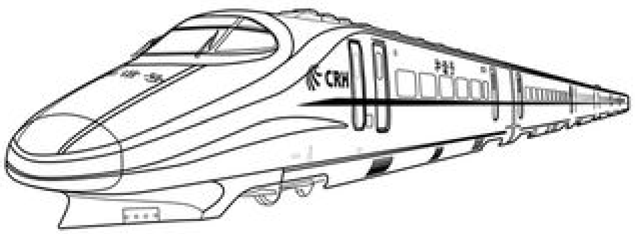
北京交通大學 BEIJING JIAOTONG UNIVERSITY

智能信息技术教育中心
The Education Center of Intelligence Information Technologies

人工智能基础

机器学习

耿阳李敖
2022年3月



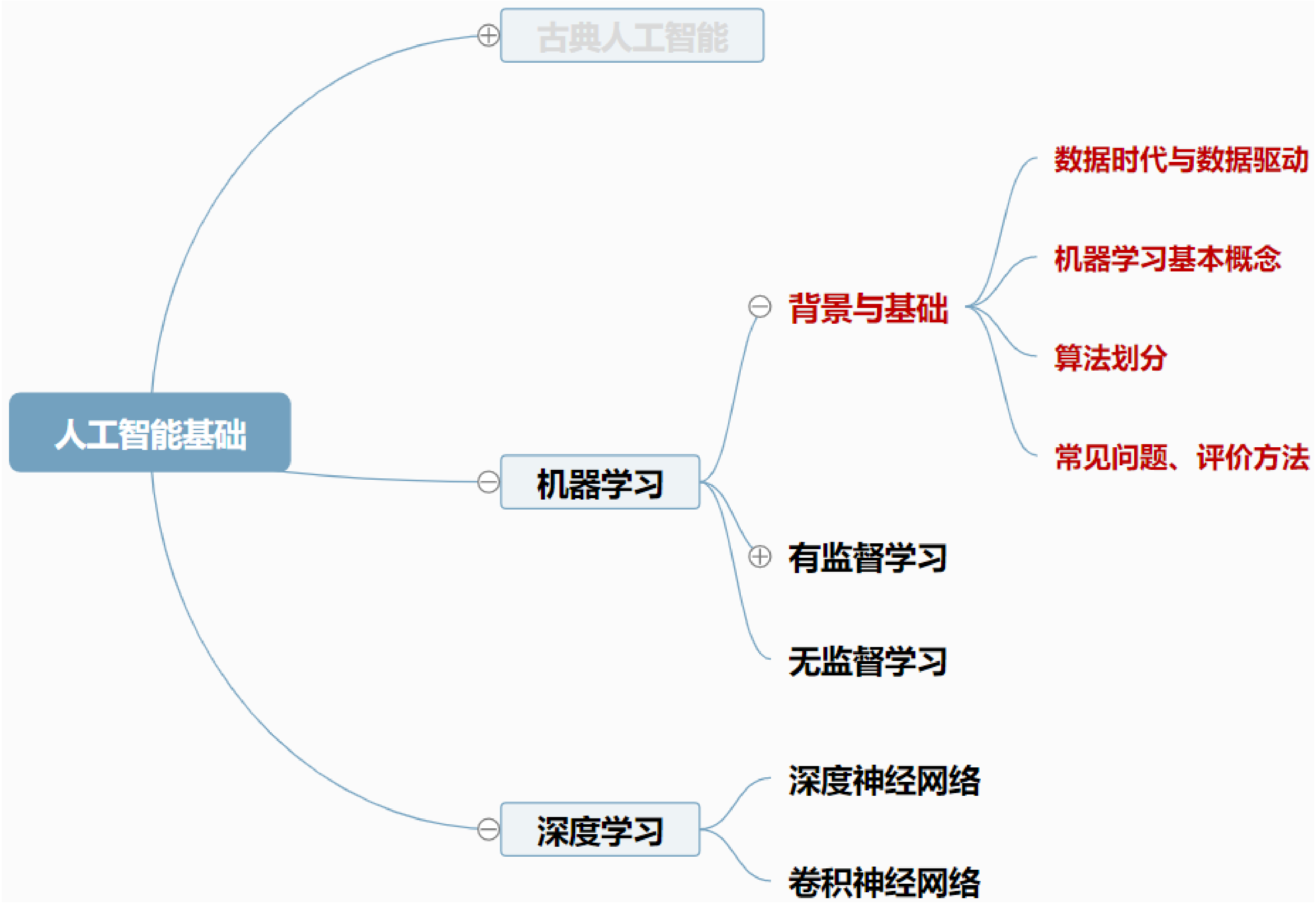
内容概要

- 机器学习基础回顾
- K-近邻算法
- 基础回归算法
- 总结

内容概要

- **机器学习基础回顾**
- K-近邻算法
- 基础回归算法
- 总结

前提回顾



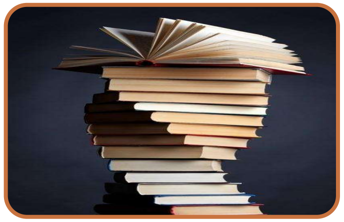
古典人工智能
人工智能基础
机器学习
背景与基础
数据时代与数据驱动
机器学习基本概念
算法划分
常见问题、评价方法
有监督学习
无监督学习
深度学习
深度神经网络
卷积神经网络

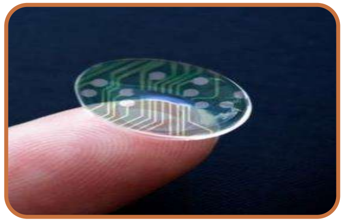
数据时代与数据驱动

数据的功能

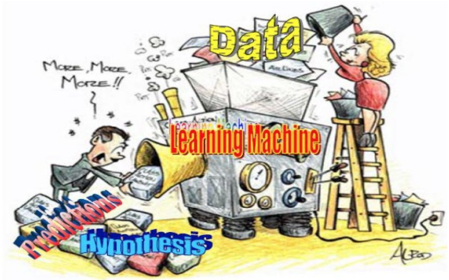
数据 → 信息 → 知识 → 智慧

数据的形式与来源



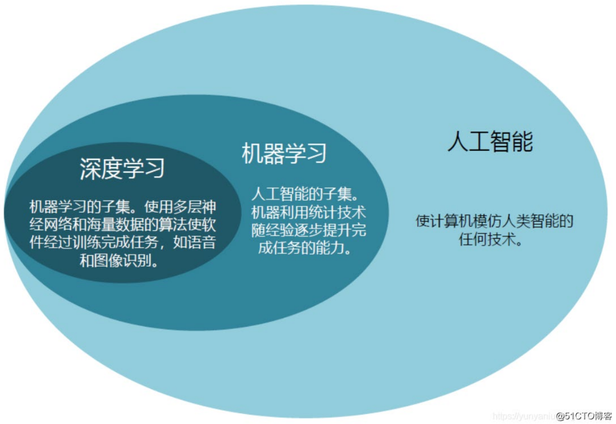



数据与机器学习之间的关系

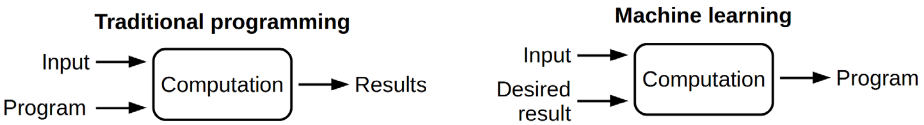


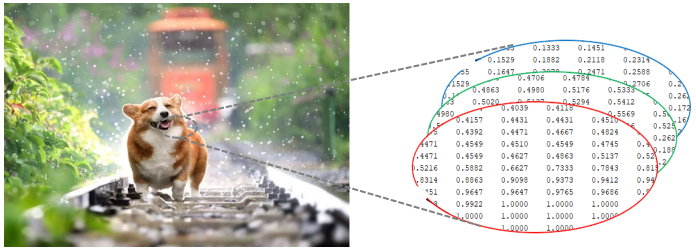
机器学习基础

机器学习在人工智能中扮演的角色



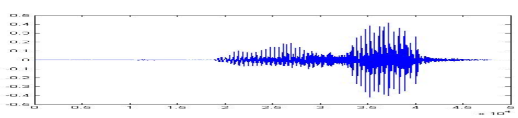
机器学习的概念与动机





机器学习的本质

≈ 基于数据学习一个映射函数

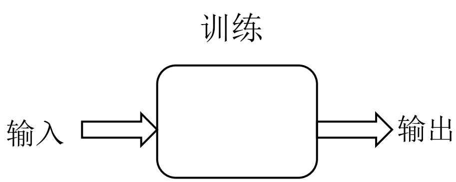
$f($  $)=$ “How are you”

$f($  $)=$ “Cat”

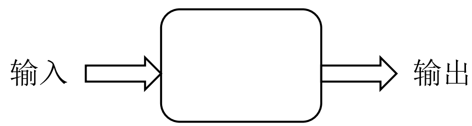
机器学习基础

机器学习分类

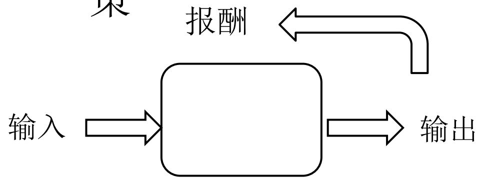
- 有监督学习
  - 显式学习方法
  - 数据具有明确的输出标记



- 无监督学习
  - 自助去理解数据，发现其中蕴含的模式或结构

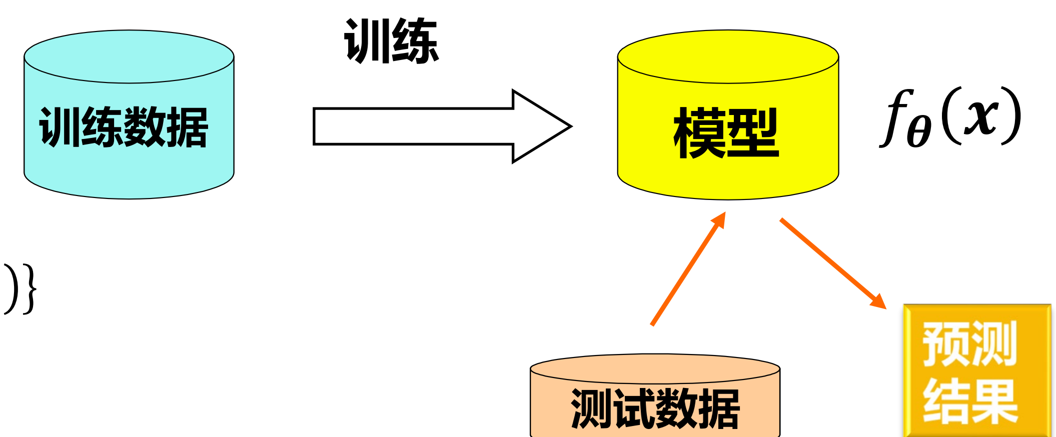


- 强化学习
  - 基于报酬的学习
  - 机器学习到如何在特定环境进行行为和决策



有监督学习框架

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = y$$

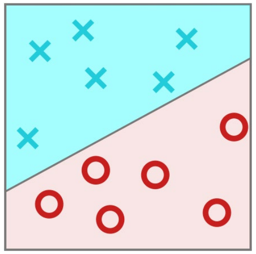


训练数据: $\{(\boldsymbol{x}^1, \hat{y}^1), (\boldsymbol{x}^2, \hat{y}^2), \ldots, (\boldsymbol{x}^N, \hat{y}^N)\}$

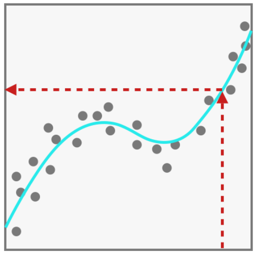
测试数据: $\{\boldsymbol{x}^{N+1}, \boldsymbol{x}^{N+2}, \ldots, \boldsymbol{x}^{N+M}\}$

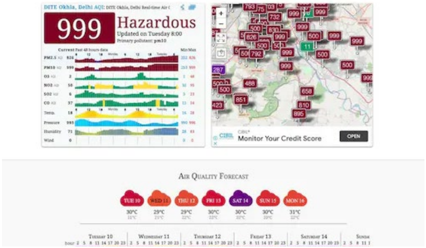
机器学习基础

有监督学习的两个基本任务



分类





回归

有监督学习四个要素

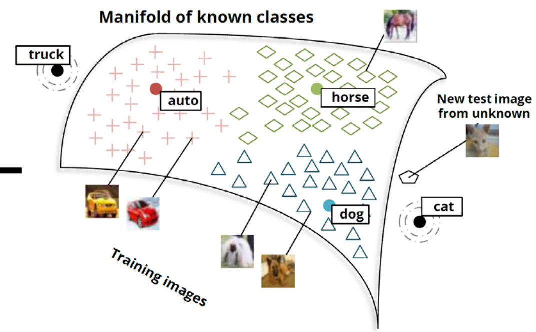
- 标注数据：对于给定的输入x，应该输出怎样的y
- 学习模型：怎样构建x与y之间的关系
- 损失函数：如何度量模型输出与目标输出之间的误差
- 优化算法：根据误差怎样更新模型使其获得更好的预测

机器学习基础

有监督学习挑战

$$\frac{1}{n}\sum_{i=1}^{n} Loss(y_i, f(x_i))$$

经验风险



$$\int_{x\times y} Loss(y, f(x))$$

期望风险

经验风险	期望风险	泛化能力	解决方法
经验风险小	期望风险大	*过拟合*	1. 增加数据量 2. 损失函数中添加正则项
经验风险大	期望风险大	*欠拟合*	1. 增加特征类型 2. 提升模型复杂度
经验风险小	期望风险小	**理想算法**	-

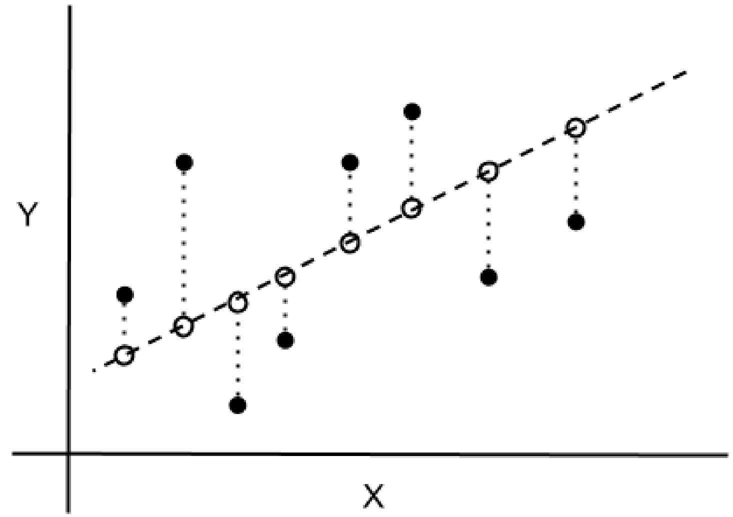
性能度量

回归任务

- 均方误差：$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$

- 绝对值误差：$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - f(x_i)|$

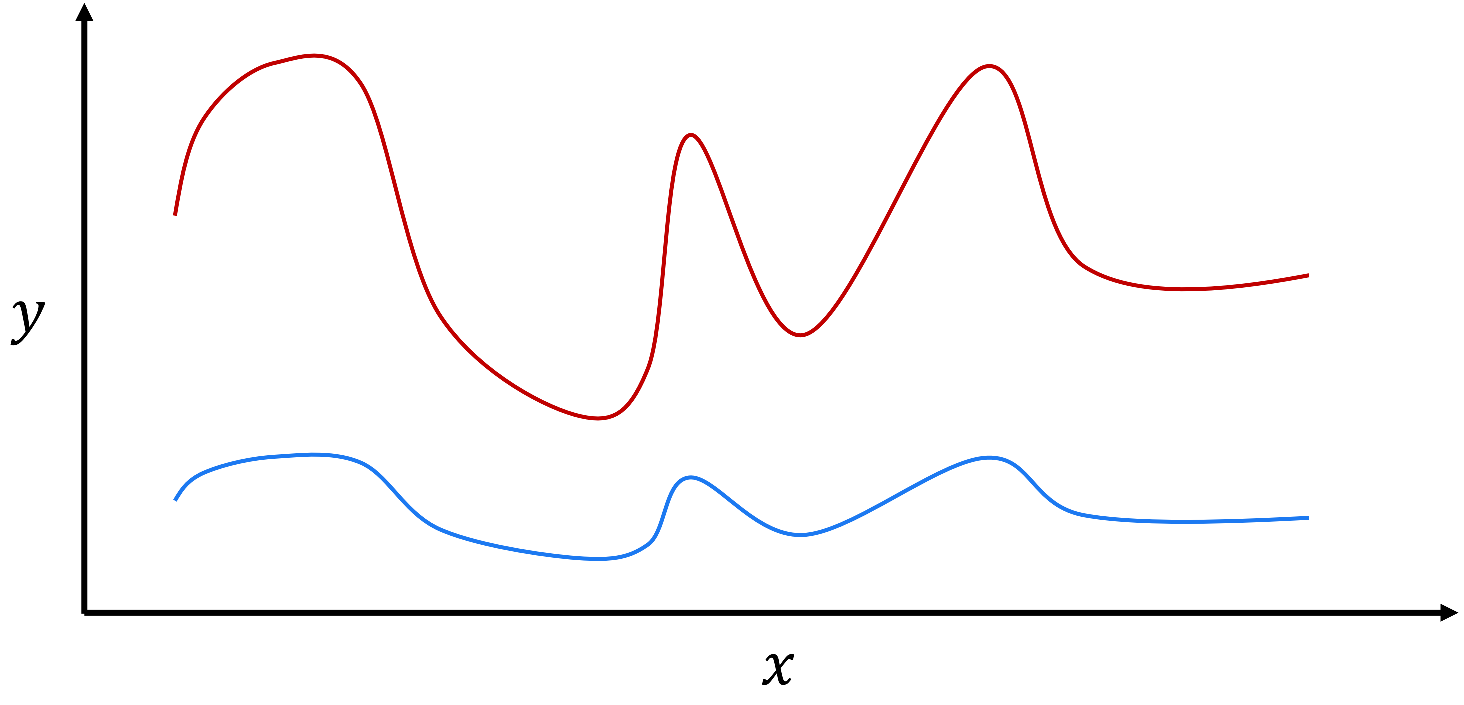
- 平均绝对百分比误差：$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - f(x_i)}{y_i}\right|$



