线性回归

Yasaka 陈博

有监督的机器学习

- data中包含(X,Y)

- 有些Y标记是连续的，回归
- 有些Y标记是离散的，分类

理解回归

- 回归这个词追根溯源来源于高尔顿，达尔文是高尔顿的表哥

- 父亲是比较高的，儿子也是比较高的
- 父亲是比较矮的，儿子也是比较矮的
- 父亲是2.26，儿子可能很高但是不会达到2.26
- 父亲是1.65，儿子可能不高，但是比1.65高

- 大自然让我们回归到一定的区间之内

最大似然估计

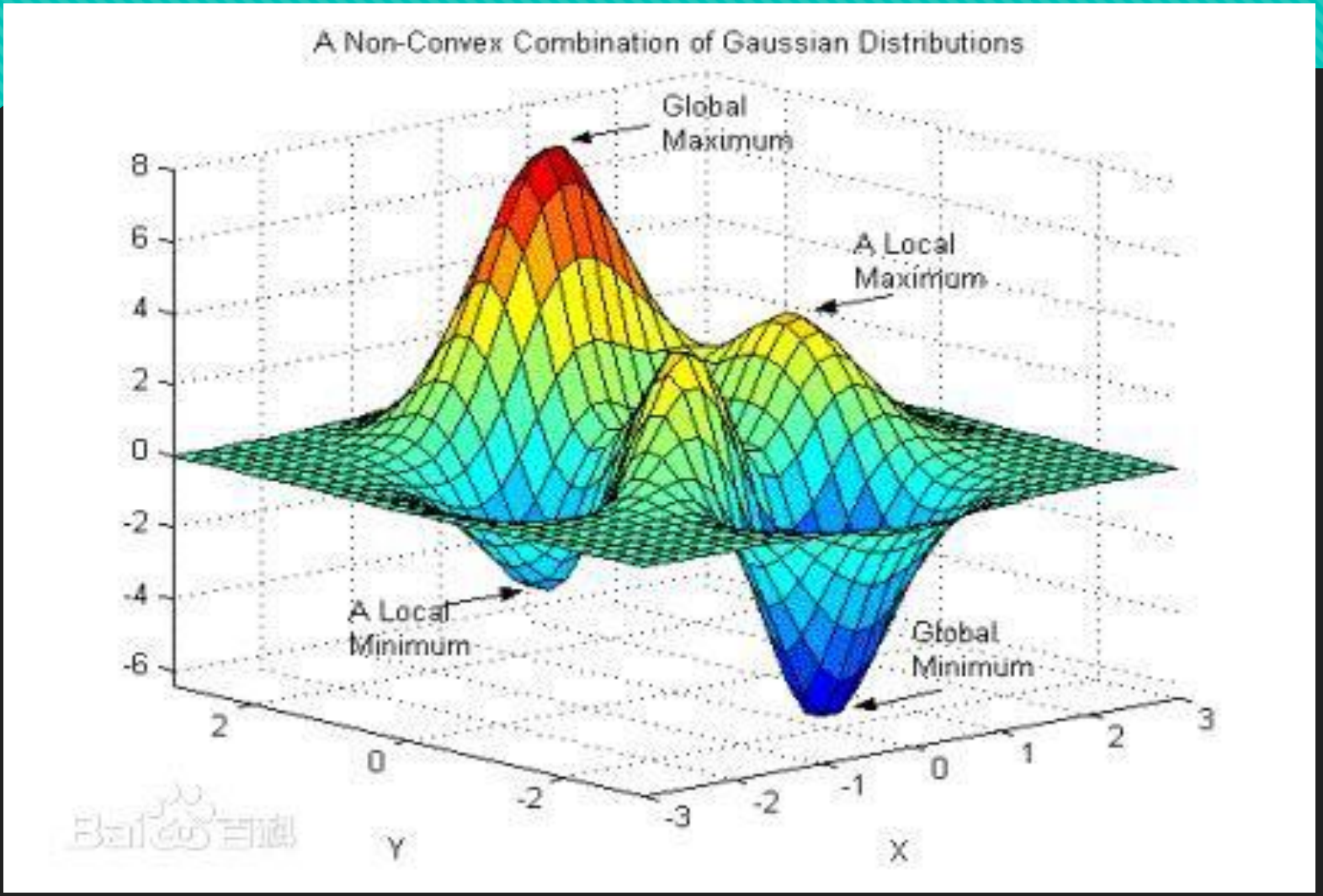
- 最大似然估计是一种统计方法，它用来求一个样本集的相关概率密度函数的参数。这个方法最早是遗传学家以及统计学家罗纳德·费雪fisher爵士在1912年至1922年间开始使用的。
- “似然”是对likelihood 的一种较为贴近文言文的翻译，“似然”用现代的中文来说即“可能性”。故而，若称之为“最大可能性估计”则更加通俗易懂。

