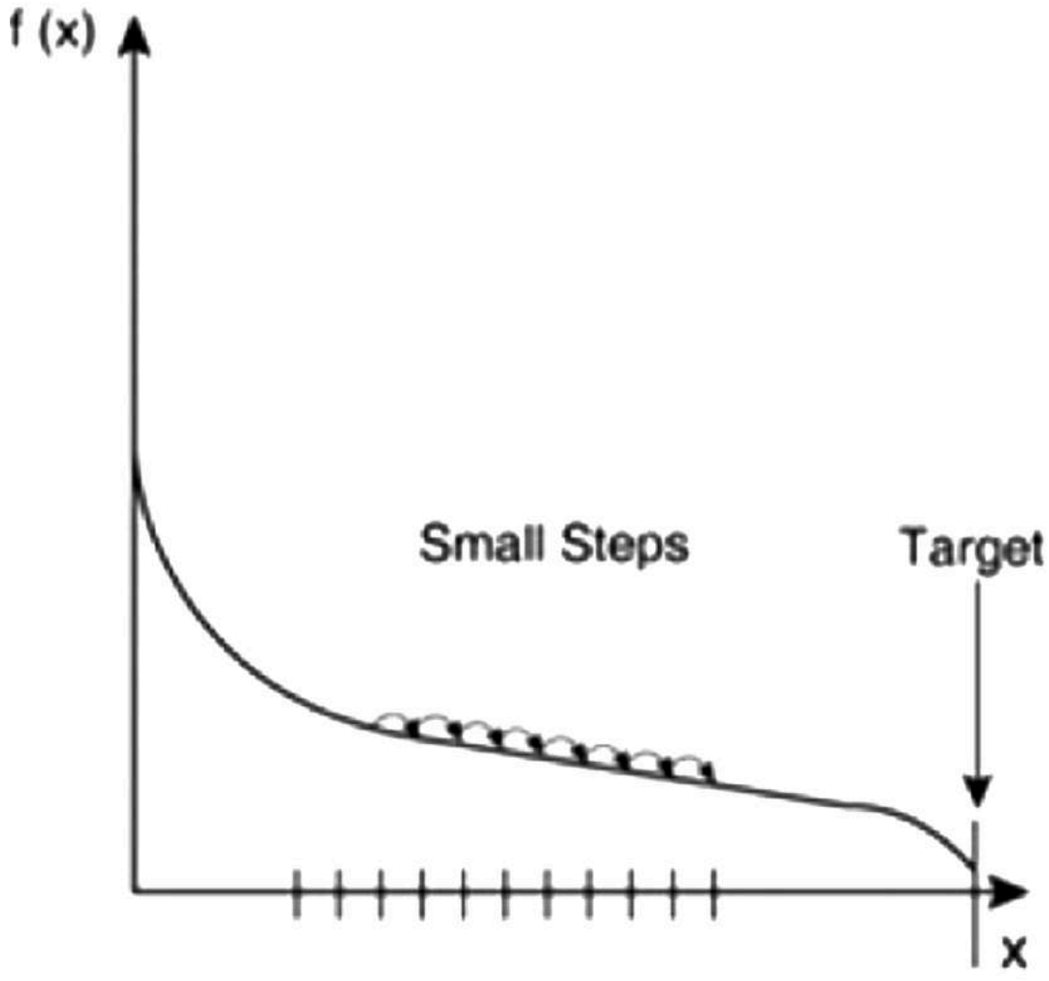
Starting point (global convergence)

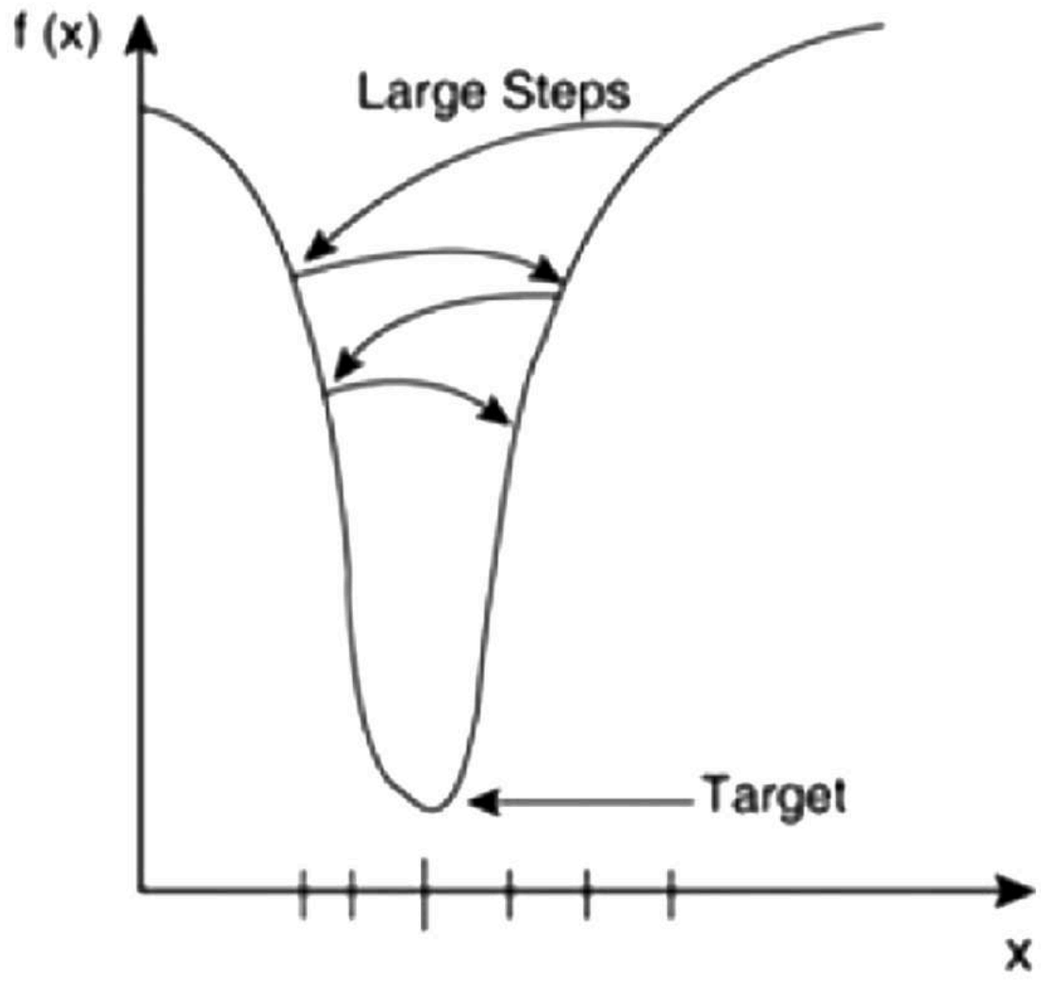
For nonconvex cases, the algorithm converges starting with any initial point. But may lead to different optimal solution.



Stepsize/Learning Rate 算法的效率取决于学习率

Too small stepsizes and too large stepsizes





Stepsize/Learning Rate

学习率的选取是batch情形下的核心

Fixed stepsize α

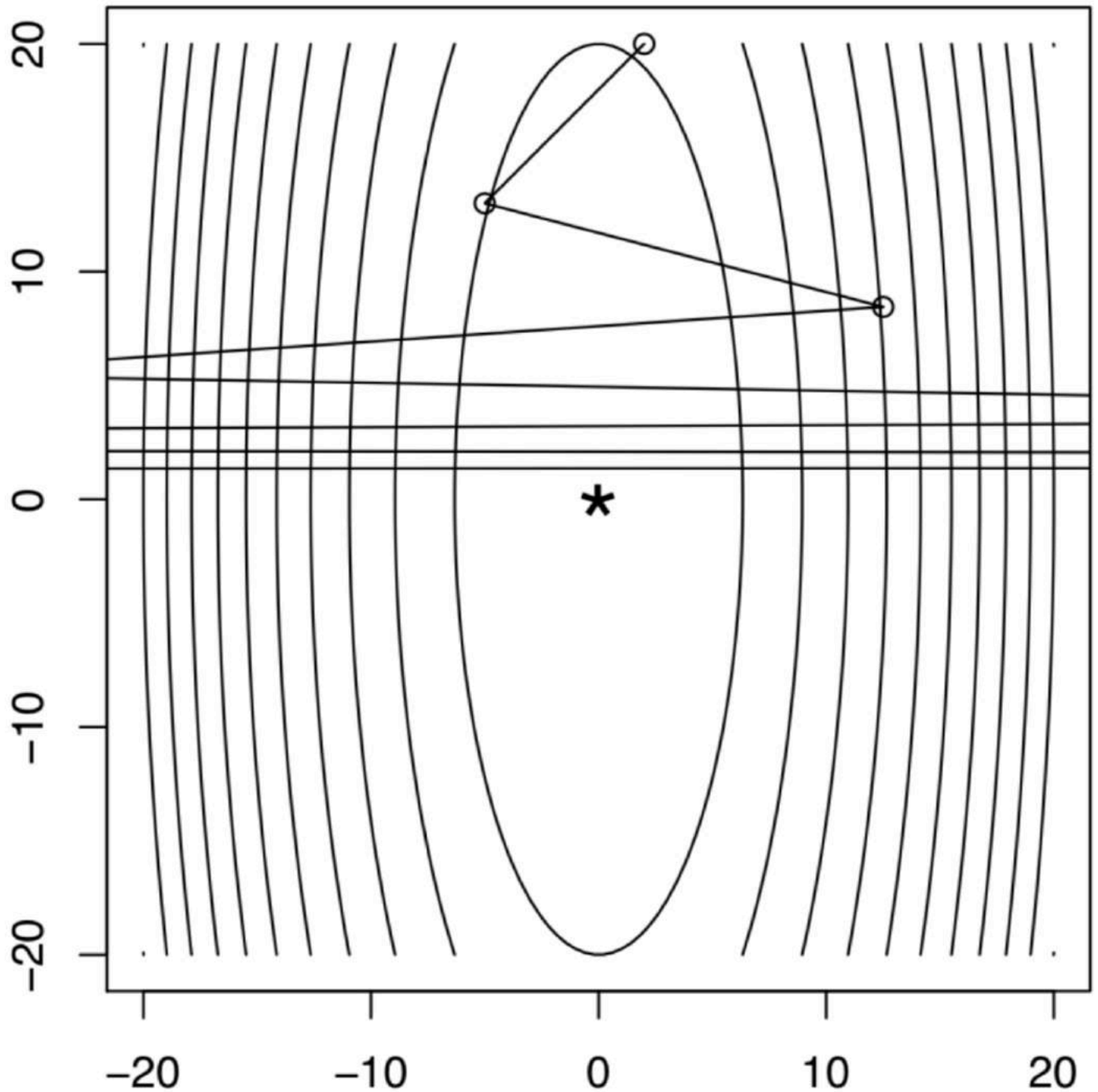
- Too small, slow progress, slow convergence
- Too large, oscillation, slow convergence, or even diverge