性能度量

回归任务

◆ 皮尔逊相关系数： $PCC = \frac{\sum_{i=1}^{n}(y_i - \bar{y}_i)\left(f(x_i) - \overline{f(x_i)}\right)}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2 \sum_{i=1}^{n}\left(f(x_i) - \overline{f(x_i)}\right)^2}}$



性能度量

分类任务

- 错误率（error rate）： $$ER = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\{y_i \neq f(x_i)\}$$

- 精度（accuracy）： $$\mathrm{Acc} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\{y_i = f(x_i)\} = 1 - ER$$

性能度量

分类任务

按照每个类别来看可以得到“混淆矩阵”：

真实类别	预测类别	
	正例	反例
正例	True Positive (真正例)	False Negative (假反例)
反例	False Positive (假正例)	True Negative (真反例)

- 查准率（Precision）： $P = \frac{TP}{TP + FP}$
- 查全率/召回率（Recall）： $R = \frac{TP}{TP + FN}$

性能度量

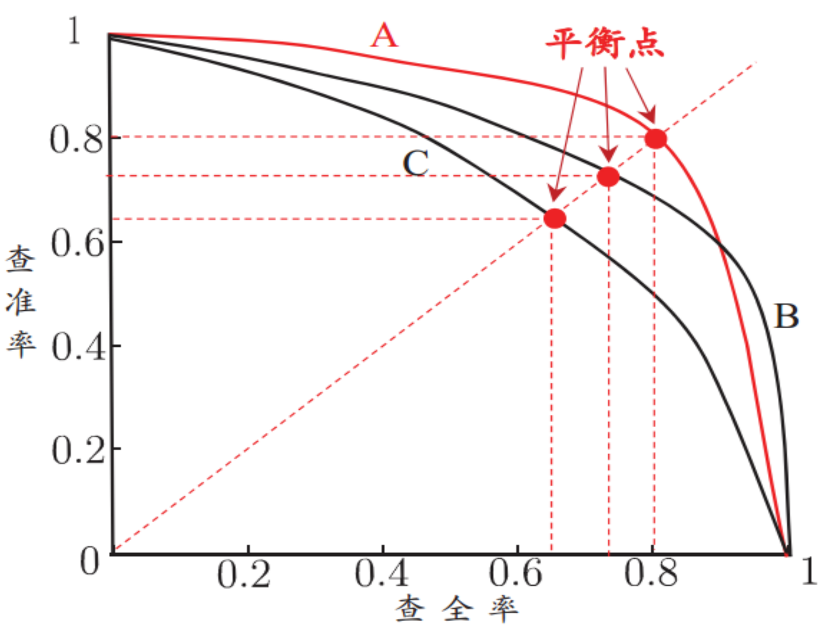
分类任务

- F1度量：

$$F1 = \frac{2 \times P \times R}{P + R}$$

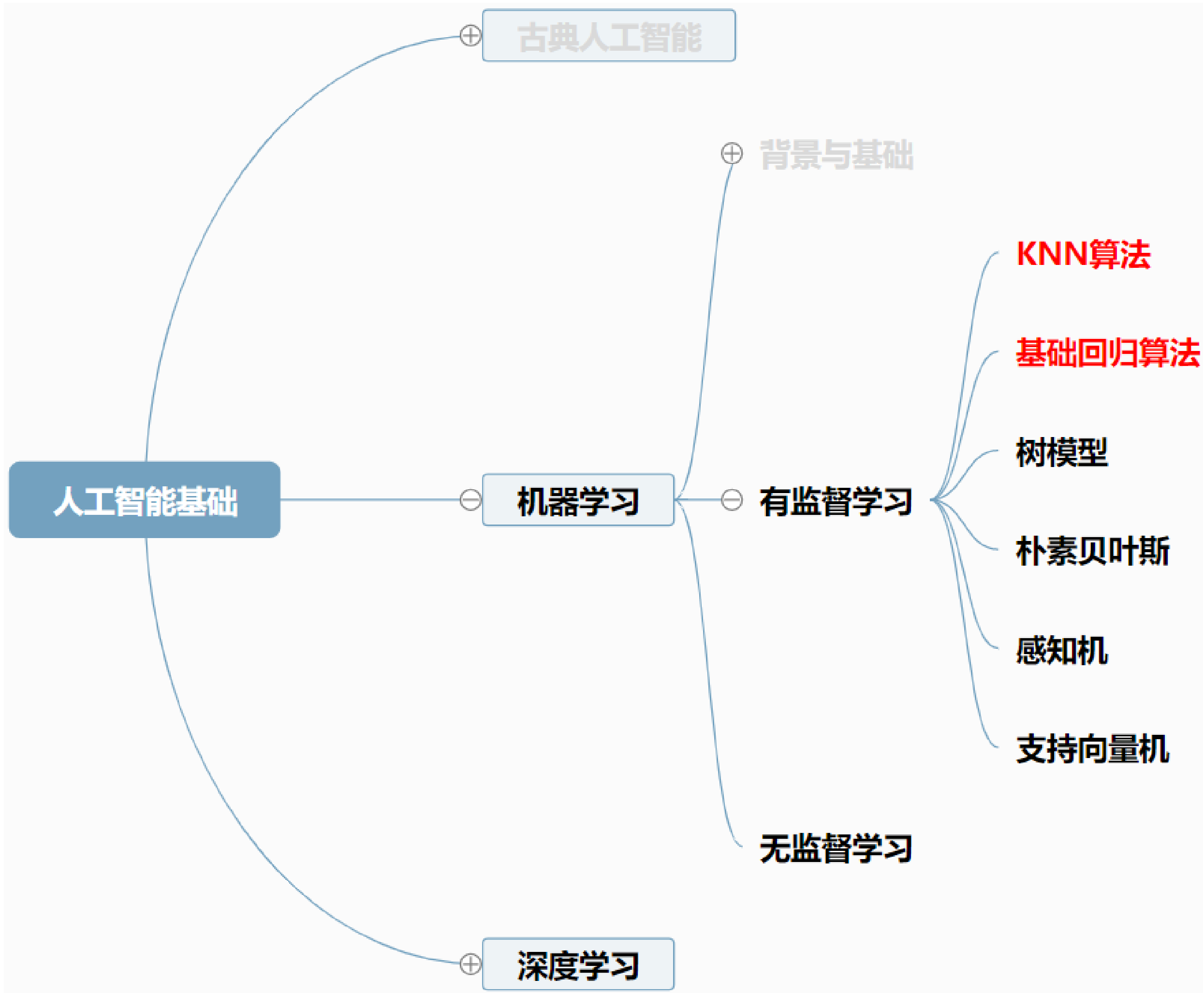
P与R的调和平均数

- PR曲线：



- 可通过比较P-R曲线下的面积；
- 利用平衡点（即P=R时的取值）；

新的内容



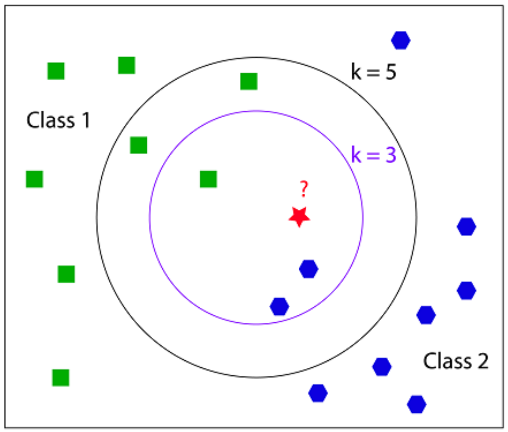
古典人工智能
背景与基础
KNN算法
基础回归算法
树模型
人工智能基础
机器学习
有监督学习
朴素贝叶斯
感知机
支持向量机
无监督学习
深度学习

内容概要

- 机器学习基础回顾
- **K-近邻算法**
- 基础回归算法
- 总结

K-近邻算法

问题引入



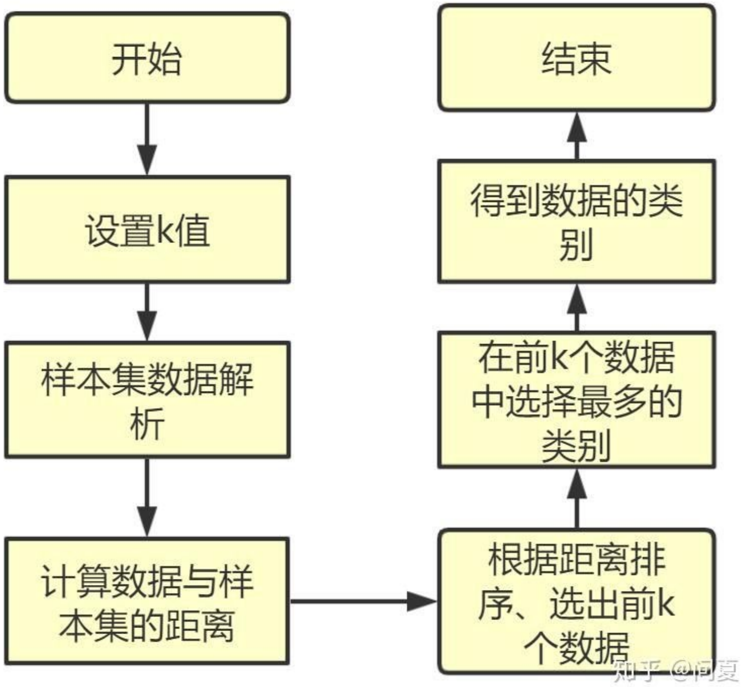
如图所示，平面上的点可以分成两类，新来的点应该属于哪一类呢?

“物以类聚，人以群分；欲知其人，先观其友”——古语

k近邻(k-Nearest Neighbor, kNN)学习是一种最直观的监督学习方法

K-近邻算法

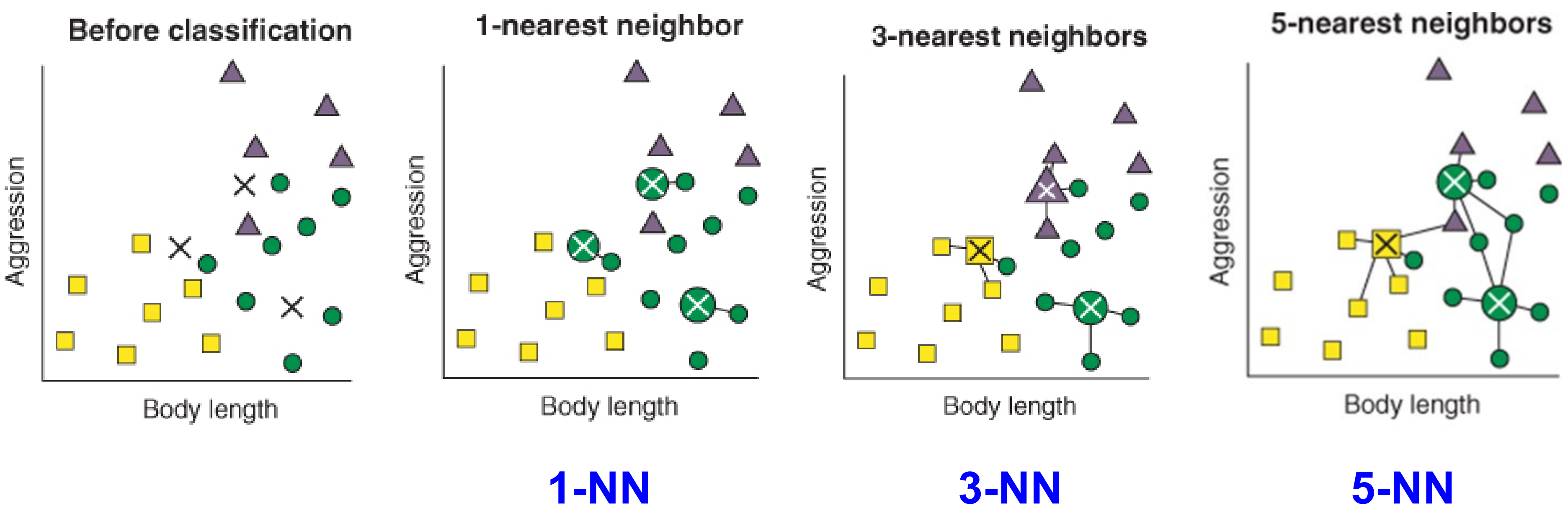
算法工作原理



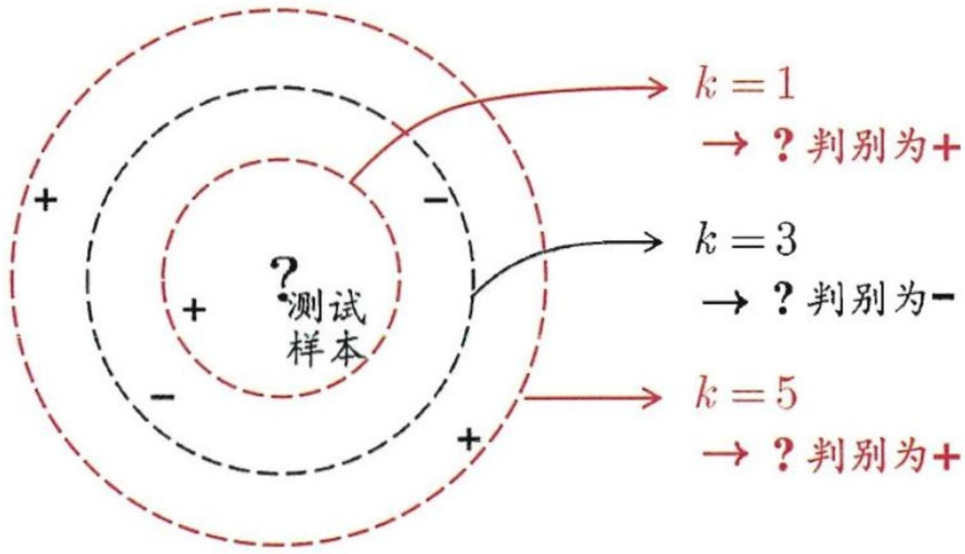
- 投票法：选择这k个样本中出现最多的类别标记作为预测结果。
- 平均法：将这k个样本的实值输出标记的平均值作为预测结果。

K-近邻算法

- **K=1， 最近邻算法：把与测试对象x_T最近的训练对象的类别赋给x_T**
- **K>1， “多数表决”：K个最接近的对象中出现频率最高的类别赋给x_T**