- 在英语语境里，likelihood 和 probability 的日常使用是可以互换的，都表示对机会 (chance) 的同义替代

线性回归

- 线性：两个变量之间存在一次方函数关系,就称它们之间存在线性关系
- 线性：线性linear,指量与量之间按比例、成直线的关系,在空间和时间上代表规则和光滑的运动

高斯分布


A Non-Convex Combination of Gaussian Distributions
Global Maximum
A Local Maximum
A Local Minimum
Global Minimum
8
6
4
2
0
-2
-4
-6
2
0
-2
-3
-2
-1
0
1
2
3
Y
X

逻辑回归

- 分类问题的首选，属于广义线性回归，卡巴斯基和司机没关系
- 二分类的，做多分类可以转多个二分类，或者Soft-max回归

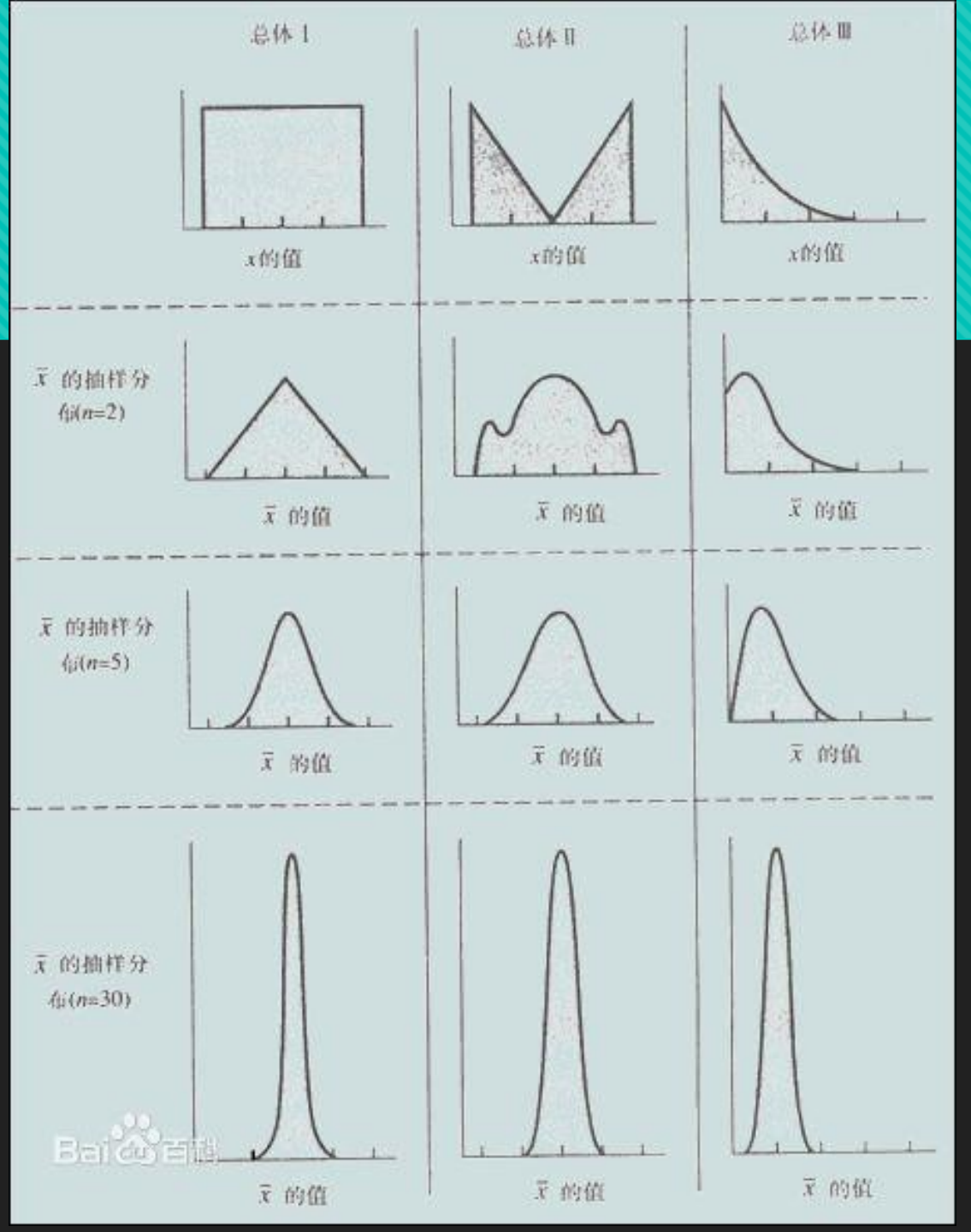
多元线性回归

- y = a*x + b
- y = w0 + w1*x1 + w2*x2
- 向量转置相乘 x0=1
- 不止两个特征
- 截距，什么都不做，本身就存在在那里！
- 物体本身就漂亮，不加修饰也漂亮

中心极限定理

- 中心极限定理（central limit theorem）是概率论中讨论随机变量序列部分和分布渐近于正态分布的一类定理。这组定理是数理统计学和误差分析的理论基础，指出了大量随机变量累积分布函数逐点收敛到正态分布的积累分布函数的条件。
- 它是概率论中最重要的一类定理，有广泛的实际应用背景。在自然界与生产中，一些现象受到许多相互独立的随机因素的影响，如果每个因素所产生的影响都很微小时，总的影响可以看作是服从正态分布的。中心极限定理就是从数学上证明了这一现象 。

中心极限定理


总体 I
总体 II
总体 III
x的值
x的值
x的值
$\bar{x}$ 的抽样分布(n=2)
$\bar{x}$ 的值
$\bar{x}$ 的值
$\bar{x}$ 的值
$\bar{x}$ 的抽样分布(n=5)
$\bar{x}$ 的值
$\bar{x}$ 的值
$\bar{x}$ 的值
$\bar{x}$ 的抽样分布(n=30)
$\bar{x}$ 的值
$\bar{x}$ 的值
$\bar{x}$ 的值

中心极限定理

- 方差我们先不管，均值我们总有办法让它去等于零0的，因为我们这里是有W0截距的，所有误差
- 我们就可以认为是独立分布的，1<=i<=m，服从均值为0，方差为某定值 什么的平方 的高斯分布
-