Dynamically changing stepsize α^k

- Too small, slow progress, slow convergence, or fail to converge to minimizers
- Too large, oscillation, slow convergence, or even diverge

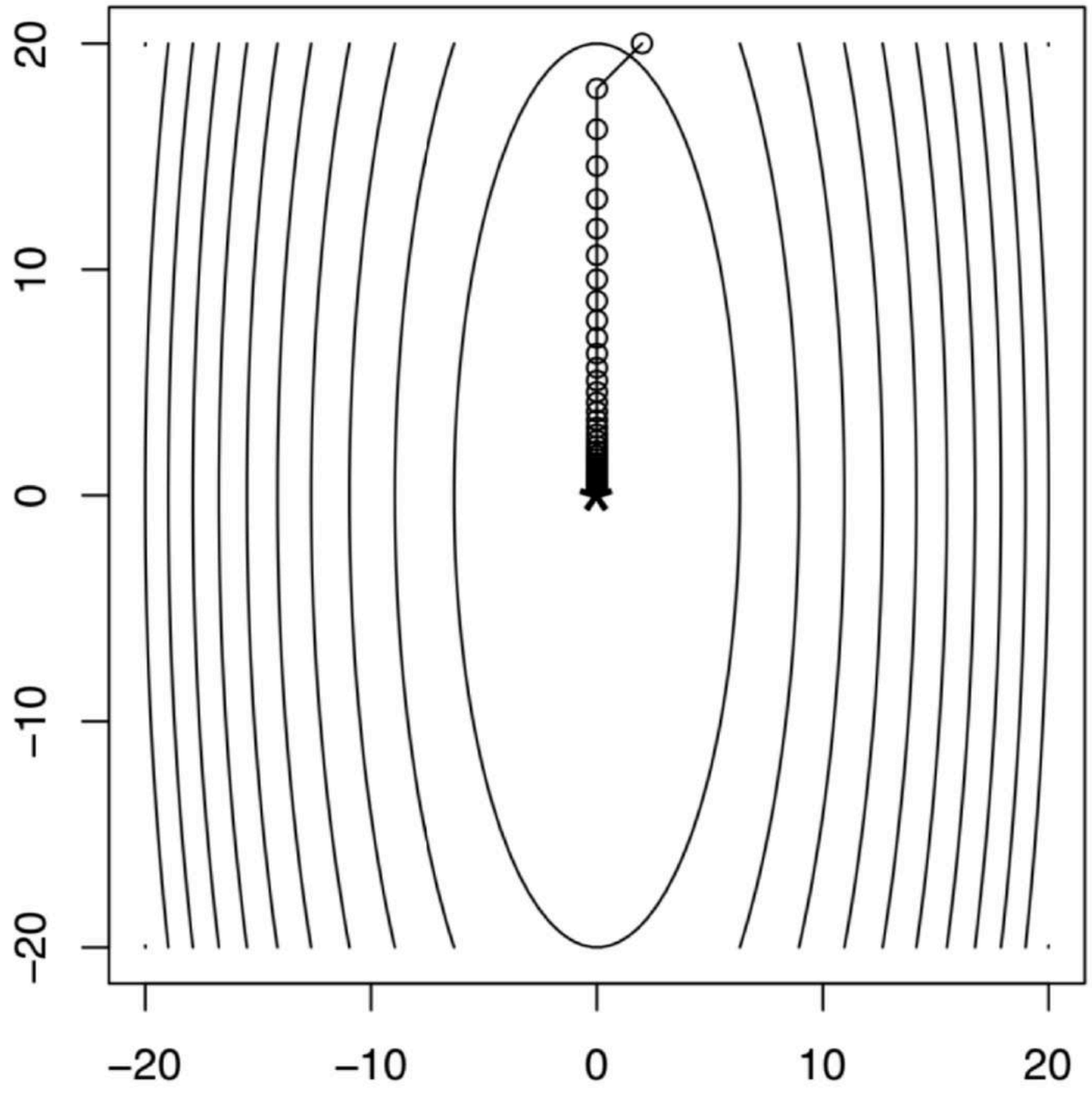
Fixed step size

Simply take $\alpha^k = \alpha$ for all $k = 1, 2, 3, ...$, can diverge if t is too big.
Consider $f(x) = (10x_1^2 + x_2^2)/2$, gradient descent after 8 steps:



Fixed step size

Same example, gradient descent after 40 appropriately sized steps:



Backtracking line search 传统技能里如何选学习率（步长）？

A way to adaptively choose the step size

- First fix a parameter $0 < \gamma < 1$
- Then at each iteration, start with $\alpha = 1$, and while

$$f(x - \alpha \nabla f(x)) > f(x) - \frac{\alpha}{2} \|\nabla f(x)\|_2^2,$$

  update $\alpha = \gamma \alpha$

Practical line search may even relax this condition as

$$f(x - \alpha \nabla f(x)) > f(x) - \frac{\alpha}{2} c_{ls} \|\nabla f(x)\|_2^2,$$

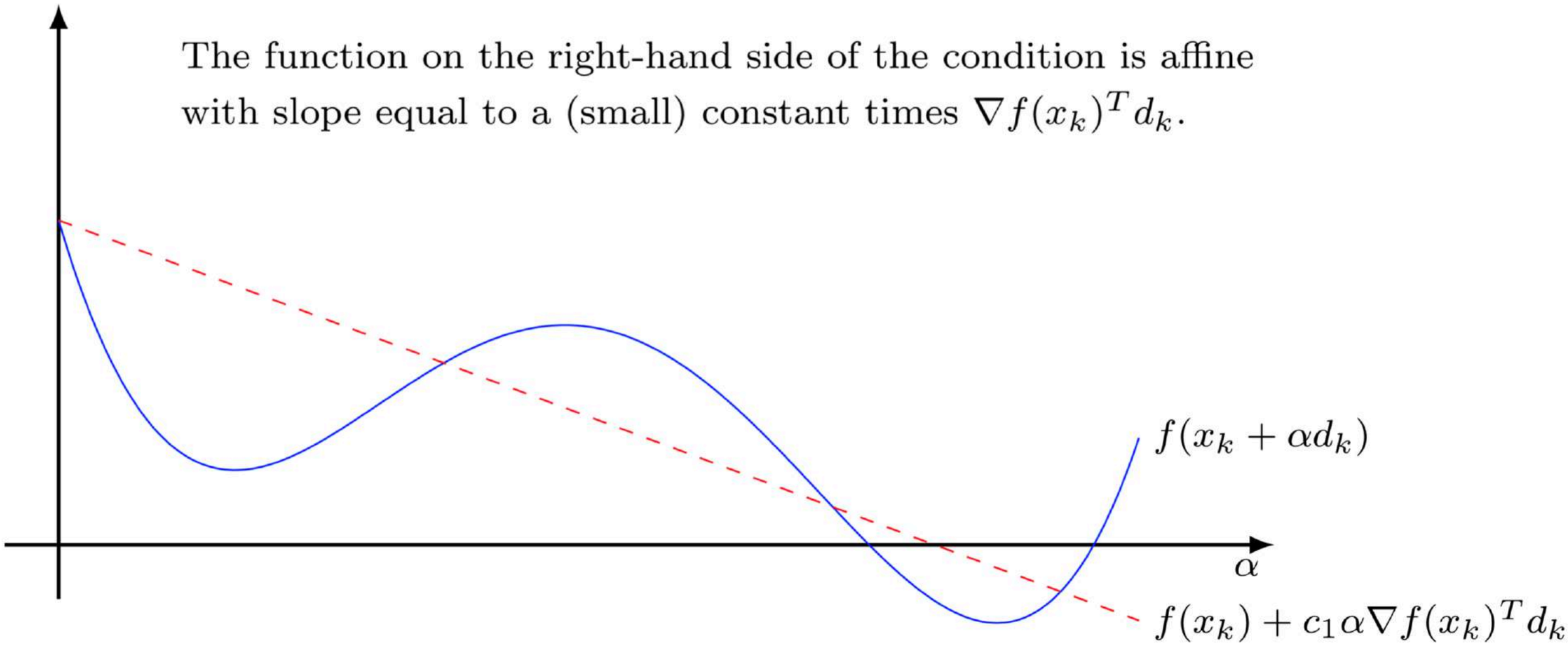
for some small $c_{ls} = 10^{-2}$.

Simple and tends to work pretty well in practice.