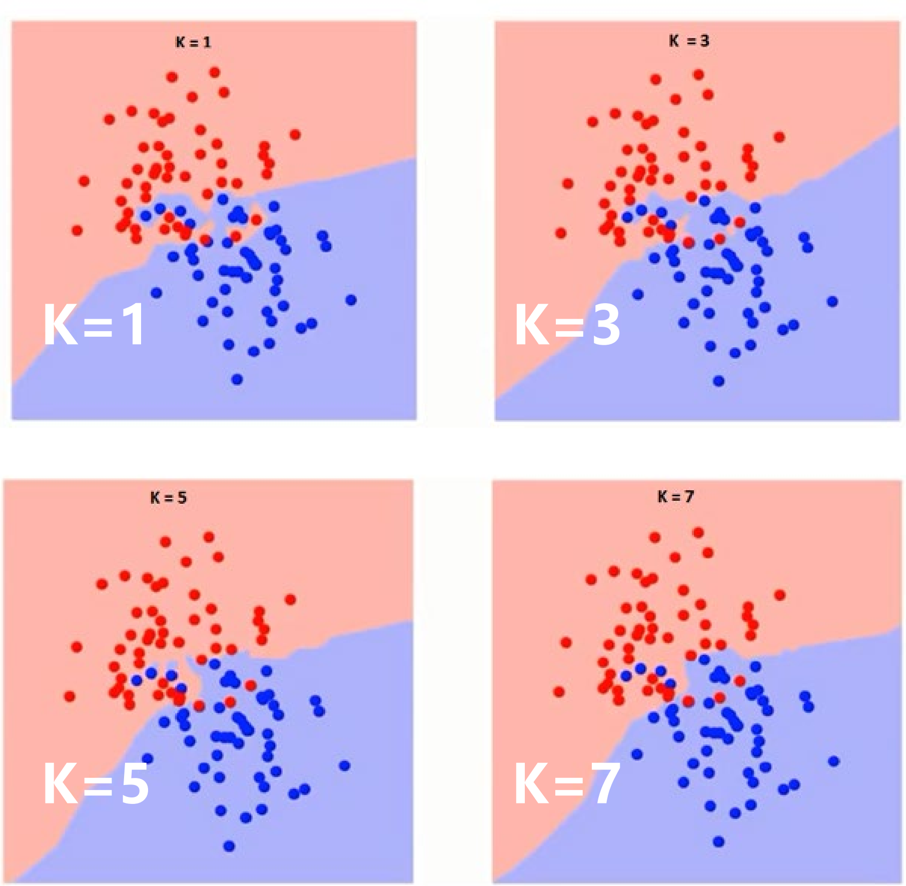
K-近邻算法



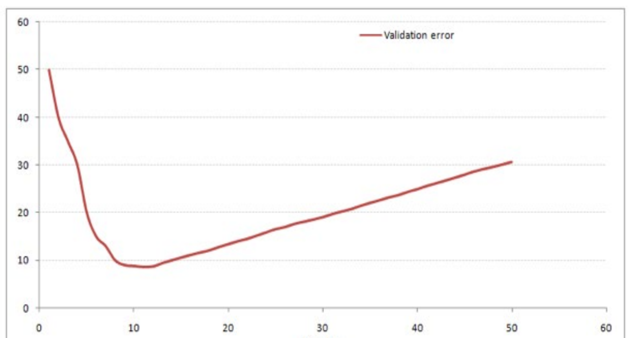
- k近邻分类器中的k是一个重要参数，当k取不同值时，分类结果会有显著不同。
- 另一方面，若采用不同的距离计算方式，则找出的“近邻”可能有显著差别，从而也会导致分类结果有显著不同

K-近邻算法

K近邻算法中的K的影响



测试样本错误率



K值

K-近邻算法

K近邻算法中的K的影响

- K值的选取是影响K近邻算法效果的主要因素。
- 不能使用K个近邻与查询对象的距离平方和来评估：此时总是倾向于选择K=1，K=1时算法易受噪声点和离群点的影响。
- K较小，容易被噪声影响，整体模型变得复杂，容易发生过拟合。
- K较大，近邻中出现很多其他类的点，容易发生错误，整体模型变得简单。
- 一般情况下，将K取一个较小的数值，而后使用交叉验证的方式来选取最优的K值。

K-近邻算法

K近邻算法中的距离度量

- **欧氏距离（Euclidean distance）**：$D(\mathbf{x},\mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$
- **一般距离定义**：对于任意实函数 $\boldsymbol{D}(\boldsymbol{x},\mathbf{y})$，其满足
  - 非负性：$D(\mathbf{x},\mathbf{y}) \geq 0$
  - 同一性：$D(\mathbf{x},\mathbf{y}) = 0 \iff \mathbf{x} = \mathbf{y}$
  - 对称性：$D(\mathbf{x},\mathbf{y}) = D(\mathbf{y},\mathbf{x})$
  - 三角不等式：$D(\mathbf{x},\mathbf{y}) \leq D(\mathbf{x},\mathbf{z}) + D(\mathbf{z},\mathbf{y})$

K-近邻算法

K近邻算法中的距离度量

- d维空间中的 **Minkowski 距离**（L_p距离）：

$$L_p(\mathbf{x},\mathbf{y}) = \left[\sum_{i=1}^{d} |x_i - y_i|^p\right]^{1/p}$$

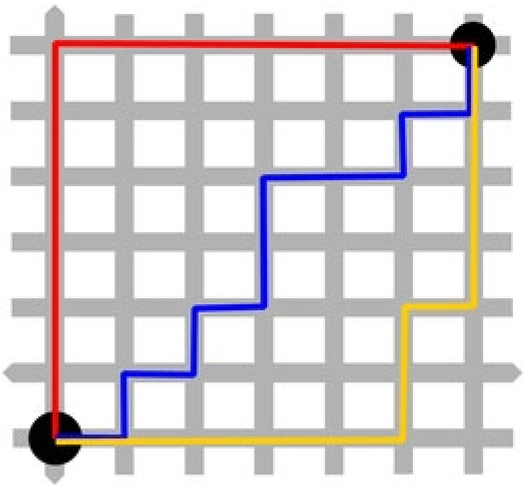
- L_1 距离：**曼哈顿距离（Manhattan distance）**
- L_2 距离：**欧式距离（Euclidean distance）**
- L_∞ 距离：**契比雪夫距离（Chebyshev distance）**

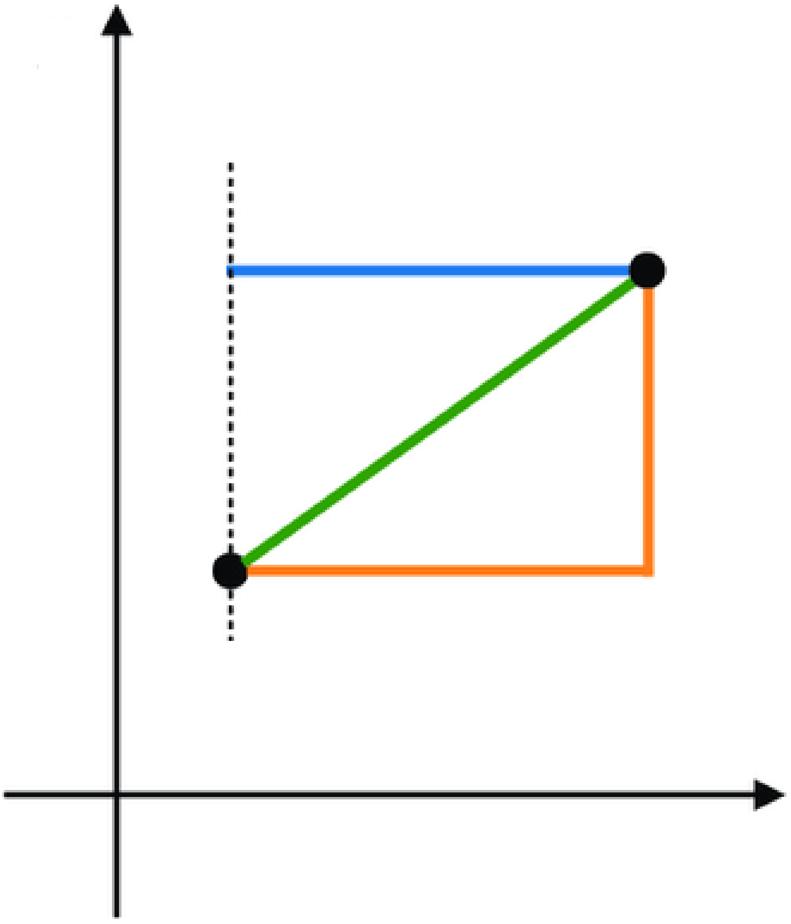
$$L_\infty(\mathbf{x},\mathbf{y}) = \max_{\boldsymbol{i}} |x_i - y_i|$$

K-近邻算法

K近邻算法中的距离度量

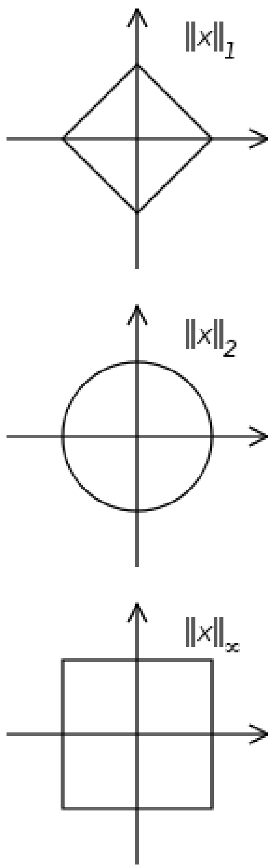
- L_1 距离： **曼哈顿距离**
- L_2 距离 ： **欧式距离**
- L_∞ 距离： **契比雪夫距离**