误差是什么

- 第i个样本实际的值 等于 预测的值 加 误差

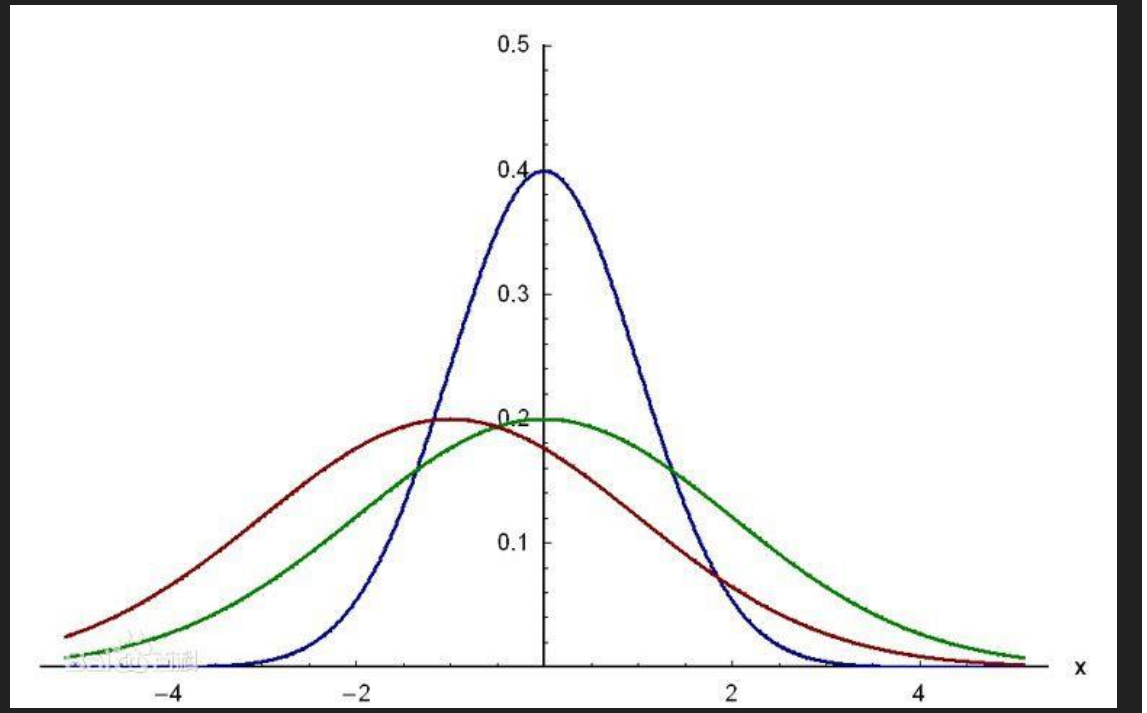
- 假定所有的样本都是独立的，有上下的震荡，震荡认为是随机变量，足够多的随机变量叠加之后形成的分布，根据中心极限定理，它