Interpretation of Backtracking line search

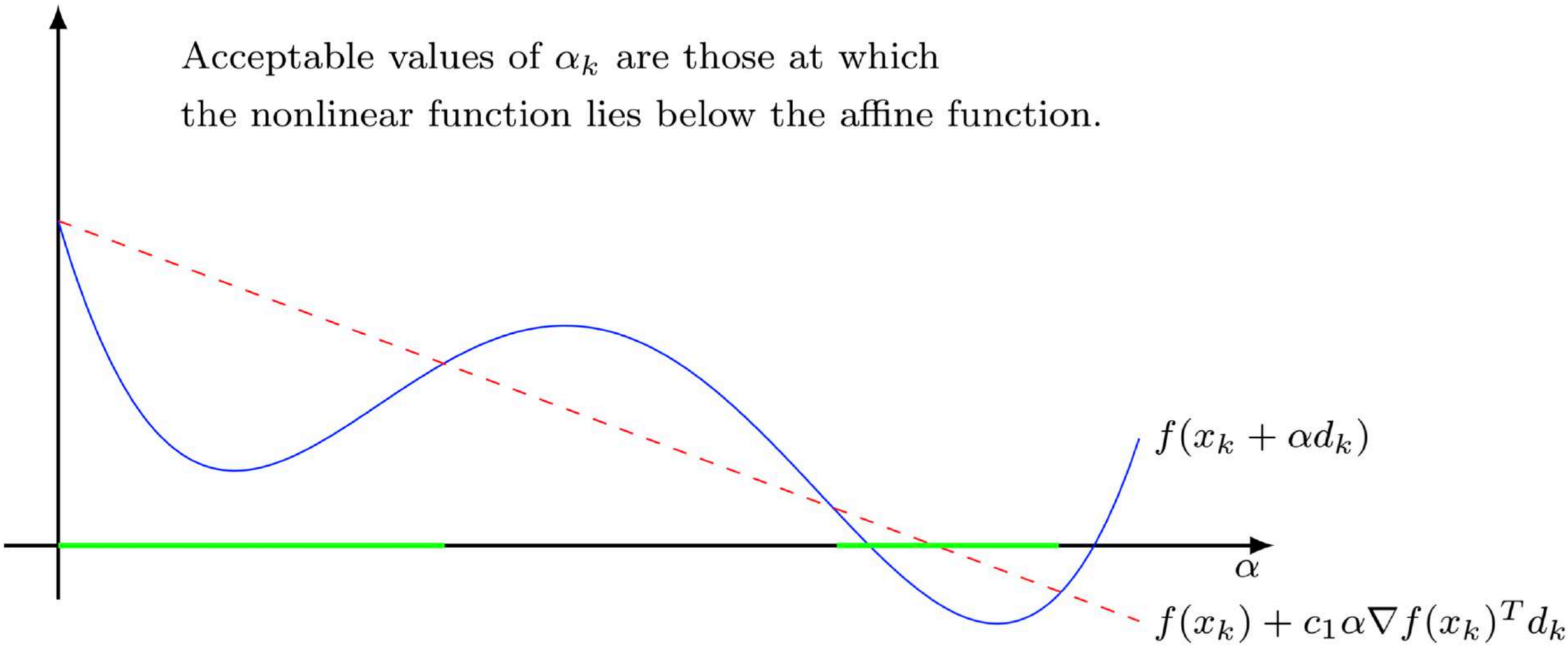
Armijo线搜索（backtracking回溯）

The function on the right-hand side of the condition is affine with slope equal to a (small) constant times $\nabla f(x_k)^T d_k$.



For us $d^k = -\nabla f(x^k)$, $\gamma = 1/2$.

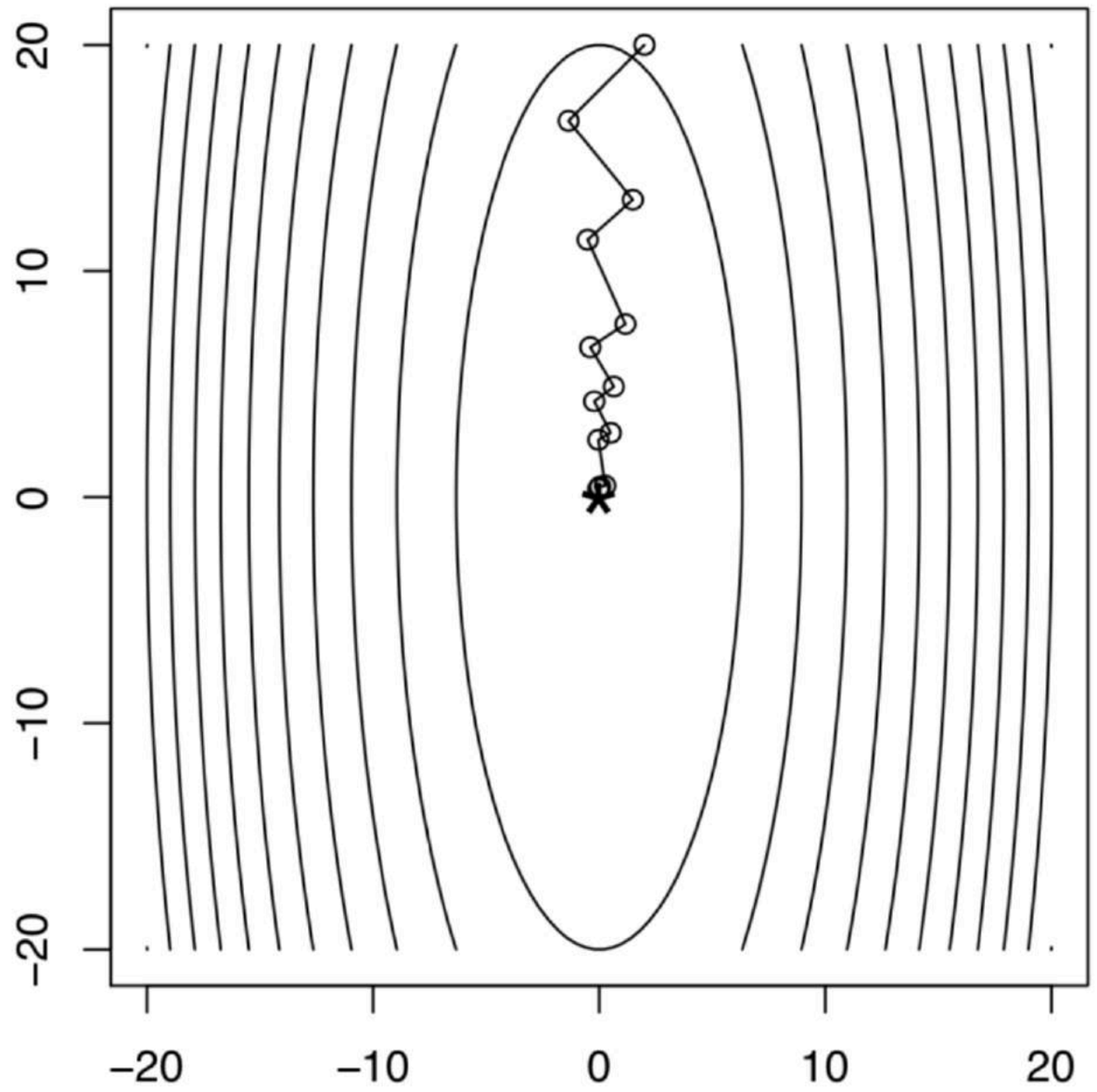
Interpretation of Backtracking line search



For us $d^k = -\nabla f(x^k)$, $\gamma = 1/2$.

Interpretation of Backtracking line search

Backtracking picks up roughly the right step size (13 steps):



Here $\gamma = 0.8$ ($\gamma \in (0.1, 0.8)$ is recommended, and often chosen as 0.5).

Exact line search 精确线搜索的劣势

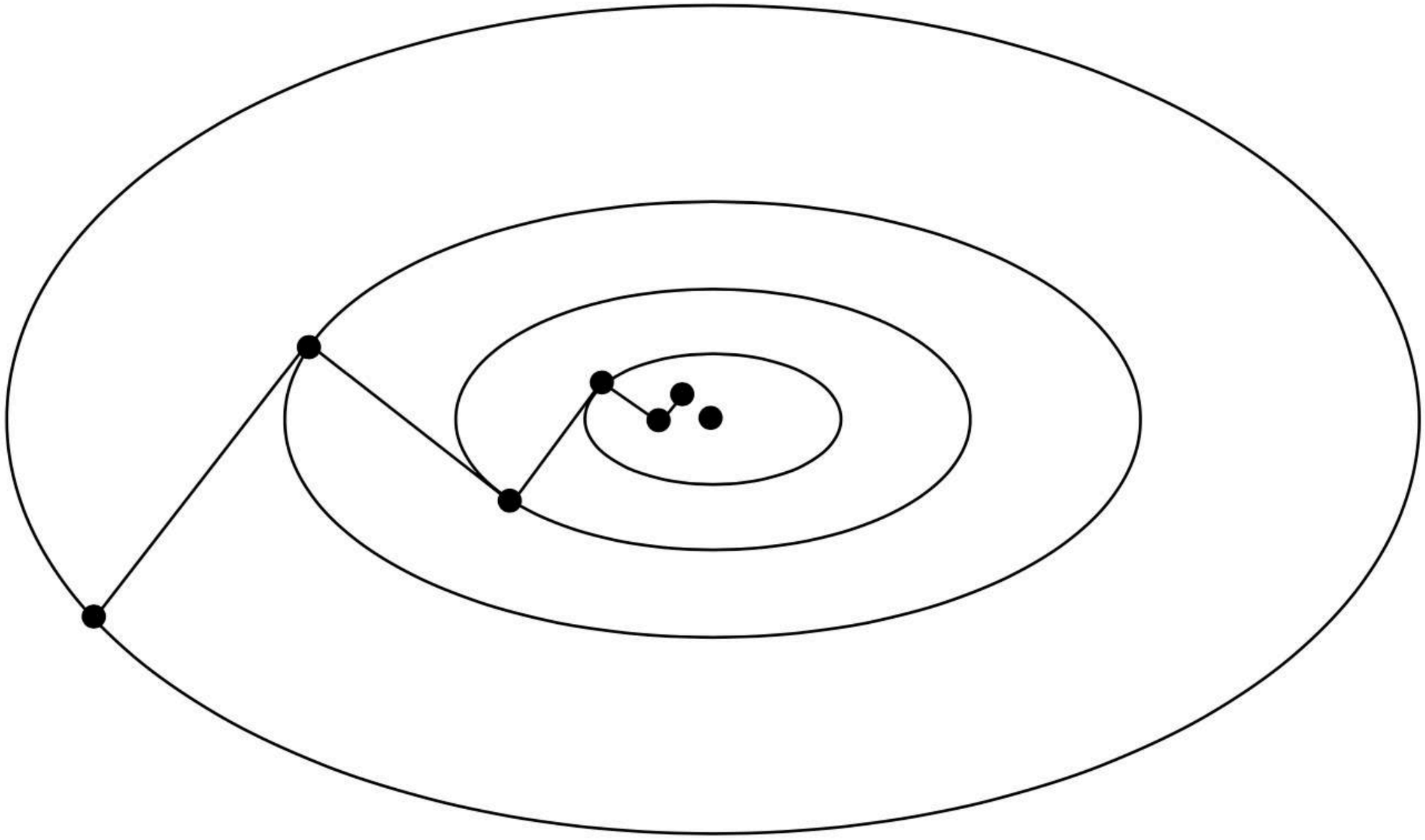
At each iteration, do the best we can along the direction of the gradient,

$$t = \arg\min_{s \geq 0} f(x - s\nabla f(x))$$

Usually not possible to do this minimization exactly

Approximations to exact line search are often not much more efficient than backtracking, and it's not worth it

最速下降并不是最好的选择（步长）



Convergence analysis 凸的时候是O(1/k)，这线性还是次线性?