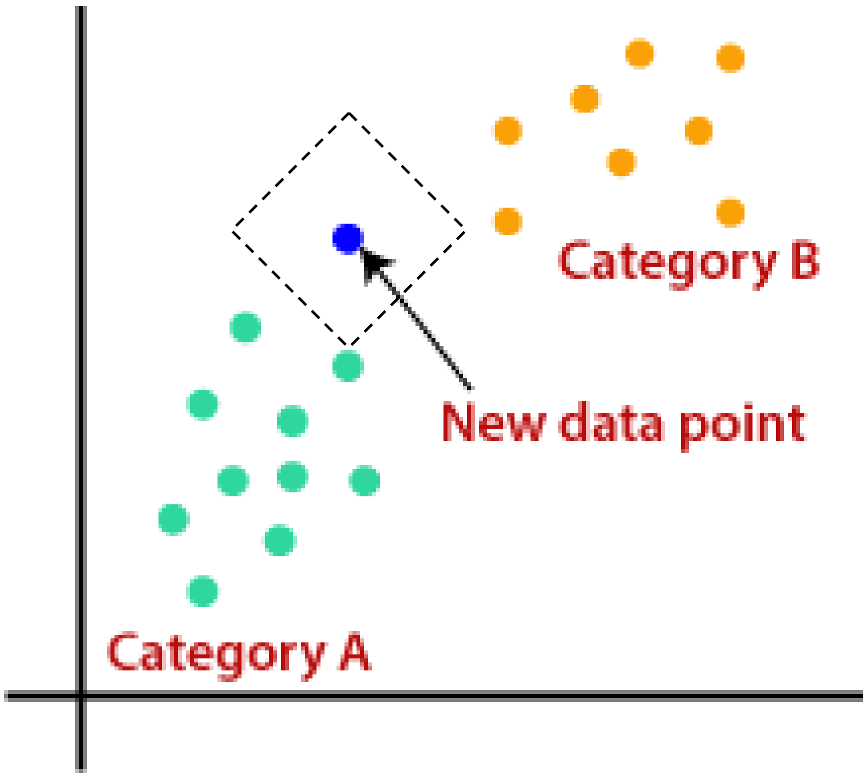




K-近邻算法

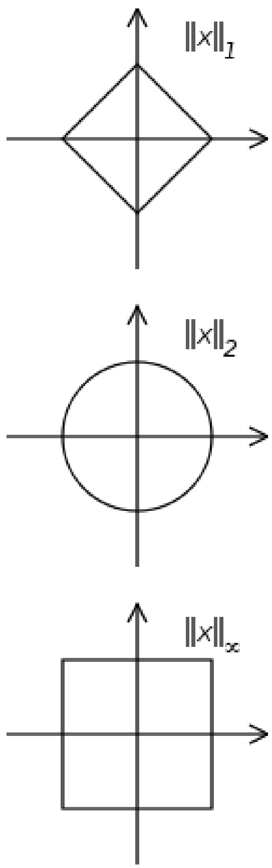
K近邻算法中的距离度量的影响

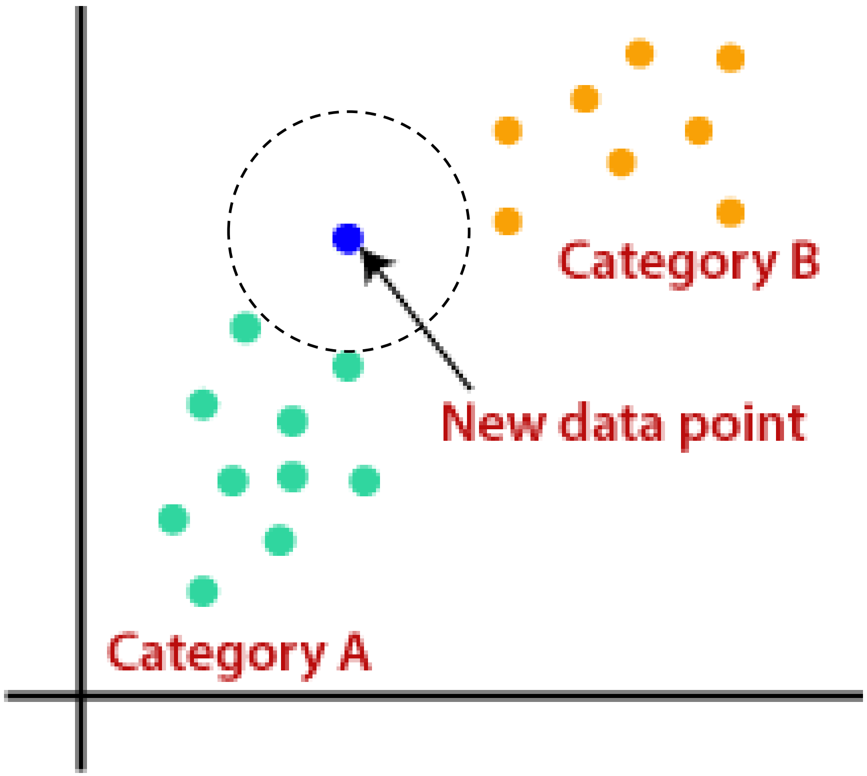




K-近邻算法

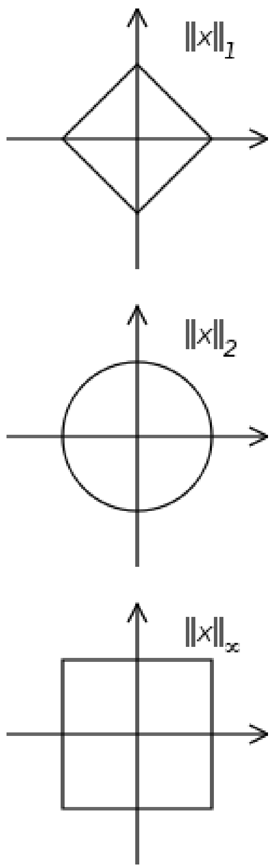
K近邻算法中的距离度量的影响

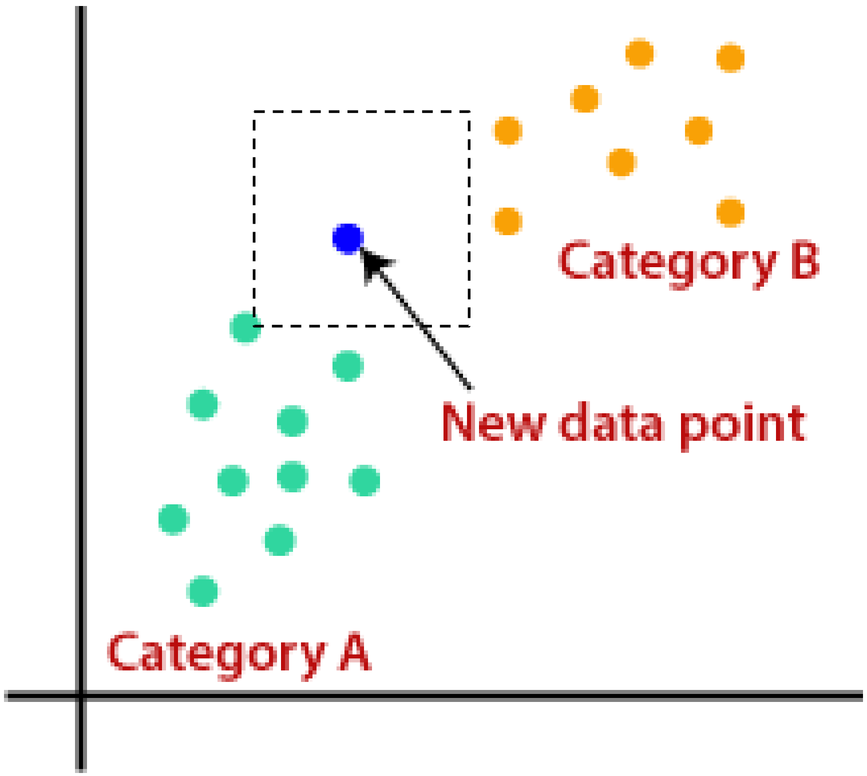




K-近邻算法

K近邻算法中的距离度量的影响





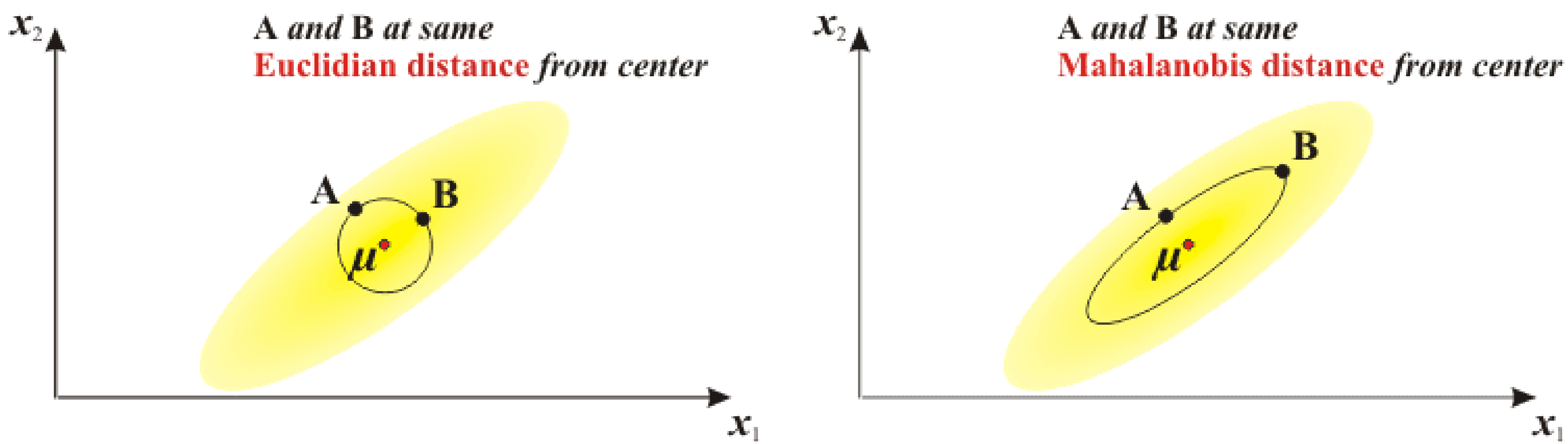
K-近邻算法

K近邻算法中的距离度量

- **马氏距离（Mahalanobis distances）**：

$$D(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mathbf{y})}$$

  - 依赖数据整体分布
  - 与度量无关，进一步，对任意线性变换具有不变性



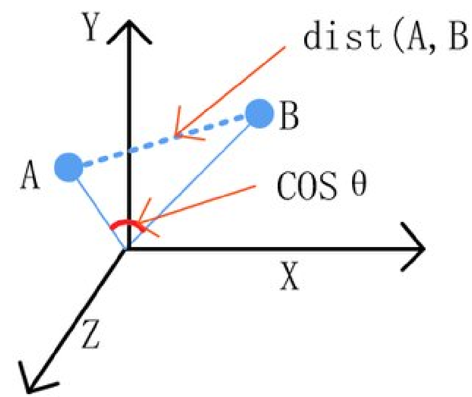
K-近邻算法

K近邻算法中的距离度量

- **余弦相似度（cosine similarity）：**

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

  - 与数据尺度无关，映射到球面上度量夹角



- **编辑距离（Levenshtein Edit Distance）**

$$\operatorname{lev}(a,b) = \begin{cases} |a| & \text{if } |b| = 0, \\ |b| & \text{if } |a| = 0, \\ \operatorname{lev}\big(\operatorname{tail}(a), \operatorname{tail}(b)\big) & \text{if } a[0] = b[0] \\ 1 + \min \begin{cases} \operatorname{lev}\big(\operatorname{tail}(a), b\big) \\ \operatorname{lev}\big(a, \operatorname{tail}(b)\big) \\ \operatorname{lev}\big(\operatorname{tail}(a), \operatorname{tail}(b)\big) \end{cases} & \text{otherwise,} \end{cases}$$

		H	Y	U	N	D	A	I
	0	1	2	3	4	5	6	7
H	1	0	1	2	3	4	5	6
O	2	1	1	2	3	4	5	6
N	3	2	2	2	2	3	4	5
D	4	3	3	3	3	2	3	4
A	5	4	4	4	4	3	2	3

K-近邻算法

K近邻算法特点

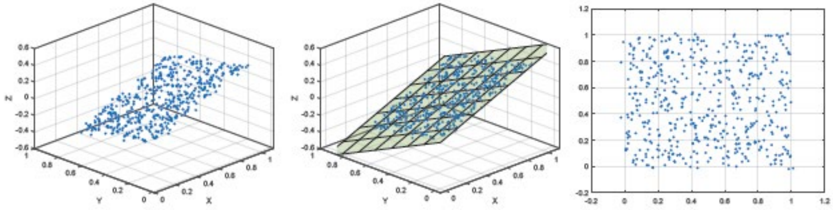
- k近邻学习没有显式的训练过程，属于“懒惰学习”。此类学习技术在训练阶段仅仅是把样本保存起来，训练时间开销为零，待收到测试样本后再进行处理。
- 存储所有训练样本，存储开销大
- 需要搜索K近邻，计算开销大

K-近邻算法

K近邻算法加速

- 降低样本维度：选择法，PCA

$$\mathcal{O}(dnK) \Longrightarrow \mathcal{O}(rnK)$$



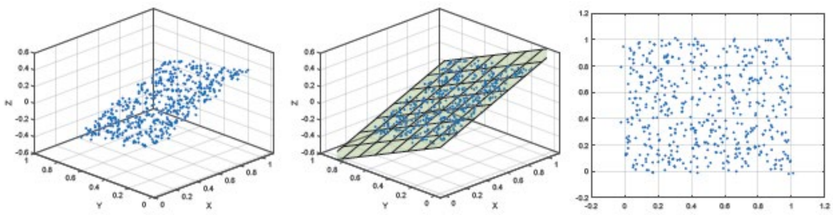
(a) (b) (c)

K-近邻算法

K近邻算法加速

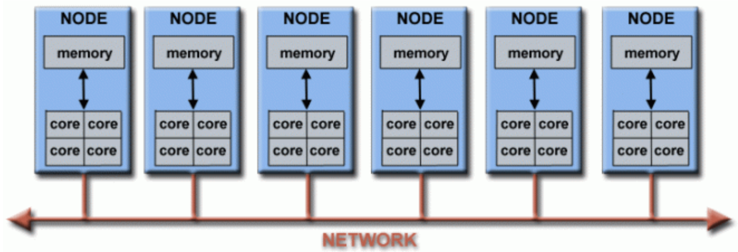
- 降低样本维度：选择法，PCA

$$\mathcal{O}(dnK) \Longrightarrow \mathcal{O}(rnK)$$


(a) (b) (c)

- 并行加速计算：分布式，GPU

$$\mathcal{O}(dnK) \Longrightarrow \mathcal{O}\left(\frac{dnK}{L}\right)$$

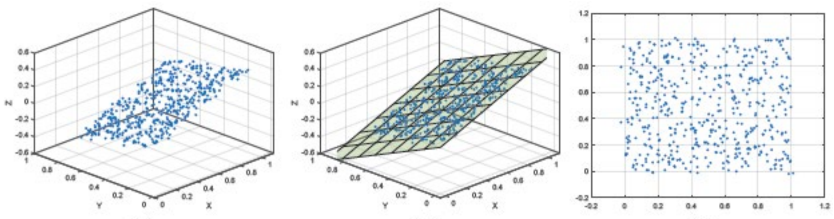


K-近邻算法

K近邻算法加速

- 降低样本维度：选择法，PCA

$$\mathcal{O}(dnK) \Longrightarrow \mathcal{O}(rnK)$$



(a) (b) (c)

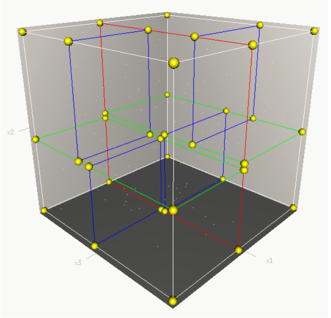
- 并行加速计算：分布式，GPU

$$\mathcal{O}(dnK) \Longrightarrow \mathcal{O}\left(\frac{dnK}{L}\right)$$

$$\mathcal{O}(dnK) \Rightarrow \mathcal{O}(d \log n\, K)$$

- 建立索引结构：KD树，Quad树

$$\mathcal{O}(dnK) \Longrightarrow \mathcal{O}(d \log n\, K)$$

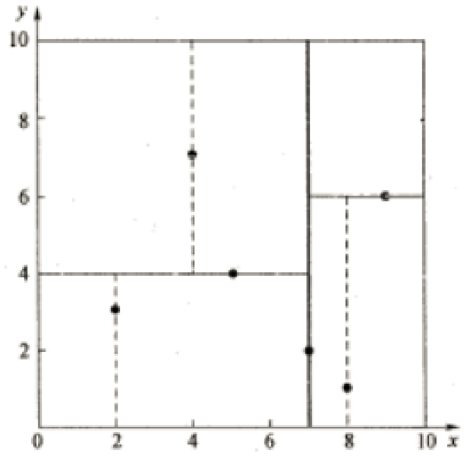


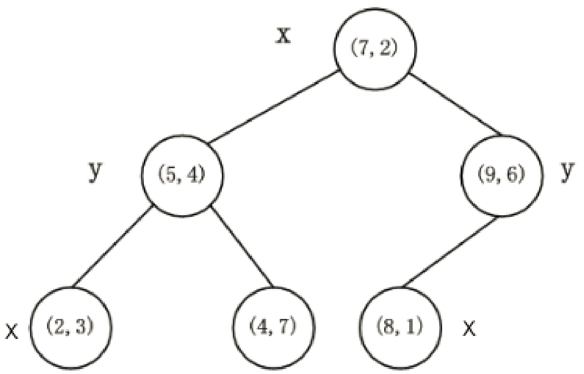
K-近邻算法

K近邻算法加速

- KD树构建

$$\begin{Bmatrix} (2,3), (5,4), \\ (9,6), (4,7), \\ (8,1), (7,2) \end{Bmatrix}$$



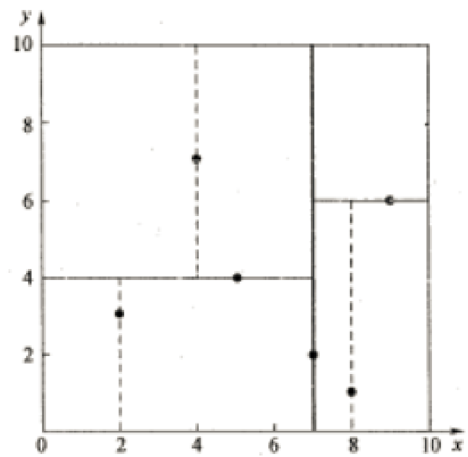


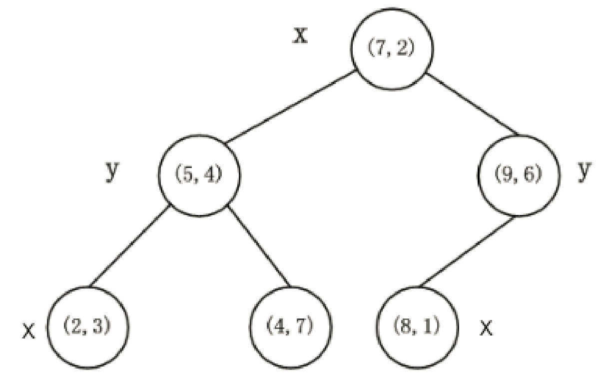
K-近邻算法

K近邻算法加速

- KD树构建

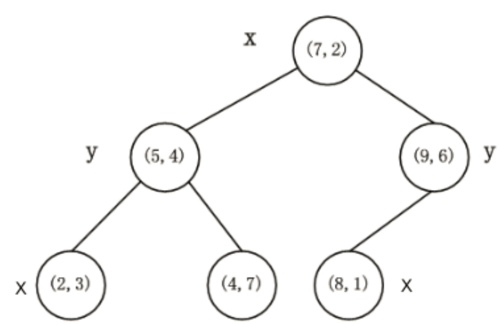
$$\left\{\begin{matrix}(2,3),(5,4),\\(9,6),(4,7),\\(8,1),(7,2)\end{matrix}\right\}$$





- 近邻搜索

搜索以 (2.1,3.1)为中心的最近邻

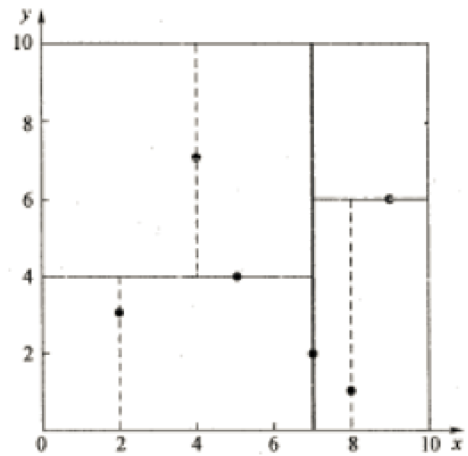


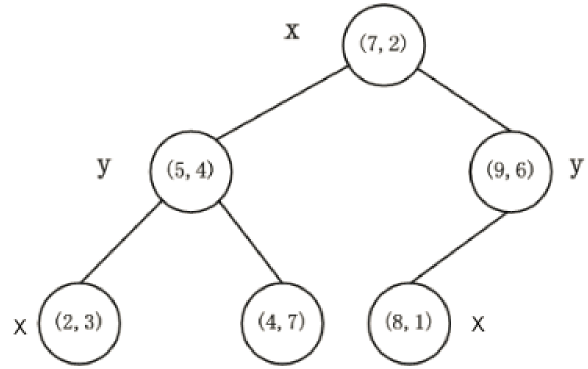
K-近邻算法

K近邻算法加速

- KD树构建

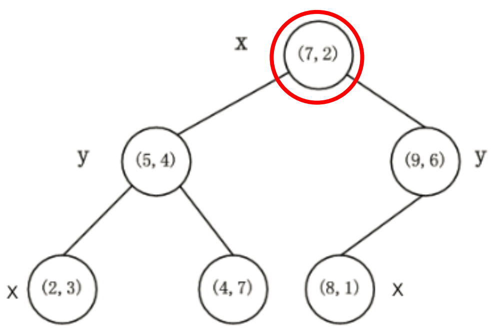
$$\left\{\begin{matrix}(2,3),(5,4),\\(9,6),(4,7),\\(8,1),(7,2)\end{matrix}\right\}$$