服从的就是正态分布，因为它是正常状态下的分布，
- 也就是高斯分布！均值是某一个值，方差是某一个值

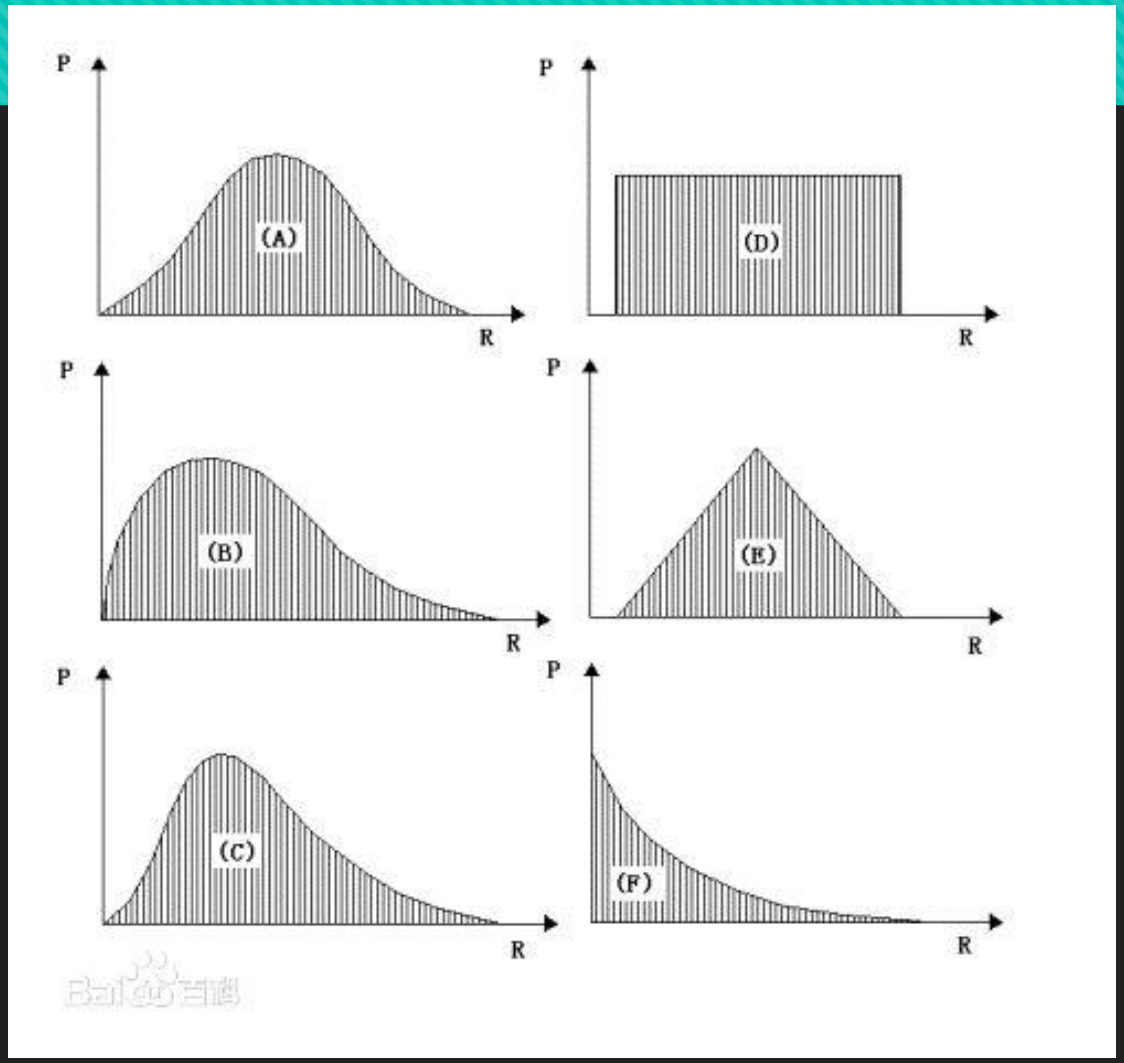
- 既然是误差符合均值为0，方差为平方的正态分布，那我们就把它的概率密度函数写出来