Assume that $f: \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable, and additionally

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \quad \text{for any } x, y.$$

i.e., ∇f is Lipschitz continuous with constant $L > 0$

Theorem

Gradient descent with fixed step size $\alpha \leq 1/L$ satisfies

$$f(x^k) - f(x^*) \leq \frac{\|x^0 - x^*\|^2}{2\alpha k}.$$

i.e., gradient descent has convergence rate $O(1/k)$

i.e., to get $f(x^k) - f(x^*) \leq \epsilon$, need $O(1/\epsilon)$ iterations

Proof

Key steps:

- ∇f Lipschitz with constant $L \implies$

$$f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\|y - x\|^2 \quad \forall x, y$$

- Letting $x^+ = x - \alpha \nabla f(x)$ and taking $0 < \alpha \leq 1/L$, we then get

$$\begin{aligned} f(x^+) &\leq f(x) + \nabla f(x)^\top (x^+ - x) + \tfrac{L}{2}\|x^+ - x\|_2^2 \\ &= f(x) + \nabla f(x)^\top (x - \alpha \nabla f(x) - x) + \tfrac{L}{2}\|x - \alpha \nabla f(x) - x\|^2] \\ &= f(x) - \alpha \nabla f(x)^\top \nabla f(x) + \tfrac{L\alpha^2}{2}\|\nabla f(x)\|^2] \\ &= f(x) - (1 - \tfrac{1}{2}L\alpha)\alpha \|\nabla f(x)\|_2^2 \end{aligned}$$

- Notice that $-(1 - \frac{1}{2}L\alpha) = \frac{1}{2}L\alpha - 1 \leq \frac{1}{2}L(1/L) - 1 = \frac{1}{2} - 1 = -\frac{1}{2}$. Therefore,

$$f(x^+) \leq f(x) - \tfrac{1}{2}\alpha \|\nabla f(x)\|_2^2$$

$\implies$ objective monotonically decreases.

- By the convexity of f, we have

$$f(x) \leq f(x^*) + \nabla f(x)^\top (x - x^*)$$

- We can use this to further derive

$$\begin{aligned}
f(x^+) &\leq f(x) - \tfrac{1}{2}\alpha \|\nabla f(x)\|_2^2 \\
&\leq f(x^*) + \nabla f(x)^\top (x - x^*) - \frac{\alpha}{2}\|\nabla f(x)\|^2 \\
&= f(x^*) - \frac{1}{2\alpha}[-2(x - x^+)^\top (x - x^*) + \|x - x^+\|^2] \\
&= f(x^*) + \frac{1}{2\alpha}[2(x - x^+)^\top (x - x^*) - \|x - x^* - (x^+ - x^*)\|^2] \\
&= f(x^*) + \frac{1}{2\alpha}[2(x - x^+)^\top (x - x^*) + 2(x - x^*)^\top (x^+ - x^*) \\
&\quad - \|x - x^*\|^2 - \|x^+ - x^*\|^2] \\
&= f(x^*) + \frac{1}{2\alpha}(\|x - x^*\|^2 - \|x^+ - x^*\|^2)
\end{aligned}$$

算法定义

- Summing over iterations:

$$\sum_{i=1}^{k}(f(x^i) - f(x^*)) \leq \frac{1}{2\alpha}(\|x^0 - x^*\|^2 - \|x^k - x^*\|^2)$$
$$\leq \frac{1}{2\alpha}\|x^0 - x^*\|^2$$

- Since $f(x^k)$ is non-increasing

$$f(x^k) - f(x^*) \leq \frac{1}{k}\sum_{i=1}^{k}(f(x^i) - f(x^*)) \leq \frac{\|x^0 - x^*\|^2}{2\alpha k}$$

Convergence analysis for backtracking (Optional)

Same assumption, $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable, and ∇f is Lipschitz continuous with constant $L > 0$

Same rate for a step size chosen by backtracking search

Theorem

Gradient descent with backtracking line search satis- fies

$$f(x^k) - f(x^*) \leq \frac{\|x^0 - x^*\|^2}{2\alpha_{\min} k}$$

with $\alpha_{\min} = \min\{1, \gamma/L\}$.

If γ is not too small, then we don't lose much compared to fixed step size (γ/L vs $1/L$)

$$f(x - \alpha\nabla f(x)) > f(x) - \frac{\alpha}{2}\|\nabla f(x)\|_2^2,$$

Only have to clarify the minimum stepsize is bounded by some constant. On the other hand, we have for any $\alpha \leq 1/L$,

$$f(x^+) \leq f(x) - \tfrac{1}{2}\alpha\|\nabla f(x)\|_2^2$$

So that the Line Search process terminate with $\alpha \geq \gamma\frac{1}{L}$ or no backtracking happens with $\alpha = 1$. Hence $\alpha_{\min} = \min\{1, \gamma/L\}$.

Linear Convergence for Strongly Convex Cases (Optional)

Strong convexity of f means for some $\mu > 0$,

$$\nabla^2 f(x) \succeq \mu I \quad \text{for any } x$$

Better lower bound than that from usual convexity

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2}\|y - x\|^2 \quad \forall x, y$$

Under Lipschitz assumption as before, and also strong convexity

Theorem

For L-smooth and μ-strongly convex f, Gradient descent with fixed step size $\alpha \leq 2/(\mu + L)$ satisfies

$$f(x^k) - f(x^*) \leq \left(\frac{L/\mu - 1}{L/\mu + 1}\right)^{2k} \frac{L}{2}\|x^0 - x^*\|^2.$$

Since $\exp(-x) \geq 1 - x$ for every x, we get:

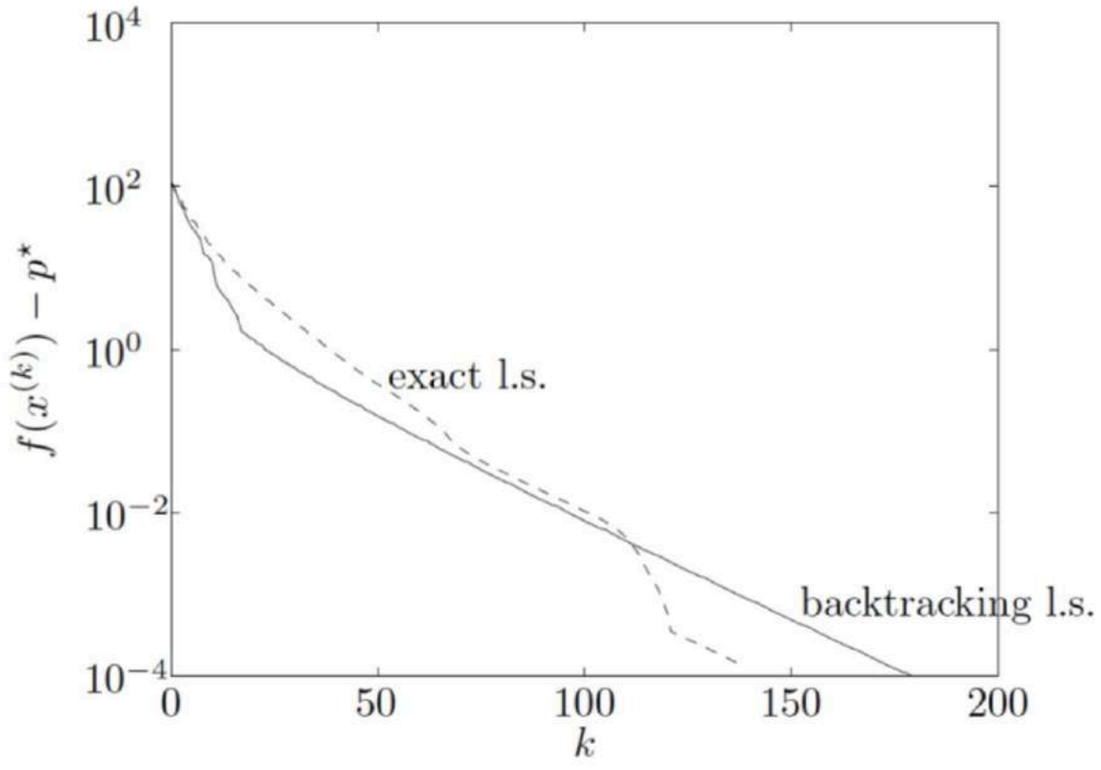
$$\left(\frac{L/\mu - 1}{L/\mu + 1}\right)^2 = \left(1 - \frac{4L\mu}{(L+\mu)^2}\right) \implies \left(\frac{L/\mu - 1}{L/\mu + 1}\right)^2 \leq \exp\left(-\frac{4L\mu}{(L+\mu)^2}\right)$$

$$\implies f(x^k) - f(x^*) \leq \frac{L}{2} \exp\left(-\frac{4L\mu k}{(L+\mu)^2}\right) \|x^0 - x^*\|^2$$

i.e., rate with strong convexity is exponentially fast!

i.e., to get $f(x^k) - f(x^*) \leq \epsilon$, need $O(\log(1/\epsilon))$ iterations

Called linear convergence, because looks linear on a semi-log plot:



Rate constant depends adversely on condition number L/μ (higher condition number) $\implies$ slower rate)

Key Steps of the Proof

Lemma

Let f be L-smooth and μ-strongly convex. Then for all $x, y \in \mathbb{R}^n$, one has

$$(\nabla f(x) - \nabla f(y))^\top (x - y) \geq \frac{\mu L}{\mu + L} \|x - y\|^2 + \frac{1}{\mu + L} \|\nabla f(x) - \nabla f(y)\|^2.$$

Proof.