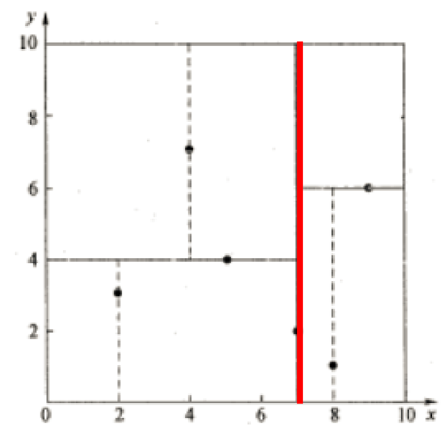




- 近邻搜索

搜索以 (2.1,3.1)为中心的最近邻



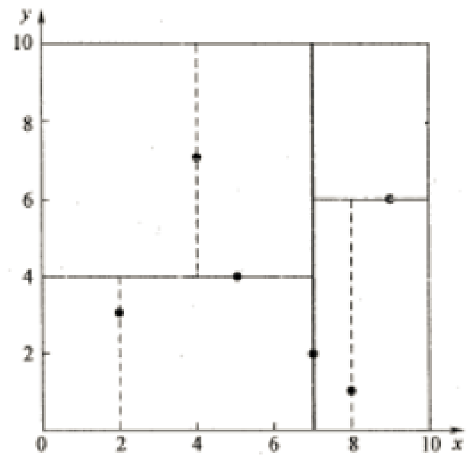


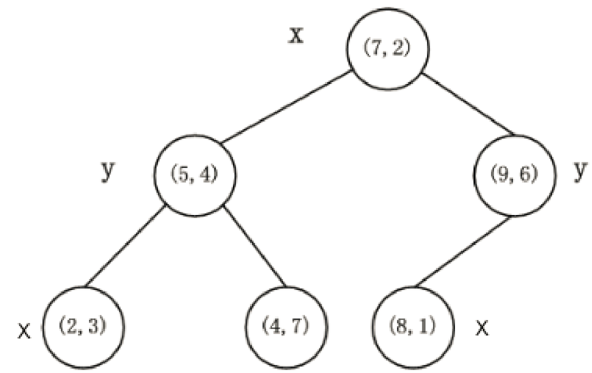
K-近邻算法

K近邻算法加速

- KD树构建

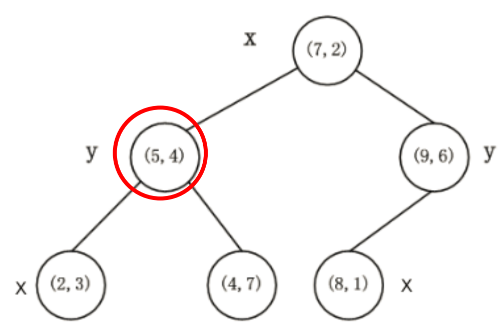
$$\left\{\begin{matrix}(2,3),(5,4),\\(9,6),(4,7),\\(8,1),(7,2)\end{matrix}\right\}$$

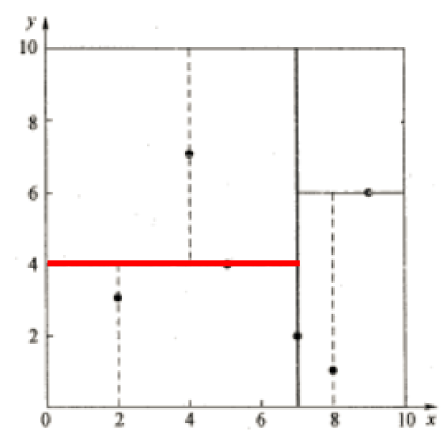




- 近邻搜索

搜索以 (2.1,3.1)为中心的最近邻



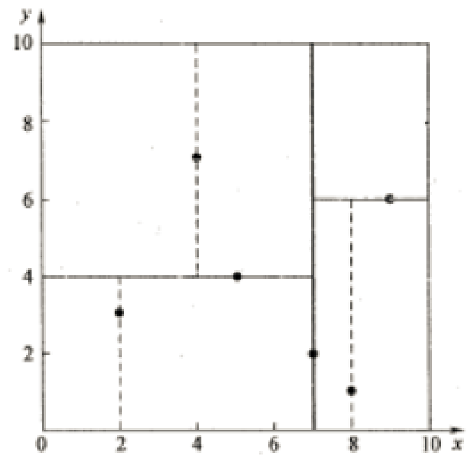


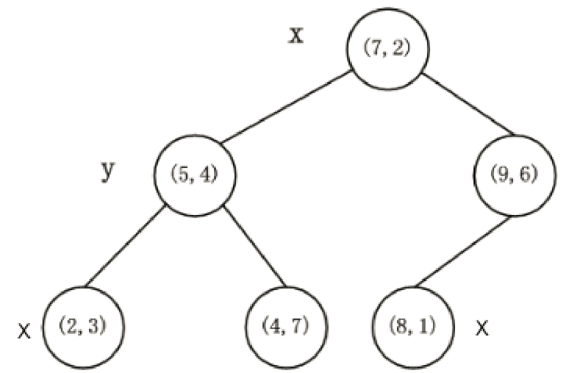
K-近邻算法

K近邻算法加速

◆ KD树构建

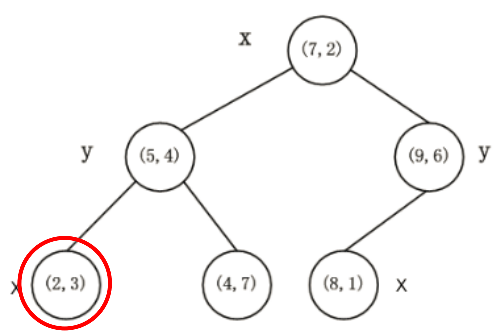
$$\left\{\begin{matrix}(2,3),(5,4),\\(9,6),(4,7),\\(8,1),(7,2)\end{matrix}\right\}$$

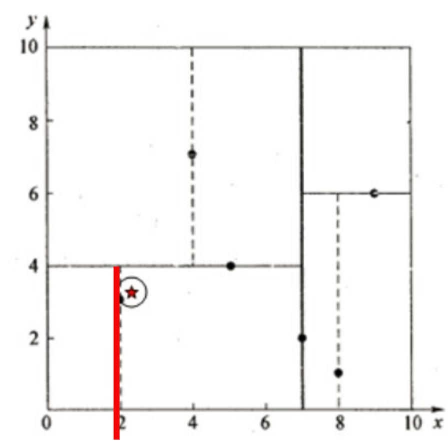




◆ 近邻搜索

搜索以 (2.1,3.1)为中心的最近邻



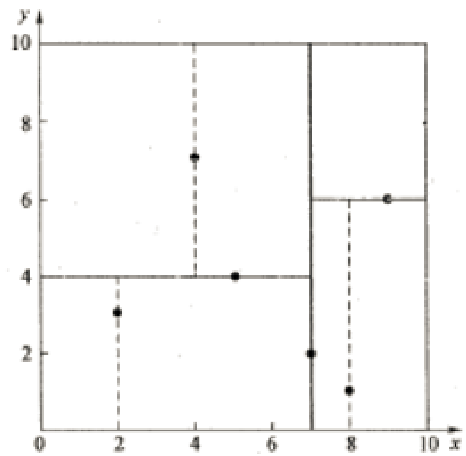


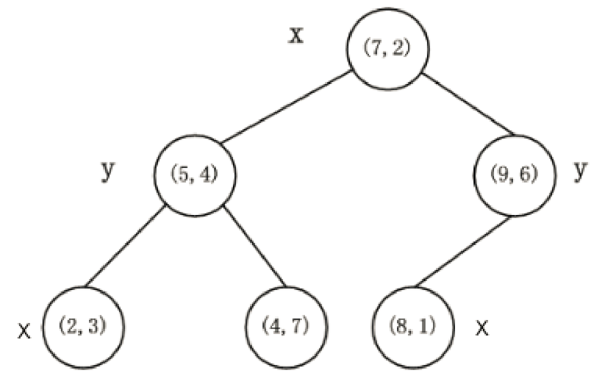
K-近邻算法

K近邻算法加速

- KD树构建

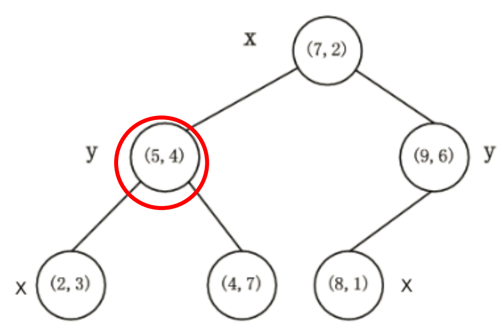
$$\left\{\begin{array}{l}(2,3),(5,4),\\(9,6),(4,7),\\(8,1),(7,2)\end{array}\right\}$$

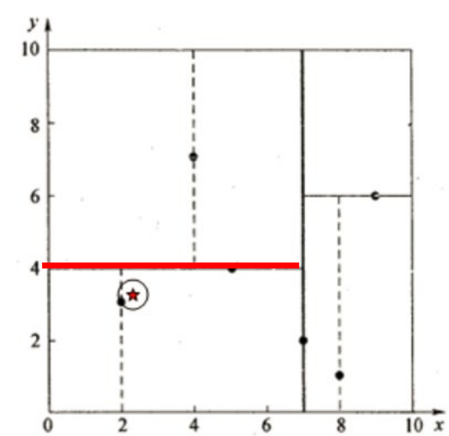




- 近邻搜索

搜索以 (2.1,3.1)为中心的最近邻



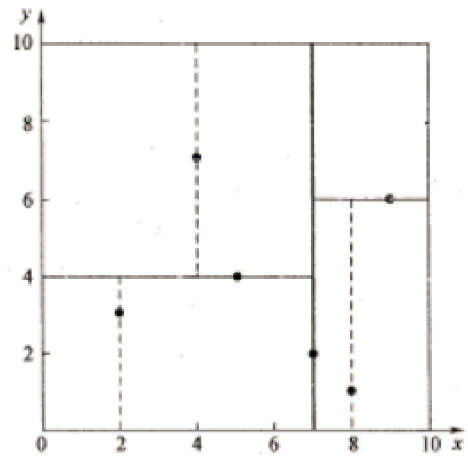
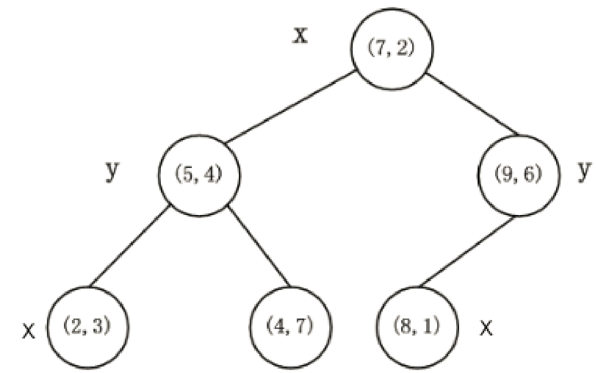


K-近邻算法

K近邻算法加速

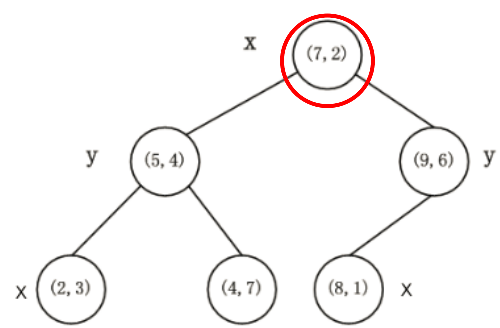
- KD树构建

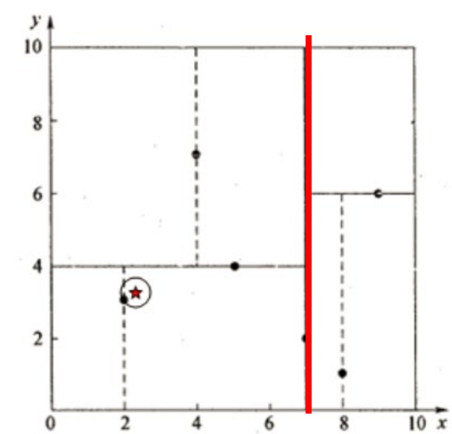
$$\begin{Bmatrix}(2,3),(5,4),\\(9,6),(4,7),\\(8,1),(7,2)\end{Bmatrix}$$

- 近邻搜索

搜索以 (2.1,3.1)为中心的最近邻





内容概要

- 机器学习基础回顾
- K-近邻算法
- **基础回归算法**
- 总结

基础回归算法

回归分析起源

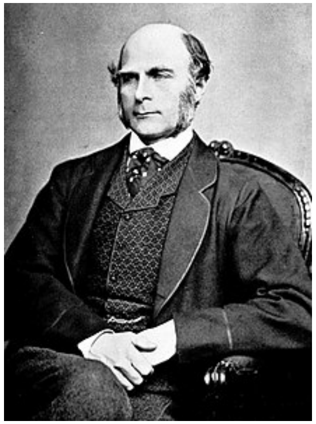
- 父母与子女身高

$$y = 33.73(\text{英寸}) + 0.516x$$

y: 子女平均身高

x:父母平均身高

- 父母平均身高每增加一个单位，其成年子女平均身高只增加0.516个单位，它反映了这种“衰退(regression)”效应（“回归”到正常人平均身高）。
- 虽然x和y之间并不总是具有“衰退”（回归）关系，但是“线性回归”这一名称就保留了下来了。



英国著名生物学家兼统计学家高尔顿
Sir Francis Galton
(1822-1911)

基础回归算法

模型抽象

该回归模型中两个参数

需要从标注数据
中学习得到
(监督学习)

$$y = 33.73(\text{英寸}) + 0.516x$$

y: 子女平均身高

x:父母平均身高

- 给出任意一对父母平均身高，则可根据上述方程，计算得到其子女平均身高
- 从父母平均身高来**预测**其子女平均身高
- 如何求取上述线性方程（预测方程）的参数?

基础回归算法

一元线性回归

标注数据

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

基础回归算法

一元线性回归

标注数据

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

学习模型

$$y = ax + b$$

基础回归算法

一元线性回归

标注数据

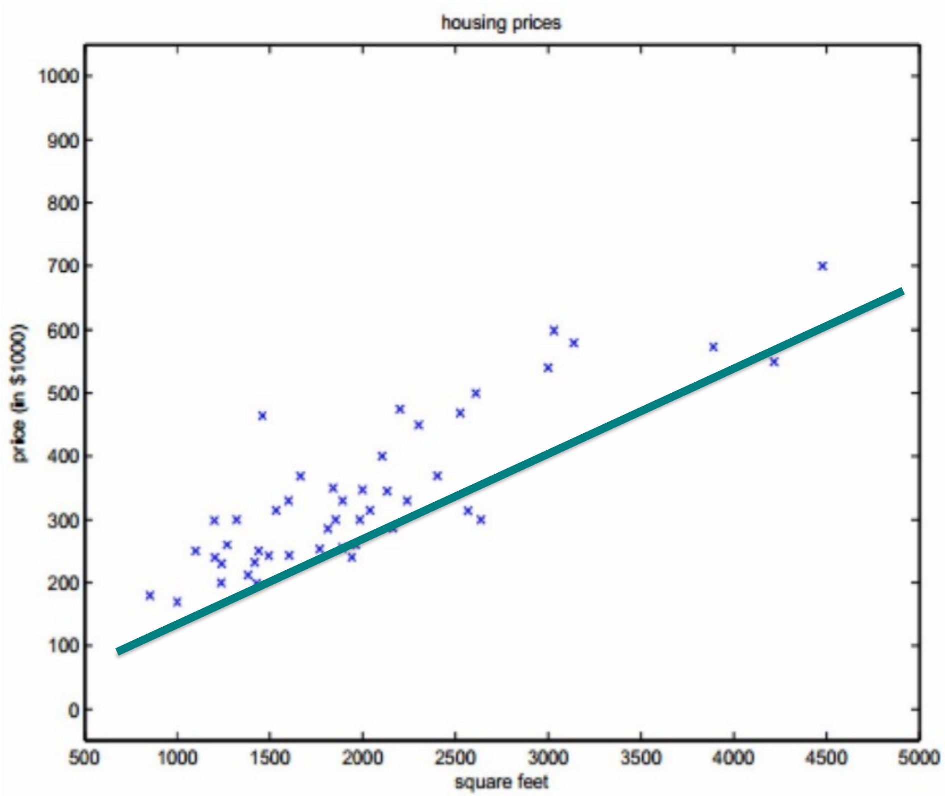
$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$$

学习模型

$$y = ax + b$$

直观动机

找一条直线是的训练集中所有的点都距其“很近”