Note that $\phi(x) := f(x) - \frac{\mu}{2}\|x\|^2$ is convex and $(L - \mu)$-smooth, so that (using a previous lemma)

$$[\nabla \phi(x) - \nabla \phi(y)]^\top (x - y) \geq \frac{1}{L - \mu} \|\nabla \phi(x) - \nabla \phi(y)\|^2$$

implying

$$[\nabla f(x) - \mu x - (\nabla f(y) - \mu y)]^\top (x - y) \geq \frac{1}{L - \mu} \|\nabla f(x) - \mu x - (\nabla f(y) - \mu y)\|^2.$$

Rearranging we have the desired result.

□

Key Steps of the Proof

Now we are ready to prove the linear convergence of GD. Indeed recall that smoothness can be defined via the inequality:

$$f(x) - f(y) \leq \nabla f(y)^\top (x - y) + \frac{L}{2}\|x - y\|^2,$$

$$\implies f(x^k) - f(x^*) \leq \nabla f(x^*)^\top (x^k - x^*) + \frac{L}{2}\|x^k - x^*\|^2$$

By strongly convexity,

$$\begin{aligned}
\|x^k - x^*\|^2 &= \|x^{k-1} - \alpha\nabla f(x^{k-1}) - x^*\|^2 \\
&= \|x^{k-1} - x^*\|^2 - 2\alpha\nabla f(x^{k-1})^\top (x^{k-1} - x^*) + \alpha^2\|\nabla f(x^{k-1})\|^2 \\
&\leq \left(1 - 2\frac{\alpha\mu L}{L+\mu}\right)\|x^{k-1} - x^*\|^2 + \left(\alpha^2 - 2\frac{\alpha}{L+\mu}\right)\|\nabla f(x^{k-1}) - \nabla f(x^*)\|^2 \\
&\leq \left(1 - 2\frac{\alpha\mu L}{L+\mu}\right)\|x^{k-1} - x^*\|^2 + \left(\alpha^2 - 2\frac{\alpha}{L+\mu}\right)L^2\|x^{k-1} - x^*\|^2 \\
&= \left(\frac{L+\mu - 2\alpha L\mu + \alpha^2 L^2(L+\mu) - 2\alpha L^2}{L+\mu}\right)\|x^{k-1} - x^*\|^2 \\
&= \left(\frac{\alpha^2 L^2(L+\mu) - 2\alpha L(L+\mu) + L + \mu}{L+\mu}\right)\|x^{k-1} - x^*\|^2 \\
&= \left(\alpha^2 L^2 - 2\alpha L + 1\right)\|x^{k-1} - x^*\|^2 \\
&= (\alpha L - 1)^2\|x^{k-1} - x^*\|^2 = 0, \quad \text{if we set } \alpha = 1/L, \text{ so what is wrong??}
\end{aligned}$$

- If $0 \leq \alpha \leq \frac{2}{L+\mu}$,

$$\|x^k - x^*\|^2 \leq \left(1 - \frac{2\alpha\mu L}{L+\mu} + (\alpha^2 - \frac{2\alpha}{L+\mu})L^2\right)\|x^{k-1} - x^*\|^2 = (\alpha L - 1)^2\|x^k - x^*\|^2$$

$$\implies \alpha^* = \min(2/(L+\mu), 1/L) = 2/(L+\mu)$$

- If $\alpha \geq \frac{2}{L+\mu}$,

$$\|x^k - x^*\|^2 \leq \left(1 - \frac{2\alpha\mu L}{L+\mu}\right)\|x^{k-1} - x^*\|^2$$

$$\implies \alpha^* = 2/(L+\mu)$$

-

$$\implies \|x^k - x^*\|^2 \leq \left(\frac{L/\mu - 1}{L/\mu + 1}\right)^2 \|x^{k-1} - x^*\|^2 \quad \text{with } \alpha = 2/(L+\mu)$$

$$\implies \|x^k - x^*\|^2 \leq \left(\frac{L/\mu - 1}{L/\mu + 1}\right)^{2k} \|x^0 - x^*\|^2.$$

$$\implies f(x^k) - f(x^*) \leq \frac{L}{2}\left(\frac{L/\mu - 1}{L/\mu + 1}\right)^{2k} \|x^0 - x^*\|^2$$

- The number L/μ is called the condition number of f.

How about backtracking line search in strongly convex cases? (Optional)

Remember the backtracking line search terminates with $\alpha_{\min} = \min\{1, \gamma/L\}$.

Theorem

For L-smooth and μ-strongly convex f, Gradient descent with backtracking line search attains linear convergence

$$f(x^k) - f(x^*) \leq (1 - \mu\alpha_{\min})^k [f(w^0) - f^*] = (1 - \min\{\tfrac{\mu\gamma}{L}, \mu\})^k [f(w^0) - f^*].$$

Question: in case we have $\min\{\frac{\mu\gamma}{L}, \mu\} = \mu$, do we have $\mu \leq 1$?

$\min\{\frac{\mu\gamma}{L}, \mu\} = \mu \implies \mu \leq \gamma\frac{\mu}{L} \leq \gamma < 1$

How about backtracking line search in strongly convex cases? (Optional)

Alternatively, we have $\alpha_{\min} = \min\{\frac{\gamma}{L}, 1\} \leq 1/L \leq \frac{2}{L+\mu}$. Previously, we have shown that

If $0 \leq \alpha \leq \frac{2}{L+\mu}$,

$$\|x^k - x^*\|^2 \leq \left(1 - \frac{2\alpha\mu L}{L+\mu} + (\alpha^2 - \frac{2\alpha}{L+\mu})L^2\right)\|x^{k-1} - x^*\|^2 = (\alpha L - 1)^2\|x^k - x^*\|^2$$

$$\implies \|x^k - x^*\|^2 \leq (\alpha_{\min} L - 1)^2\|x^0 - x^*\|^2.$$

$$\implies f(x^k) - f(x^*) \leq \frac{L}{2}(\alpha_{\min} L - 1)^{2k}\|x^0 - x^*\|^2$$

If $\alpha_{\min} = 1$, then $\gamma/L \geq 1 \implies L \leq \gamma < 1$, and

$$\implies f(x^k) - f(x^*) \leq \frac{L}{2}(1 - L)^{2k}\|x^0 - x^*\|^2$$

If $\alpha_{\min} = \gamma/L$, then

$$\implies f(x^k) - f(x^*) \leq \frac{L}{2}(1 - \gamma)^{2k}\|x^0 - x^*\|^2$$

Overall, we have

$$\boxed{f(x^k) - f(x^*) \leq \frac{L}{2}\max\left[1 - L, 1 - \gamma\right]^{2k}\|x^0 - x^*\|^2}$$

How realistic are these conditions?

How realistic is Lipschitz continuity of ∇f?

- This means $\nabla^2 f(x) \preceq LI$
- E.g., consider $f(x) = \frac{1}{2}\|y - Ax\|^2$ (linear regression). Here $\nabla^2 f(x) = A^\top A$, so ∇f Lipschitz with $L = \sigma_{\max}^2(A) = \|A\|^2$

How realistic is strong convexity of f?

- Recall this is $\nabla^2 f(x) \succeq dI$
- E.g., again consider $f(x) = \frac{1}{2}\|y - Ax\|^2$, so $\nabla^2 f(x) = A^\top A$ and we need $d = \sigma_{\min}^2(A)$, $L = \sigma_{\max}^2(A)$
- If A is wide, then $\sigma_{\min}(A) = 0$, and f can't be strongly convex
- Even if $\sigma_{\min}(A) > 0$, can have a very large condition number $\kappa(A) = \sqrt{\kappa(A^T A)} = \sqrt{L/d}$

Practicalities

Pros and cons:

- Pro: simple idea, and each iteration is cheap
- Pro: Very fast for well-conditioned, strongly convex problems
- Con: Often slow, because interesting problems aren't strongly convex or well-conditioned
- Con: can't handle nondifferentiable functions

Acceleration

- There are accelerated gradient methods for strongly-convex functions. They improve the rate to

$$f(w^k) - f^* \leq \left(1 - \sqrt{\frac{\mu}{L}}\right)^k [f(w^0) - f^*]$$

  which is a faster linear convergence rate.

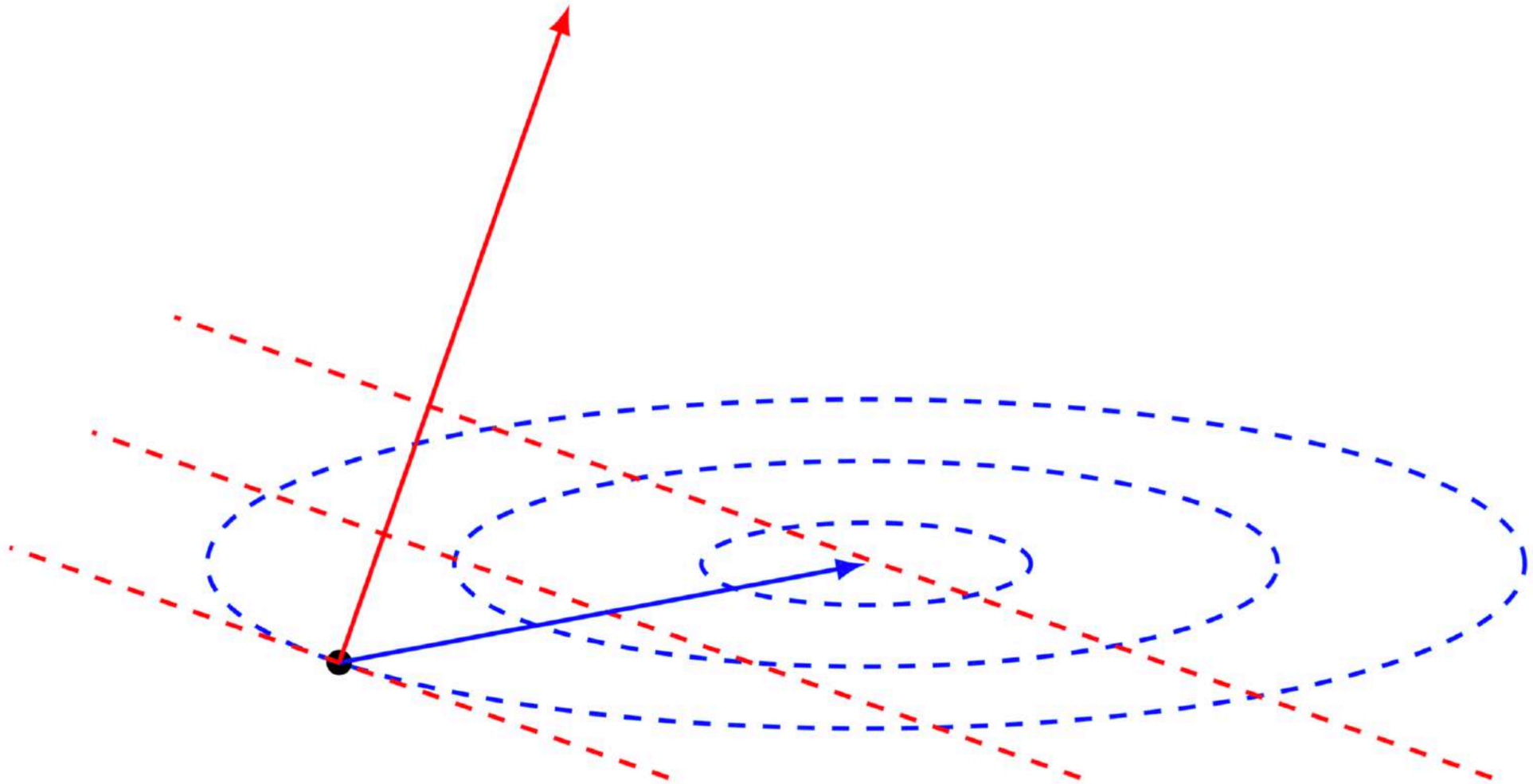
- Alternately, Newton's method achieves superlinear convergence rate.
  - Under strong-convexity and using both ∇f and $\nabla^2 f$ being Lipschitz.
  - But unfortunately this gives a superlinear iteration cost.

- There are also linear-time approximations to Newton:
  - Barzilai-Borwein step-size for gradient descent.
  - Limited-memory Quasi-Newton methods like L-BFGS.
  - Hessian-free Newton methods.

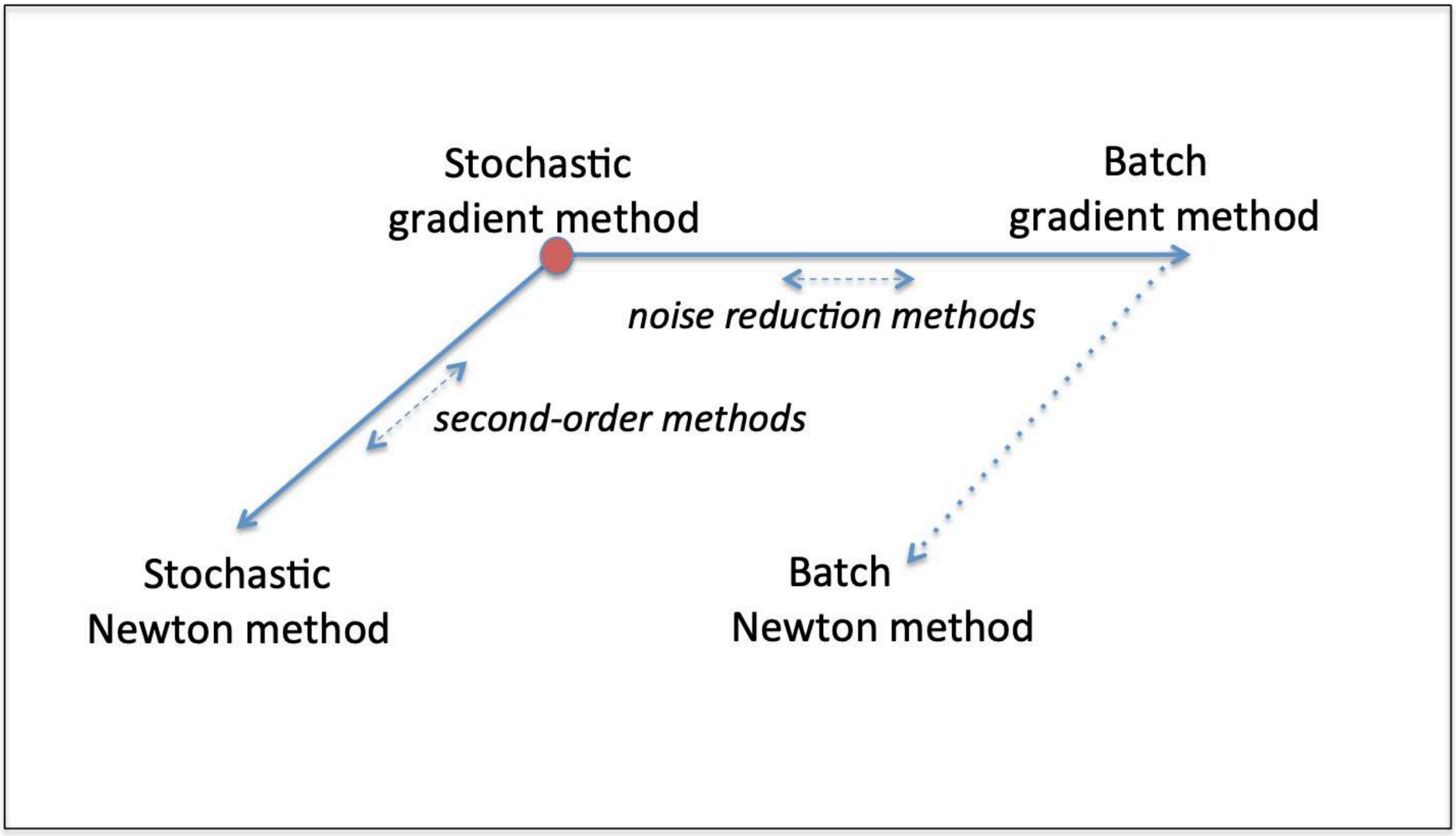
- Work amazing for many problems, but don't achieve superlinear convergence.

Keep in mind that steepest may not be wise

最速下降并不是最好的选择（方向、步长）



Newton method


Stochastic
gradient method
Batch
gradient method
noise reduction methods
second-order methods
Stochastic
Newton method
Batch
Newton method

Systems of equations v.s. Nonlinear Optimization

In most cases, solving the nonlinear optimization is to solve nonlinear equation

$$\min_{x \in \mathbb{R}^n} f(x) \quad \longrightarrow \quad \nabla f(x) = 0$$

We can also transform a system of equations to a nonlinear optimization problem

$$F(x) = 0 \quad \longrightarrow \quad \min_{x \in \mathbb{R}^n} \frac{1}{2} \|F(x)\|_2^2$$

But, generally we don't take this transformation...

Basic iteration of Newton Method

For nonlinear optimization $\min_{x \in \mathbb{R}^n} f(x)$

$$x^{k+1} \leftarrow x^k - \alpha [\nabla^2 f(x^k)]^{-1} \nabla f(x^k)$$

For system of equations $F(x) = 0$

$$x^{k+1} \leftarrow x^k - \alpha [\nabla F(x^k)]^{-1} F(x^k)$$

They can be derived in a similar way...

For our purpose, we want to $\min_{x \in \mathbb{R}^n} E_{in}(x)$

Or solve $\nabla E_{in}(x) = 0$

Let's focus on solving the equations…

Intuitions:

We can motivate Newton's Method in 3 ways (that are basically all the same).

- At the current point $(x_k, F(x_k))$, draw a tangent line until it hits the x-axis; call that point x_{k+1}.
- Create an affine model of F at x_k:

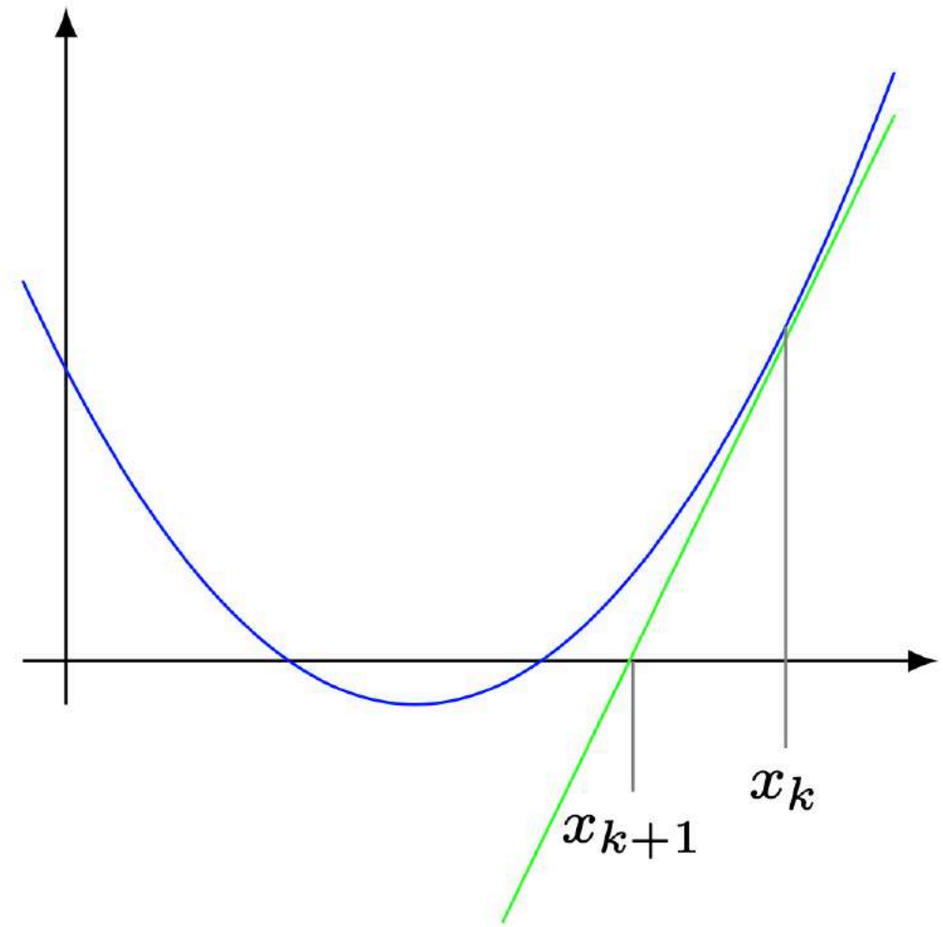
$$m_k(x) = F(x_k) + F'(x_k)(x - x_k);$$

call x_{k+1} the solution to $m_k(x) = 0$.