基础回归算法

一元线性回归

损失函数

$$L(a,b)=\frac{1}{N}\sum_{i=1}^{N}(ax_i+b-y_i)^2$$

$$=\frac{1}{N}\sum_{i=1}^{N}a^2x_i^2+2abx_i-2ax_iy_i+b^2-2by_i+y_i^2$$

$$=\frac{1}{N}\left(a^2\sum_{i=1}^{N}x_i^2+2ab\sum_{i=1}^{N}x_i-2a\sum_{i=1}^{N}x_iy_i+Nb^2-2b\sum_{i=1}^{N}y_i+\sum_{i=1}^{N}y_i^2\right)$$

$$=\frac{1}{N}(a^2C_1+2abC_2-2aC_3+Nb^2-2bC_4+C_5)$$

基础回归算法

一元线性回归

损失函数

$$L(a,b)=\frac{1}{N}(a^2C_1+2abC_2-2aC_3+b^2N-2bC_4+C_5)$$

$C_1=\sum_{i=1}^{N}x_i^2$

$C_2=\sum_{i=1}^{N}x_i$

$C_3=\sum_{i=1}^{N}x_iy_i$

$C_4=\sum_{i=1}^{N}y_i$

$C_5=\sum_{i=1}^{N}y_i^2$

优化算法

$$\frac{\partial}{\partial a}L(a,b)=\frac{1}{N}(2aC_1+2bC_2-2C_3)=0$$

$$\frac{\partial}{\partial b}L(a,b)=\frac{1}{N}(2aC_2+2bN-2C_4)=0$$

$$\begin{cases}aC_1+bC_2=C_3\\ aC_2+bN=C_4\end{cases}$$

$$a^*=\frac{C_2C_4-NC_3}{C_2^2-NC_1}$$

$$b^*=\frac{C_2C_3-C_1C_4}{C_2^2-NC_1}$$

基础回归算法

一元线性回归

优化算法

$$L(a,b)=\frac{1}{N}(a^2C_1+2abC_2-2aC_3+b^2N-2bC_4+C_5)$$

梯度下降法

$$L(a,b)=L(a_0,b_0)+\frac{\partial L}{\partial a_0}(a-a_0)+\frac{\partial L}{\partial b_0}(b-b_0)+o\left(\sqrt{(a-a_0)^2+(b-b_0)^2}\right)$$

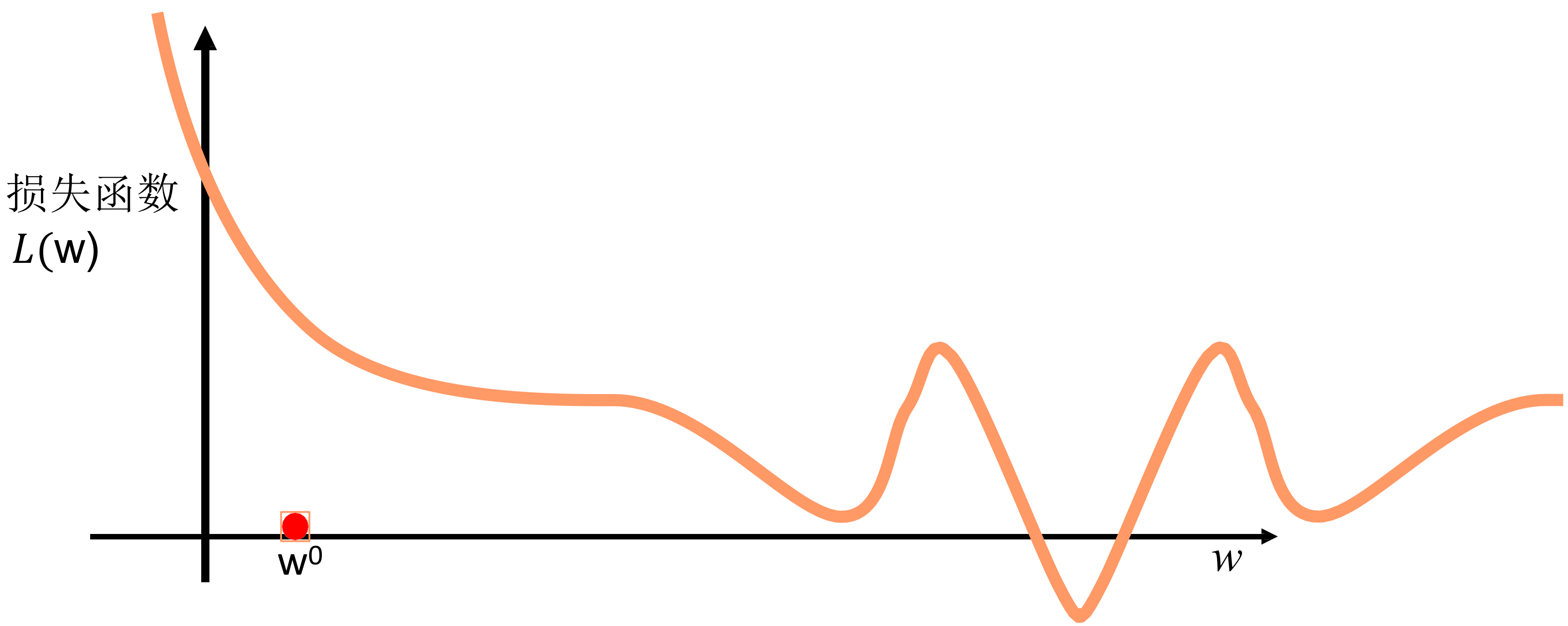
$$\nabla_{(a_0,b_0)}=\left(\frac{\partial L}{\partial a_0},\frac{\partial L}{\partial b_0}\right)$$