- Write the Taylor series of F at x_k:

$$F(x) = F(x_k) + F'(x_k)(x - x_k) + \tfrac{1}{2}F''(x_k)(x - x_k)^2 + \cdots;$$

approximate $F(x)$ with the affine portion, solve the resulting affine equation, and call the solution x_{k+1}.



Intuition:

The "affine model" is perhaps the best interpretation.

- We can write an equation, and then approximate an integral:

$$\begin{aligned} F(x) = & \; F(x_k) + \int_{x_k}^{x} F'(z)dz \\ \approx & \; F(x_k) + F'(x_k)(x - x_k) =: m_k(x). \end{aligned}$$

- Solving for x in $m_k(x) = 0$ yields the familiar formula:

$$x_{k+1} = x_k - \frac{F(x_k)}{F'(x_k)}.$$

Will it work??? Yes! (sometimes)

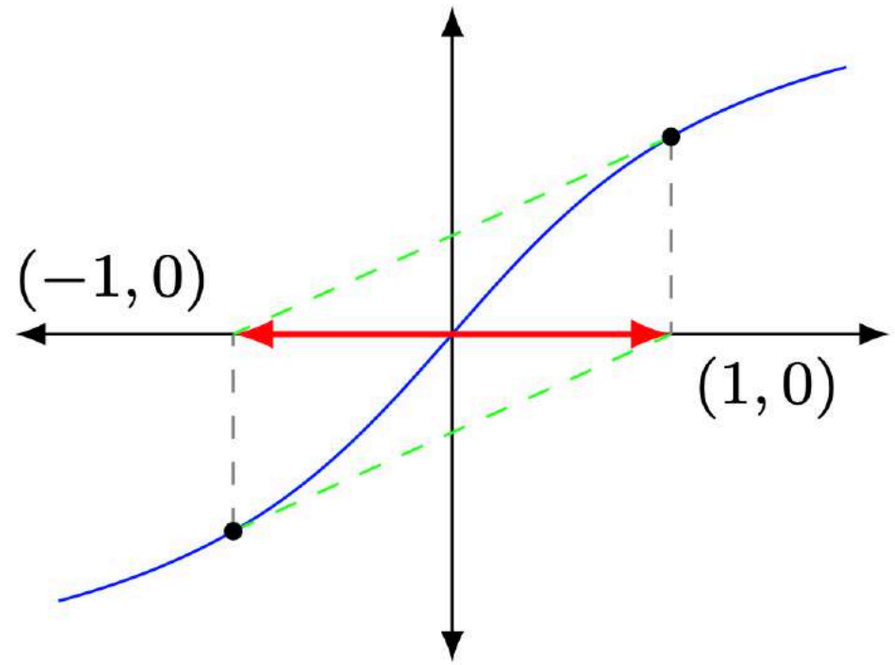
(Original) Newton's method is not globally convergent

Newton's method can fail in many ways:

- Certain starting points can lead to cycling and even divergence.
- May have $F'(x_k) = 0$. (So what?)
- $F(x_k)$ may be undefined/imaginary.

Also, it might not fail, but in some situations it can converge very slooooooowly...

Still, it is very powerful.



Multi-variate cases

Suppose we have $F : \mathbb{R}^n \to \mathbb{R}^n$.

- Form an affine model of the function at x_k:

$$\begin{aligned} F(x) = &\ F(x_k) + \int_{[x_k, x]} \nabla F(z)^T \cdot dz \\ \approx &\ F(x_k) + \nabla F(x_k)^T (x - x_k). \end{aligned}$$

- Solve for the step/displacement d_k (i.e., let $d = x - x_k$):

$$\nabla F(x_k)^T d = -F(x_k).$$

- Update the iterate

$$x_{k+1} = x_k + d_k.$$

- We can think about this as simultaneously solving individual affine models for each of the components of F:

$$F^i(x) \approx F^i(x_k) + \nabla F^i(x_k)^T d \quad \Rightarrow \quad \nabla F^i(x_k)^T d = -F^i(x_k),\ i = 1, \ldots, n.$$

Example:

Suppose we aim to solve

$$F(x) = \begin{bmatrix} x_1^2 + x_1 x_2 \\ e^{x_1} - x_2 \end{bmatrix} = 0.$$

(Can we solve this analytically? How many solutions does it have?)

- Pick a starting point, say $x_0 = (1, 1)$.
- Evaluate $F(x)$ and $\nabla F(x)^T$:

$$F(x) = \begin{bmatrix} x_1^2 + x_1 x_2 \\ e^{x_1} - x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ e - 1 \end{bmatrix}$$

$$\text{and} \quad \nabla F(x)^T = \begin{bmatrix} 2x_1 + x_2 & x_1 \\ e^{x_1} & -1 \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ e & -1 \end{bmatrix}.$$

- Solve the Newton system:

$$\begin{bmatrix} 3 & 1 \\ e & -1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = - \begin{bmatrix} 2 \\ e - 1 \end{bmatrix}.$$

- Update $x_1 = x_0 + d_0$, reevaluate, solve, update, reevaluate, solve....

Example:

```
>> x = newton('example',[1;1]);
==================================
  k      ||F(x)||        ||d||
==================================
   0   2.6368e+000   6.5211e-001
   1   6.5261e-001   2.9418e-001
   2   8.5037e-002   5.8346e-002
   3   4.1779e-003   4.2638e-003
   4   2.2801e-005   2.6147e-005
   5   8.2119e-010   -----------
==================================
>> x

x =

    0.0000
    1.0000
```

```
>> x = newton('example',[-1;1]);
==================================
  k      ||F(x)||        ||d||
==================================
   0   6.3212e-001   6.5353e-001
   1   4.6100e-002   4.1159e-002
   2   2.4495e-004   2.2103e-004
   3   6.9278e-009   -----------
==================================
>> x

x =

   -0.5671
    0.5671
```

Example:

Suppose we aim to solve

$$F(x) = \arctan(x) = 0.$$

(Can we solve this analytically? How many solutions does it have?)

- Pick a starting point, say $x_0 = 1$.
- Evaluate $F(x)$ and $\nabla F(x)^T$:

$$F(x) = \arctan(x) \approx 0.7854$$

$$\text{and} \quad \nabla F(x)^T = \frac{1}{1+x^2} = \frac{1}{2}.$$

- Solve the Newton system:

$$\frac{1}{2}d = -0.7854.$$

- Update $x_1 = x_0 + d_0$, reevaluate, solve, update, reevaluate, solve....

Example:

```
>> x = newton('example2',1.391);
================================
  k     ||F(x)||       ||d||
================================
   0  9.4749e-001  2.7808e+000
   1  9.4708e-001  2.7763e+000
   2  9.4598e-001  2.7647e+000
   3  9.4308e-001  2.7342e+000
   4  9.3539e-001  2.6555e+000
   5  9.1489e-001  2.4597e+000
   6  8.5946e-001  2.0165e+000
   7  7.0810e-001  1.2272e+000
   8  3.5527e-001  4.0417e-001
   9  3.3148e-002  3.3184e-002
  10  2.4303e-005  2.4303e-005
  11  9.5692e-015  ----------
================================
>> x

x =

 -9.5692e-015
```

```
>> x = newton('example2',1.392);
================================
  k     ||F(x)||       ||d||
================================
   0  9.4783e-001  2.7844e+000
   1  9.4798e-001  2.7859e+000
   2  9.4835e-001  2.7899e+000
   3  9.4934e-001  2.8006e+000
   4  9.5194e-001  2.8288e+000
   5  9.5878e-001  2.9047e+000
   6  9.7661e-001  3.1160e+000
   7  1.0221e+000  3.7576e+000
   8  1.1304e+000  6.2188e+000
   9  1.3314e+000  2.3681e+001
  10  1.5198e+000  5.8438e+002
  11  1.5690e+000  5.0052e+005
  12  1.5708e+000  3.9262e+011
  13  1.5708e+000  2.4214e+023
  14  1.5708e+000  9.2101e+046
  15  1.5708e+000  1.3324e+094
  16  1.5708e+000  2.7888e+188
  17  1.5708e+000  Inf
  18  1.5708e+000  Inf
  19  NaN  ----------
================================
>> x

x =

   NaN
```

Convergent of Newton's method

- Newton's method converges quadratically! (under nice assumptions)

```
>> x = newton('example',[-1;1]);
================================
  k      ||F(x)||        ||d||
================================
  0   6.3212e-001   6.5353e-001
  1   4.6100e-002   4.1159e-002
  2   2.4495e-004   2.2103e-004
  3   6.9278e-009   -----------
================================
>> x

x =

   -0.5671
    0.5671
```

- (Why else do you think something so old is still around!)
- We will go through the entire proof.
- First, we need to discuss our assumptions...

Convergent of Newton's method

Assumptions

For some point $x_* \in \mathbb{R}^n$ *such that* $F(x_*) = 0$, *the following hold:*

- F *is continuously differentiable in an open convex set* $\mathcal{X} \subseteq \mathbb{R}^n$ *with* $x_* \in \mathcal{X}$.
- *The Jacobian of* F *at* x_* *is invertible and is bounded in norm by* $M > 0$, *i.e.,*

$$\|(\nabla F(x_*)^T)^{-1}\|_2 \leq M.$$

- *For some neighborhood of* x_* *with radius* $r > 0$ *contained in* $\mathcal{X}$, *i.e.,*

$$\mathbb{B}(x_*, r) := \{x \in \mathbb{R}^n \mid \|x - x_*\|_2 \leq r\} \in \mathcal{X},$$

the Jacobian of $F(x)$ *is Lipschitz continuous with constant* L *in* $\mathbb{B}(x_*, r)$.

Theorem

There exists $\epsilon > 0$ *such that for all* $x_0 \in \mathbb{B}(x_*, \epsilon)$, *the sequence defined by*

$$\begin{aligned} \nabla F(x_k)^T d_k &= -F(x_k) \\ x_{k+1} &= x_k + d_k, \quad k = 0, 1, 2, \ldots \end{aligned}$$

is well-defined, converges to x_*, *and for some* $c > 0$ *satisfies*

$$\|x_{k+1} - x_*\|_2 \leq c\|x_k - x_*\|_2^2.$$

proof...

Proof, part 1.

Consider $\bar{\epsilon} > 0$ such that, with $\|x_0 - x_*\|_2 \leq \bar{\epsilon}$, the Jacobian $\nabla F(x_0)^T$ is nonsingular. This guarantees that the iteration is well-defined at x_0. Let

$$\epsilon := \min\{\bar{\epsilon}, \tfrac{1}{2ML}\} > 0.$$

Recall that if A is nonsingular and $\|A^{-1}(B - A)\|_2 < 1$, then B is nonsingular and

$$\|B^{-1}\|_2 \leq \frac{\|A^{-1}\|_2}{1 - \|A^{-1}(B - A)\|_2}.$$

Thus, since $\nabla F(x_*)^T$ is nonsingular and

$$\begin{aligned}\|\nabla F(x_*)^{-T}(\nabla F(x_0)^T - \nabla F(x_*)^T)\|_2 \leq\ & \|\nabla F(x_*)^{-T}\|_2 \|\nabla F(x_0)^T - \nabla F(x_*)^T\|_2 \\ \leq\ & ML\|x_0 - x_*\|_2 \leq ML\epsilon \leq \tfrac{1}{2},\end{aligned}$$

we know that $\nabla F(x_0)^T$ is nonsingular and

$$\|\nabla F(x_0)^{-T}\|_2 \leq \frac{\|\nabla F(x_*)^{-T}\|_2}{1 - \|\nabla F(x_*)^{-T}(\nabla F(x_0)^T - \nabla F(x_*)^T)\|_2} \leq 2M.$$

proof...

Proof, part 2.

We now show that for some $c > 0$ we have $\|x_1 - x_*\|_2 \leq c\|x_0 - x_*\|_2^2$. (This is the relationship we need for quadratic convergence, which will be nice if we can show that we actually converge!) The difference between x_1 and x_* can be written as

$$\begin{aligned} x_1 - x_* &= x_0 - x_* - \nabla F(x_0)^{-T} F(x_0) \\ &= x_0 - x_* - \nabla F(x_0)^{-T}(F(x_0) - F(x_*)) \\ &= \nabla F(x_0)^{-T}(F(x_*) - \underbrace{F(x_0) - \nabla F(x_0)^T(x_* - x_0)}_{\text{affine model of } F(x) \text{ at } x_0}). \end{aligned}$$

Recalling a result from multivariable calculus, we then find

$$\begin{aligned} \|x_1 - x_*\|_2 &\leq \|\nabla F(x_0)^{-T}\|_2 \|F(x_*) - F(x_0) - \nabla F(x_0)^T(x_* - x_0)\|_2 \\ &\leq (2M)(\tfrac{1}{2}L\|x_0 - x_*\|_2^2) \\ &\leq ML\|x_0 - x_*\|_2^2. \end{aligned}$$

Thus, the result holds for $c = ML$.

proof...

Proof, part 3.

Finally, we show that $\|x_1 - x_*\|_2 \leq \frac{1}{2}\|x_0 - x_*\|_2$. (This shows that $x_1 \in \mathbb{B}(x_*, \epsilon)$, so all of our results so far will continue to hold for $k = 1, 2, \ldots$, meaning that we converge and do so quadratically!) We have shown already that

$$\|x_1 - x_*\|_2 \leq ML\|x_0 - x_*\|_2^2,$$

and since $\|x_0 - x_*\|_2 \leq \epsilon \leq (2ML)^{-1}$ (by definition of ϵ), we have

$$\|x_1 - x_*\|_2 \leq \frac{1}{2}\|x_0 - x_*\|_2.$$

How to minimize $E_{in}(\mathbf{w})$ in logistic regression

Regression - pseudoinverse (analytic), from solving the normal equation

Logistic regression - analytic won't work

We can only numerically/iteratively set $\nabla E_{in}(\mathbf{w}) \rightarrow 0$

Note that $E_{in}(\mathbf{w})$ is convex

Fixed learning rate (batch) gradient descent

1: Initialize at step $k = 0$ to $\mathbf{w}_0$
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Compute the gradient $\mathbf{g}^k = \nabla E_{in}(\mathbf{w}^k)$
4: Move in the direction $\mathbf{v}^k = -\mathbf{g}^k$
5: Update the weights $\mathbf{w}_{k+1} = \mathbf{w}^k + \eta \mathbf{v}^k$
6: Iterate "until it is time to stop"
7: **end for**
8: Return the final weights

Cases with 0-1 labels

$$E_{in}(\mathbf{w}) = -\sum_{i=1}^{m} \left[y_i \ln \sigma(\mathbf{w}^T \mathbf{x}_i) + (1 - y_i) \ln \left(1 - \sigma(\mathbf{w}^T \mathbf{x}_i)\right) \right]$$

$$\mathbf{g} = \frac{d}{d\mathbf{w}} E_{in}(\mathbf{w}) = \sum_{i=1}^{m} (\sigma_i - y_i)\mathbf{x}_i = \mathbf{X}^T(\boldsymbol{\sigma} - \mathbf{y})$$

$$\mathbf{H} = \frac{d}{d\mathbf{w}} g(\mathbf{w})^T = \sum_{i=1}^{m} (\nabla_{\mathbf{w}} \sigma_i)(\mathbf{x}_i)^T$$

$$= \sum_{i=1}^{m} \sigma_i(1 - \sigma_i)\mathbf{x}_i(\mathbf{x}_i)^T = \mathbf{X}^T\mathbf{D}\mathbf{X}$$

where $\mathbf{D} = \mathsf{diag}(\sigma_i(1 - \sigma_i))$.

Newton's algorithm

Machine Leanring, A probabilistic perspective. Chapter 8

$$\begin{aligned}\mathbf{w}_{k+1} &= \mathbf{w}^k - \mathbf{H}^{-1}\mathbf{g} \\ &= \ldots \\ &= (\mathbf{X}^T\mathbf{D}^k\mathbf{X})^{-1}\mathbf{X}^T\mathbf{D}^k\mathbf{z}^k\end{aligned}$$

where we have defined the working response as

$$\mathbf{z}^k = \mathbf{X}\mathbf{w}^k + \mathbf{D}^{k^{-1}}(\mathbf{y} - \boldsymbol{\sigma}^k)$$

Newton's method: Iteratively reweighted least squares algorithm (IRLS)

$$
\begin{aligned}
\mathbf{w}_{k+1} &= \mathbf{w}^k - \mathbf{H}^{k^{-1}}\mathbf{g}^k \\
&= \mathbf{w}^k + (\mathbf{X}^T\mathbf{D}^k\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{y} - \boldsymbol{\sigma}^k) \\
&= (\mathbf{X}^T\mathbf{D}^k\mathbf{X})^{-1}[(\mathbf{X}^T\mathbf{D}^k\mathbf{X})\mathbf{w}^k + \mathbf{X}^T(\mathbf{y} - \boldsymbol{\sigma}^k)] \\
&= (\mathbf{X}^T\mathbf{D}^k\mathbf{X})^{-1}\mathbf{X}^T[\mathbf{D}^k\mathbf{X}\mathbf{w}^k + \mathbf{y} - \boldsymbol{\sigma}^k] \\
&= (\mathbf{X}^T\mathbf{D}^k\mathbf{X})^{-1}\mathbf{X}^T\mathbf{D}^k\mathbf{z}^k
\end{aligned}
$$

This is equivalent to

$$
\min_{\mathbf{w}} \quad \sum_{i=1}^{m} \mathbf{D}_i^k \left(\mathbf{z}_i^k - \mathbf{w}^T\mathbf{x}_i\right)^2
$$

where $\mathbf{z}_i^k = \mathbf{w}^{k^T}\mathbf{x}_i^k + \frac{y_i - \sigma_i^k}{\sigma_i^k(1-\mu_i^k)}$

Quasi-Newton methods

Quasi-Newton update satisfies the secant condition $\mathbf{H}^k\mathbf{s}^k = \mathbf{y}^k$, i.e.,

$$\mathbf{H}^k(\mathbf{x}^k - \mathbf{x}_{k-1}) = \nabla f^k - \nabla f_{k-1}$$

Interpretation: define second-order approximation at $\mathbf{x}^k$

$$m^k(\mathbf{z}) = f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{z} - \mathbf{x}^k) + \frac{1}{2}(\mathbf{z} - \mathbf{x}^k)^T\mathbf{H}^k(\mathbf{z} - \mathbf{x}^k)$$

secant condition implies that gradient of m^k agrees with f at $\mathbf{x}_{k-1}$:

$$\nabla m^k(\mathbf{x}_{k-1}) = \nabla f(\mathbf{x}^k) + \mathbf{H}^k(\mathbf{x}_{k-1} - \mathbf{x}^k) = \nabla f(\mathbf{x}_{k-1})$$

Stochastic Gradient Descent (SGD)
Cases with +1,-1 labels

A variation of GD that considers only the error on one data point

$$E_{in}(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m} \ln(1 + e^{-y_i \cdot \mathbf{w}^T \mathbf{x}}) = \frac{1}{m}\sum_{i=1}^{m} loss(\mathbf{w}, \mathbf{x}_i, y_i)$$

- Pick a random data point $(\mathbf{x}_{j(k)}, y_{j(k)})$
- Run an iteration of GD on $loss(\mathbf{w}, \mathbf{x}_{j(k)}, y_{j(k)})$

$$\mathbf{w}_{k+1} \leftarrow \mathbf{w}^k - \eta \nabla_{\mathbf{w}} loss(\mathbf{w}, \mathbf{x}_{j(k)}, y_{j(k)})$$

SGD of Logistic regression:

$$\mathbf{w}_{k+1} \leftarrow \mathbf{w}^k + y_{j(k)}\mathbf{x}_{j(k)} \left(\frac{\eta}{1 + e^{y_{j(k)} \mathbf{w}^T \mathbf{x}_{j(k)}}} \right)$$

Stochastic Gradient Descent (SGD)

SGD of Logistic regression:

$$\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + y_{j(k)}\mathbf{x}_{j(k)} \left(\frac{\eta}{1 + e^{y_{j(k)}\mathbf{w}^{k\,T}\mathbf{x}_{j(k)}}} \right)$$

Recall PLA:

$$\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + y_{j(k)}\mathbf{x}_{j(k)}$$

- The 'average' move is the same as GD
- Computation: fraction $1/m$ cheaper per step
- Stochastic: helps escape local minima
- Simple
- Similar to PLA