在(a_0,b_0)的一个小的局部区域内，沿着$-\nabla_{(a_0,b_0)}$（负梯度方向）移动一定会使得函数值下降

基础回归算法

一元线性回归

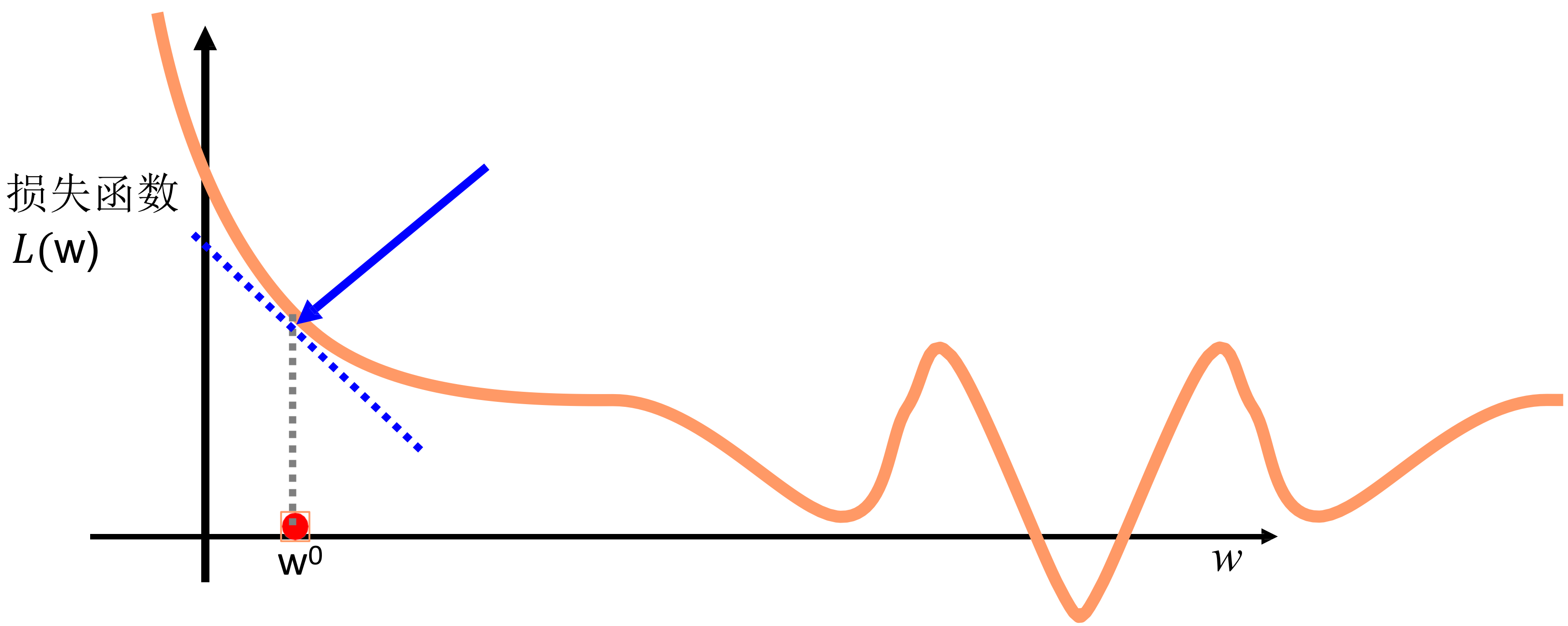
优化算法



基础回归算法

一元线性回归

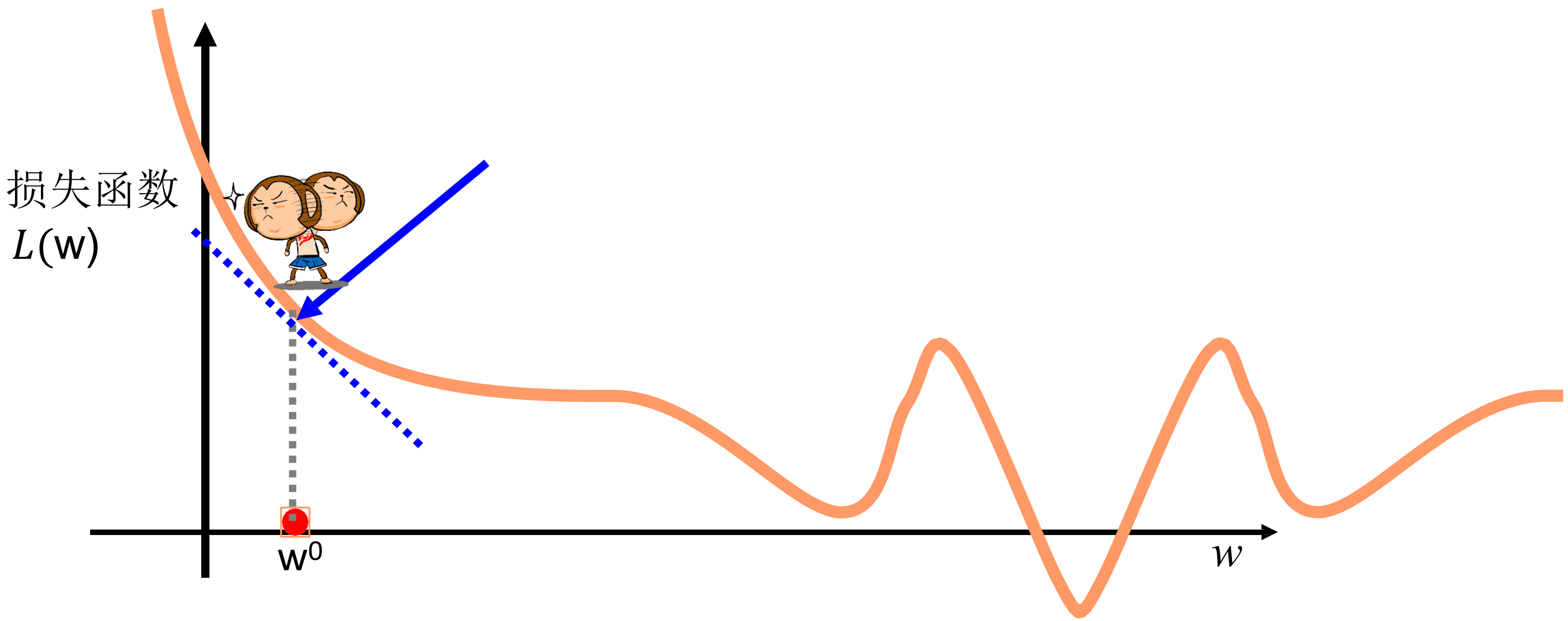
优化算法



基础回归算法

一元线性回归

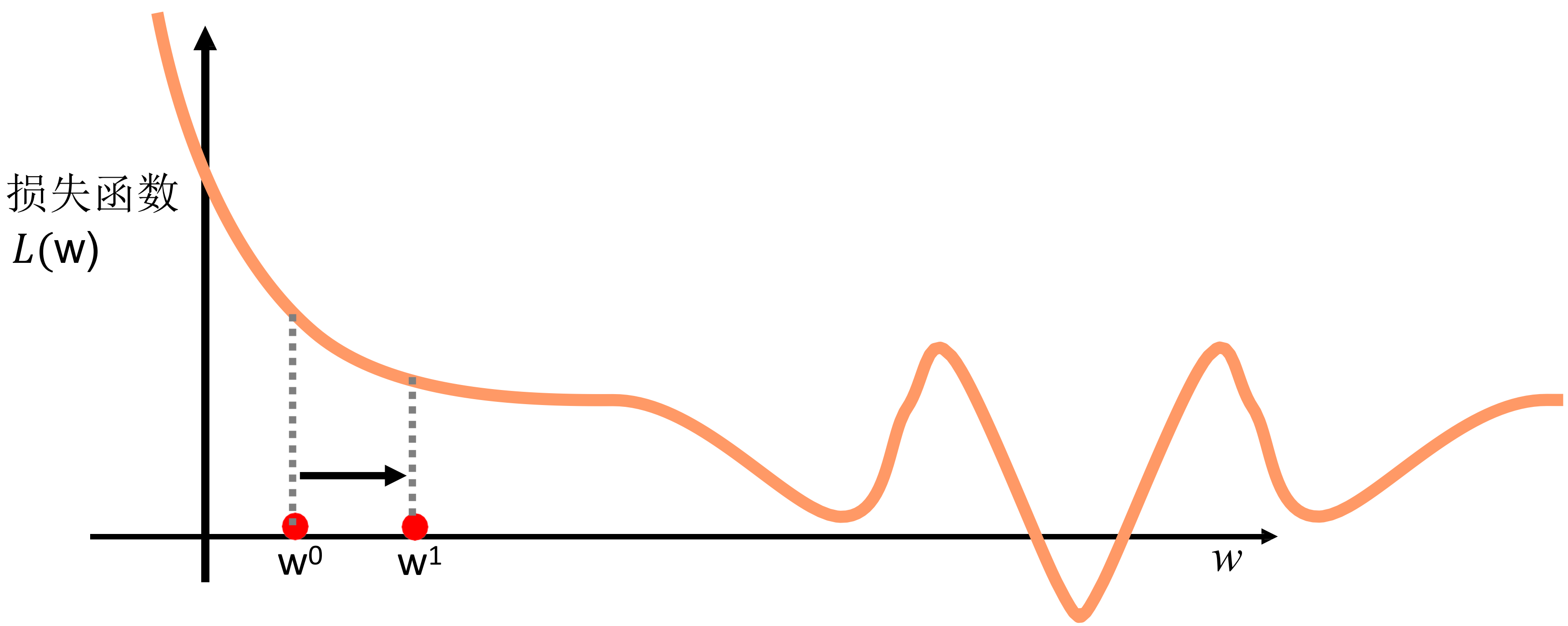
优化算法



基础回归算法

一元线性回归

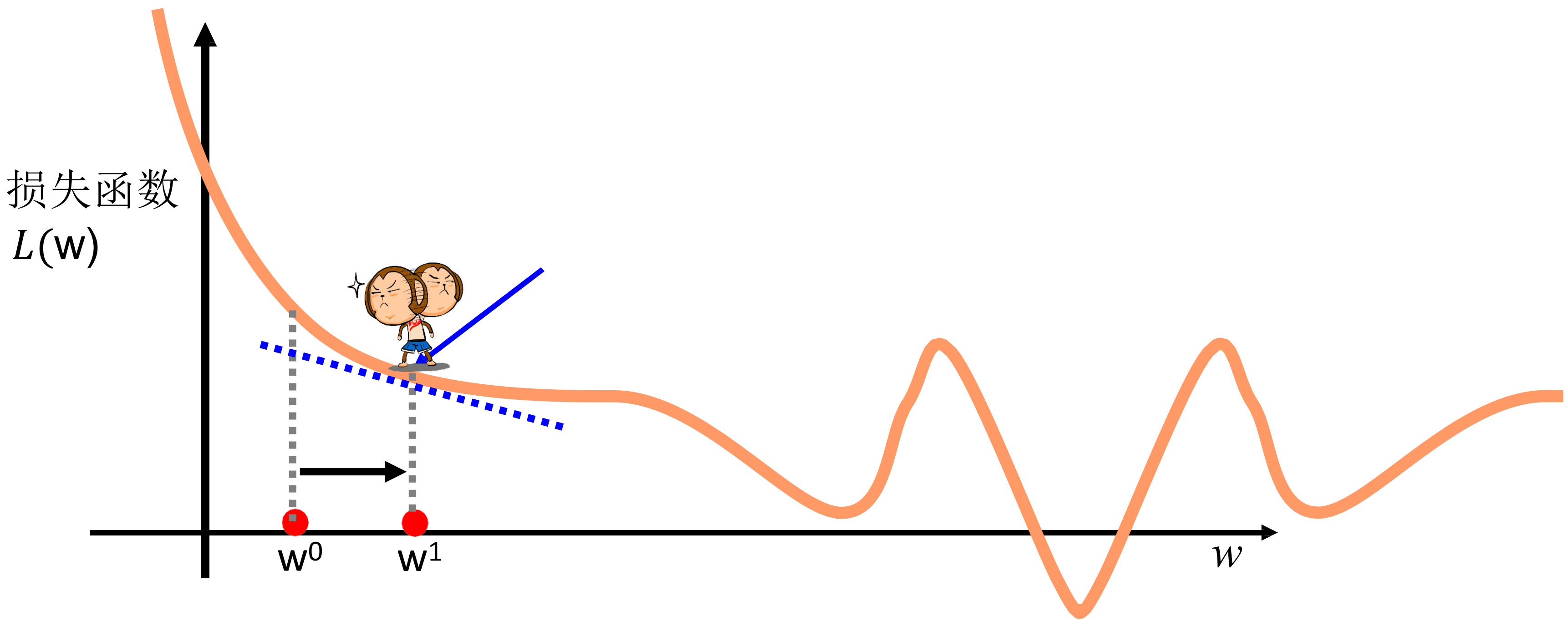
优化算法



基础回归算法

一元线性回归

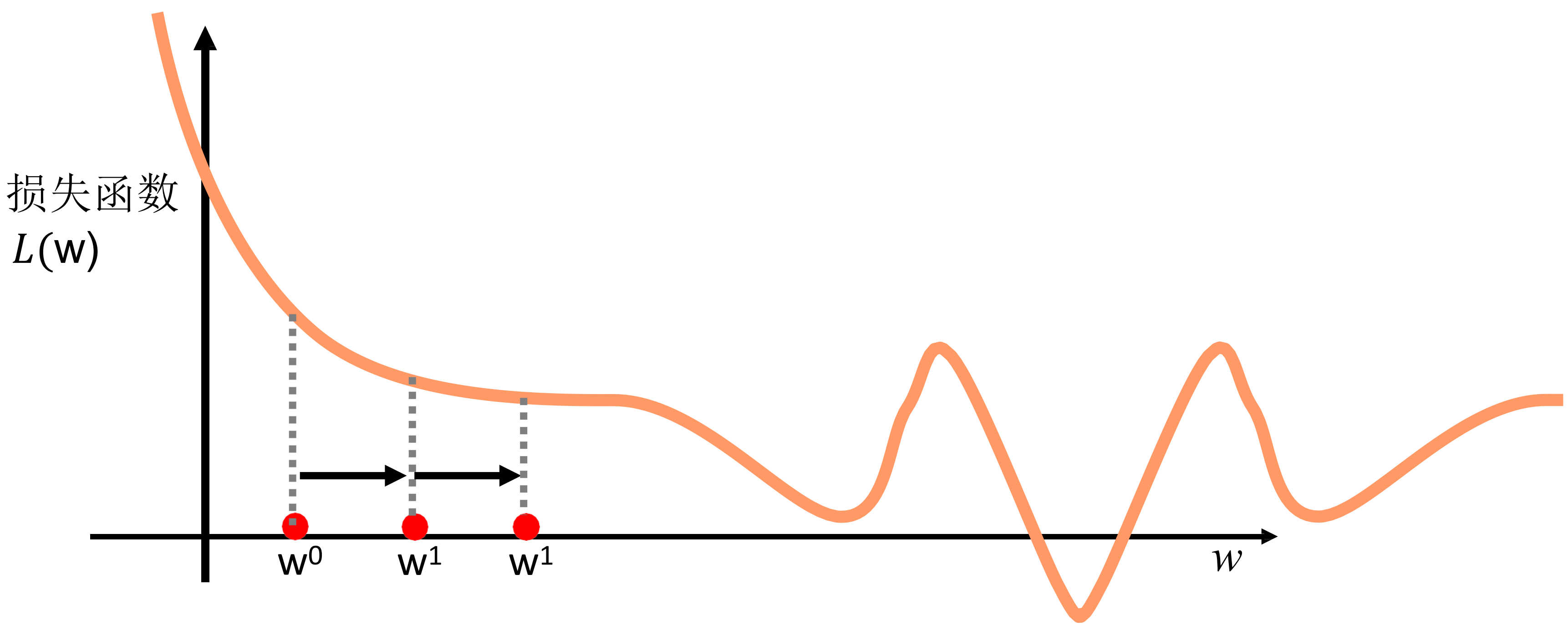
优化算法



基础回归算法

一元线性回归

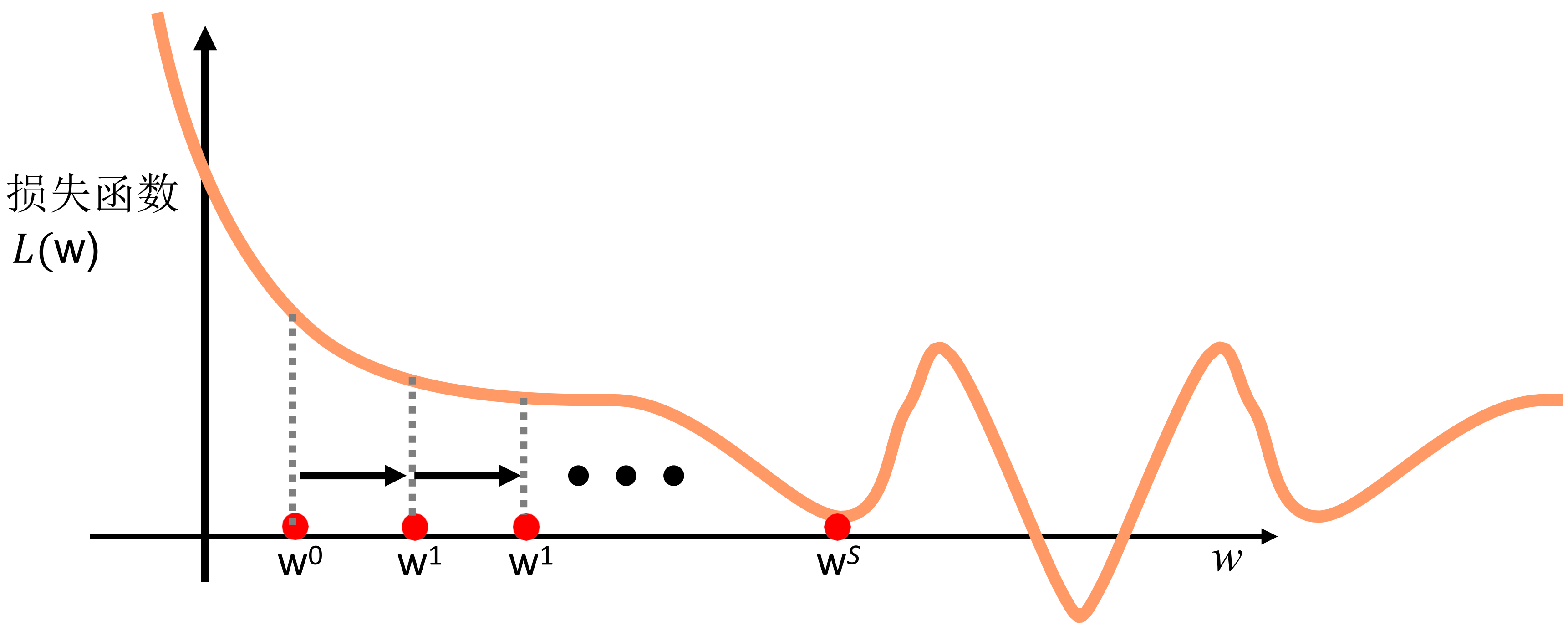
优化算法



基础回归算法

一元线性回归

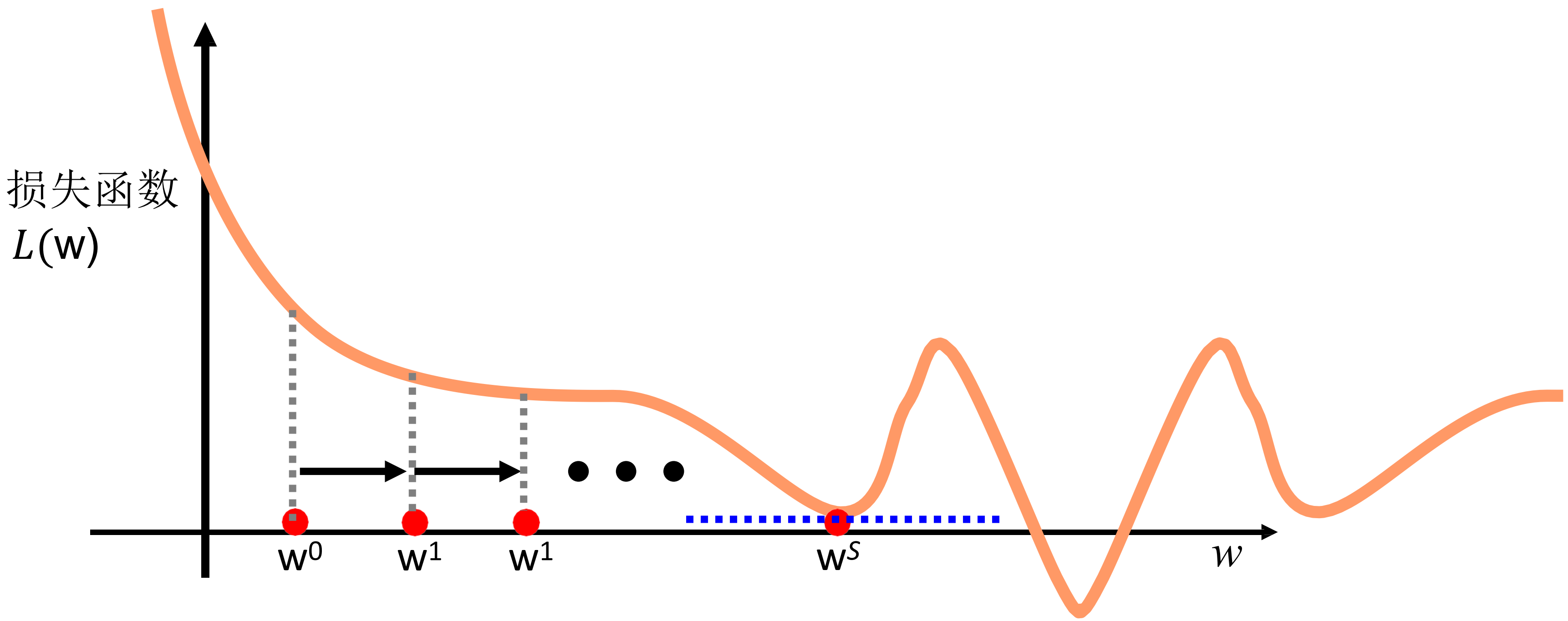
优化算法



基础回归算法

一元线性回归

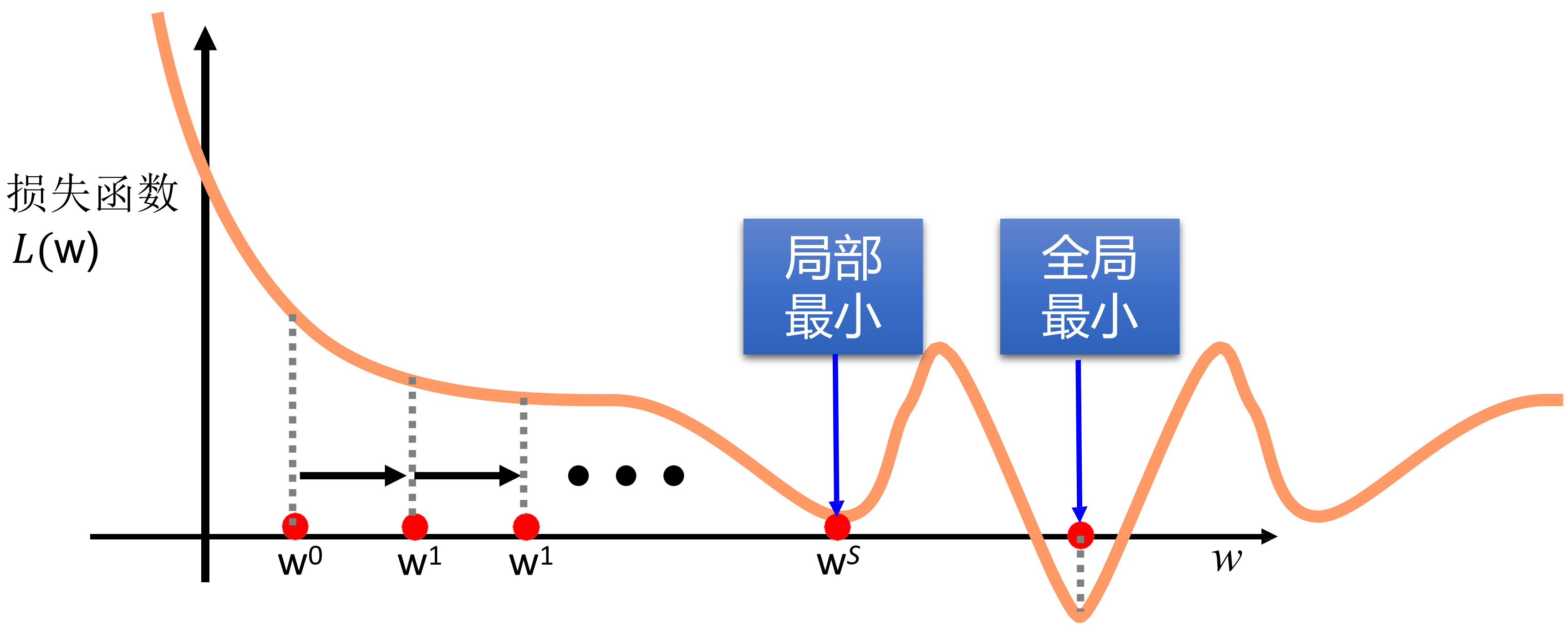
优化算法



基础回归算法

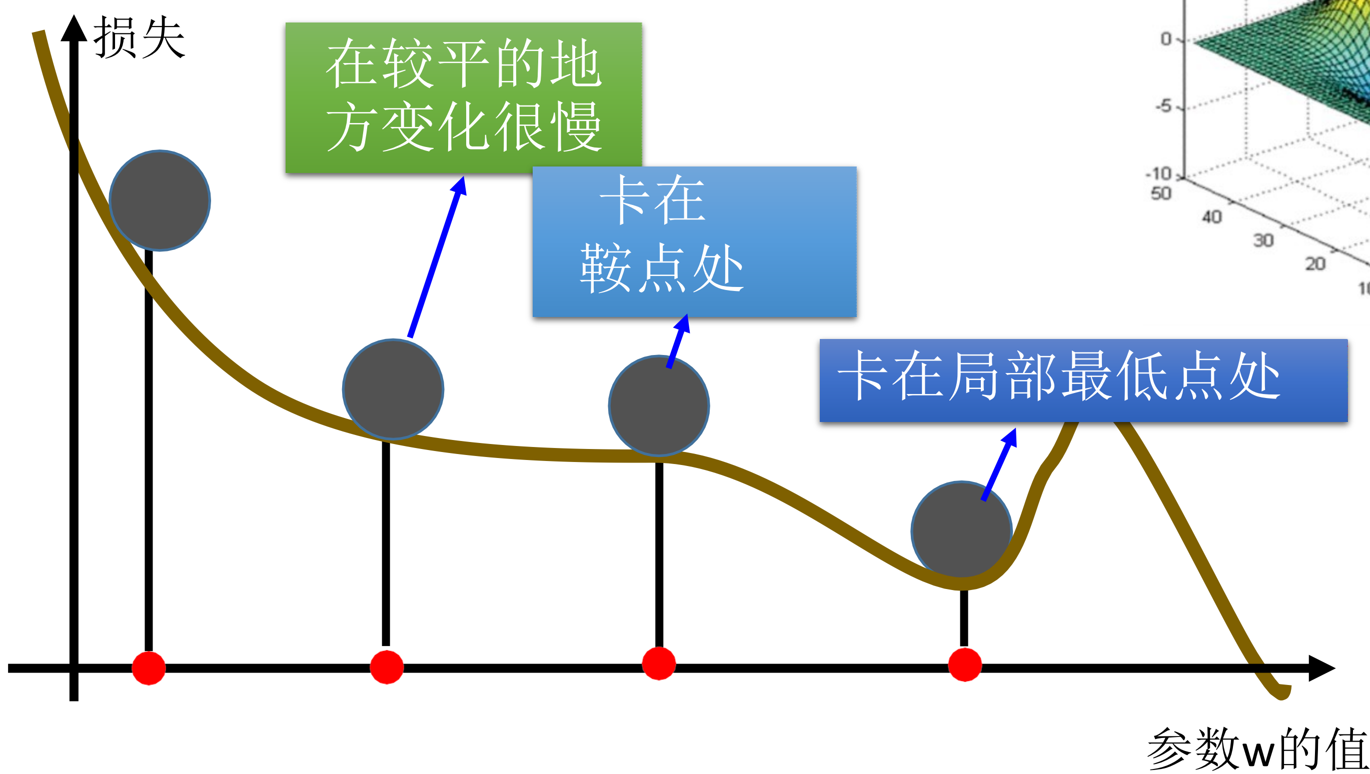
一元线性回归

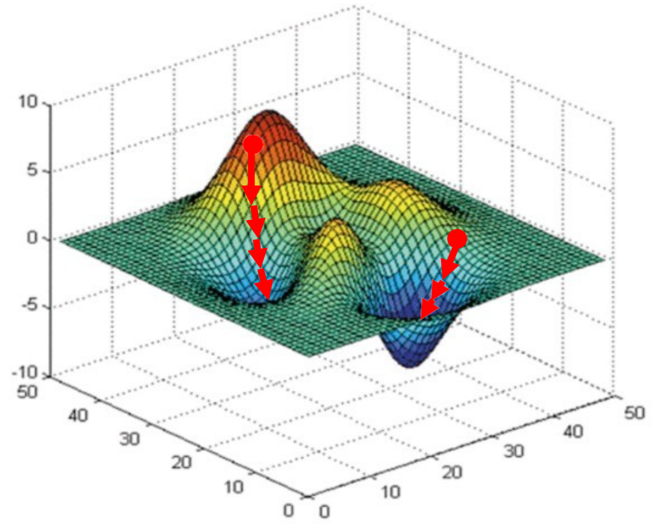
优化算法



基础回归算法

梯度下降可能的问题





基础回归算法

多元线性回归

面积 ($feet^2$)	卧室数目（个）	层数（个）	房子的年龄（年）	房价 (\$1000)
2104	5	2	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

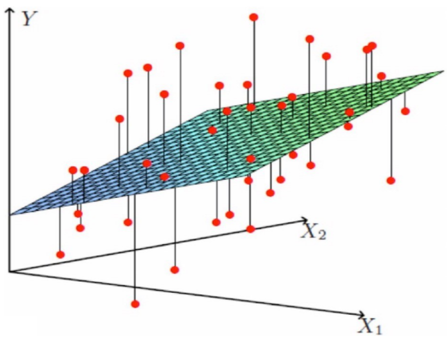
面积 ($feet^2$)	卧室个数	层数	房子年龄	价格 (\$1000)
1500	3	2	30	?

基础回归算法

多元线性回归

标注数据

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$$



学习模型

$$y = a^1x^1 + a^2x^2 + \cdots a^Dx^D + b = \mathbf{a}^T\mathbf{x} + b$$

$$\mathbf{a} = (a^1, a^2, \ldots, a^D)^T \in \mathbb{R}^D \qquad \mathbf{x} = (x^1, x^2, \ldots, x^D)^T \in \mathbb{R}^D$$

损失函数

$$L(\mathbf{a}, b) = \sum_{i=1}^{N} (\mathbf{a}^T\mathbf{x}_i + b - y_i)^2$$

基础回归算法

多元线性回归

优化算法

$$L(\mathbf{a}, b) = \sum_{i=1}^{N} (\mathbf{a}^T \mathbf{x}_i + b - y_i)^2$$

$$\frac{\partial}{\partial \mathbf{a}} L(\mathbf{a}, b) = \sum_{i=1}^{N} 2(\mathbf{a}^T \mathbf{x}_i + b - y_i)\mathbf{x}_i = 2\left(\sum_{i=1}^{N} \mathbf{x}_i \mathbf{a}^T \mathbf{x}_i + \mathbf{x}_i b - \mathbf{x}_i y_i\right)$$

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$$

$$\mathbf{y} = [y_1, y_2, \dots, y_N] \in \mathbb{R}^N$$

$$= 2\big(\mathbf{X}\mathbf{X}^T \mathbf{a} + \mathbf{X}(\mathbf{b} - \mathbf{y})\big) = 0$$

$$\mathbf{X}\mathbf{X}^T \mathbf{a} = \mathbf{X}(\mathbf{y} - \mathbf{b})$$

基础回归算法

多元线性回归

优化算法

$$\frac{\partial}{\partial b} L(\mathbf{a}, b) = \sum_{i=1}^{N} 2(\mathbf{a}^T \mathbf{x}_i + b - y_i) = 0$$

$$\Downarrow$$

$D+1$个未知数，$D+1$个方程

$$\begin{cases} b = \frac{1}{N} \sum_{i=1}^{N} y_i - \mathbf{a}^T \mathbf{x}_i \\ \mathbf{X}\mathbf{X}^T \mathbf{a} = \mathbf{X}(\mathbf{y} - \mathbf{b}) \end{cases}$$

基础回归算法

线性向量回归

标注数据

$$\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$$

学习模型

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$$

$$\mathbf{x} \in \mathbb{R}^D \qquad \mathbf{y} \in \mathbb{R}^M \qquad \mathbf{A} \in \mathbb{R}^{M \times D} \qquad \mathbf{b} \in \mathbb{R}^M$$

损失函数

$$L(\mathbf{A}, \mathbf{b}) = \sum_{i=1}^{N} \|\mathbf{A}\mathbf{x}_i + \mathbf{b} - \mathbf{y}_i\|^2$$

基础回归算法

线性向量回归

优化算法

$$L(\mathbf{A},\mathbf{b}) = \sum_{i=1}^{N} \|\mathbf{A}\mathbf{x}_i + \mathbf{b} - \mathbf{y}_i\|^2 = \sum_{i=1}^{N} \left\|\widetilde{\mathbf{A}}\tilde{\mathbf{x}}_i - \mathbf{y}_i\right\|^2$$

$$= \left\|\widetilde{\mathbf{A}}\widetilde{\mathbf{X}} - \mathbf{Y}\right\|_F^2$$

$$\widetilde{\mathbf{A}} = [\mathbf{A}\ \mathbf{b}] \in \mathbb{R}^{M\times(D+1)}$$

$$\tilde{\mathbf{x}} = [\mathbf{x}\ 1]^T \in \mathbb{R}^{D+1}$$

$$\widetilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_N] \in \mathbb{R}^{(D+1)\times N}$$

$$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N] \in \mathbb{R}^{M\times N}$$

$$\widetilde{\mathbf{A}}^* = \mathbf{Y}\widetilde{\mathbf{X}}^T\left(\widetilde{\mathbf{X}}\,\widetilde{\mathbf{X}}^T\right)^{-1} = \mathbf{Y}\widetilde{\mathbf{X}}^\dagger$$

$\widetilde{\mathbf{X}}^\dagger$ 为 $\widetilde{\mathbf{X}}$ 的伪逆（**Moore–Penrose逆**）

基础回归算法

一元多项式回归

标注数据

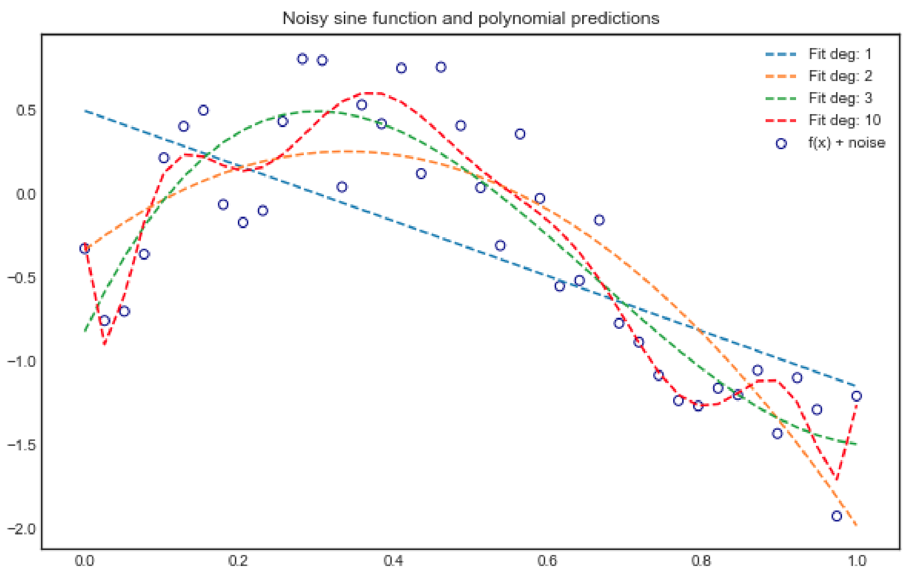
$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$$

学习模型

$$y = \sum_{r=1}^{R} a_r x^r + b = \sum_{r=0}^{R} a_r x^r$$

直观动机

找一条R阶多项式曲线使得的训练集中所有的点都距其“很近”



基础回归算法

一元多项式回归

损失函数

$$L(a_1, a_2, \dots, a_R) = \sum_{i=1}^{N}\left(\sum_{r=0}^{R} a_r x_i^r - y_i\right)^2$$

$$= \sum_{i=1}^{N}(\mathbf{a}^T\mathbf{x} - y_i)^2$$

$$\mathbf{a} = [a_0, a_1, \dots, a_R]^T \in \mathbb{R}^{R+1}$$

$$\mathbf{x} = [x^0, x^1, \dots, x^R]^T \in \mathbb{R}^{R+1}$$

多元线性回归，解之

基础回归算法

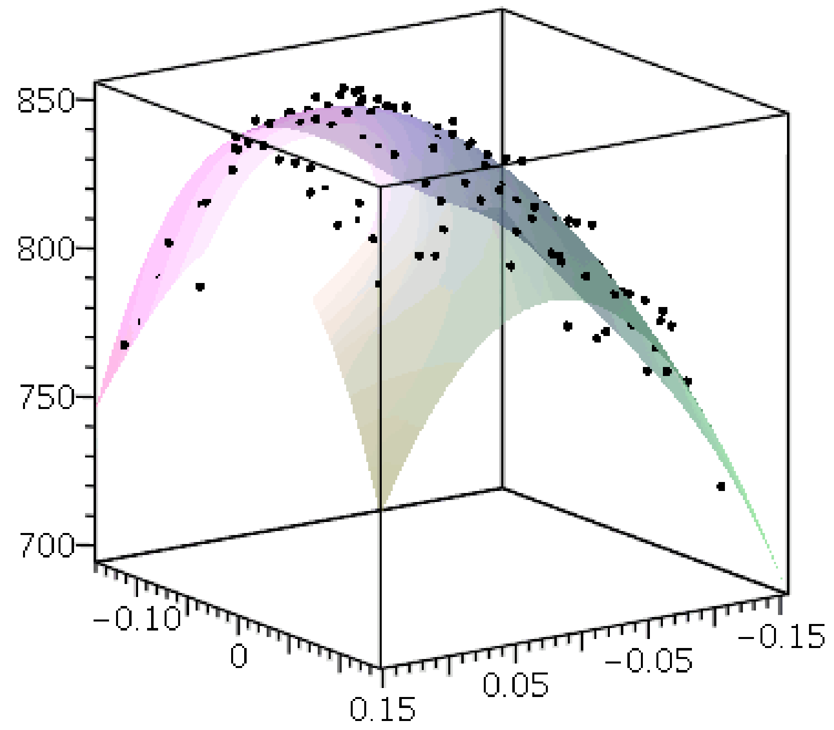
多元多项式回归

标注数据

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$$

学习模型

$$y = \sum_{d=1}^{D}\sum_{r=0}^{R} a_{dr} x_d^r$$



基础回归算法

多元多项式回归

损失函数

多元线性回归，解之

$$L(a_1, a_2, \dots, a_R) = \sum_{i=1}^{N}\left(\sum_{d=1}^{D}\sum_{r=0}^{R} a_{dr} x_{id}^{r} - y_i\right)^2 = \sum_{i=1}^{N}(\mathbf{a}^T\mathbf{x} - y_i)^2$$

$$\mathbf{a} = [a_{10}, a_{11}, \dots, a_{1R}, a_{20}, \dots, a_{D0}, a_{D1}, \dots, a_{DR}]^T \in \mathbb{R}^{D\times(R+1)}$$

$$\mathbf{x} = \left[x_1^0, x_1^1, \dots, x_1^R, x_2^0, \dots, x_D^0, x_D^1, \dots, x_D^R\right]^T \in \mathbb{R}^{D\times(R+1)}$$

基础回归算法

多项式向量回归

标注数据

$$\{(\mathbf{x}_1,\mathbf{y}_1),(\mathbf{x}_2,\mathbf{y}_2),\dots,(\mathbf{x}_N,\mathbf{y}_N)\}$$

学习模型

损失函数

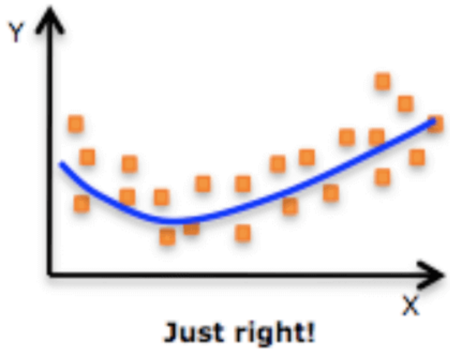
损失函数

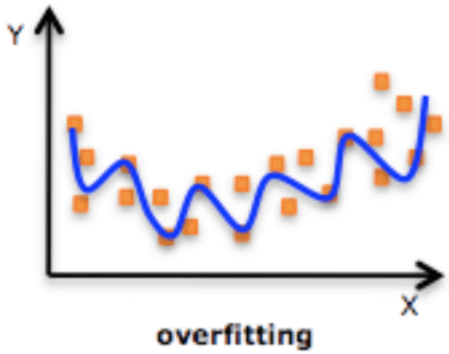
基础回归算法

带正则项的线性回归

$$L(\mathbf{a}) = \sum_{i=1}^{N} (\mathbf{a}\mathbf{x}_i - y_i)^2$$

过拟合问题



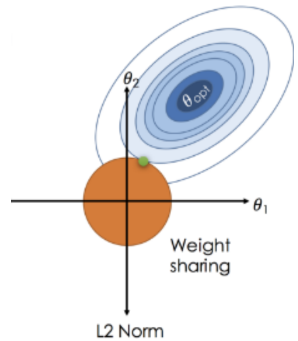


基础回归算法

带正则项的线性回归

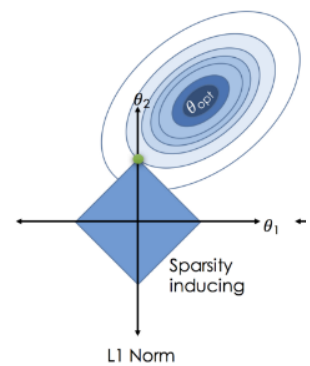
LASSO

$$L(\mathbf{a}) = \sum_{i=1}^{N} (\mathbf{a}\mathbf{x}_i - y_i)^2 + \lambda \|\mathbf{a}\|_2^2$$



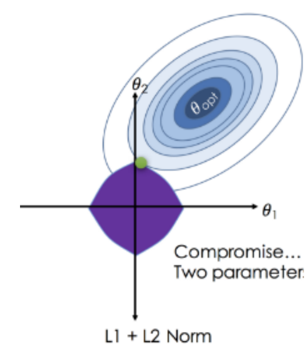
Ridge Regression

$$L(\mathbf{a}) = \sum_{i=1}^{N} (\mathbf{a}\mathbf{x}_i - y_i)^2 + \lambda \|\mathbf{a}\|_1$$



Elastic Net

$$L(\mathbf{a}) = \sum_{i=1}^{N} (\mathbf{a}\mathbf{x}_i - y_i)^2 + \lambda_1 \|\mathbf{a}\|_1 + \lambda_2 \|\mathbf{a}\|_2^2$$



内容概要

- 机器学习基础回顾
- K-近邻算法
- 基础回归算法
- **总结**

总结

1. K近邻算法

- ✓ 基本思想及步骤
- ✓ 参数K的影响
- ✓ 不同距离度量及其作用
- ✓ K近邻算法的加速：KD树

2. 基础回归算法

- ✓ 一元线性回归：思想、求解方法
- ✓ 多元，向量线性回归
- ✓ 一元多项式回归、多元多项式回归
- ✓ 带正则项的线性回归及其几何意义

北京交通大學
BEIJING JIAOTONG UNIVERSITY