概率密度函数

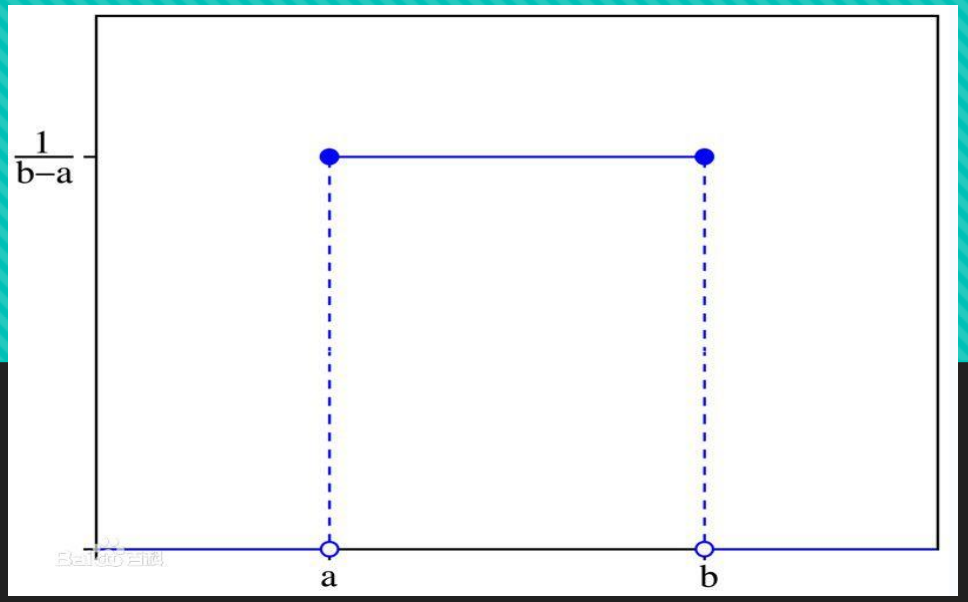
- 在数学中，连续型随机变量的概率密度函数是一个描述这个随机变量的输出值，在某个确定的取值点附近的可能性的函数。而随机变量的取值落在某个区域之内的概率则为概率密度函数在这个区域上的积分

概率密度函数





概率密度函数



- 最简单的概率密度函数是均匀分布的密度函数。最简单的概率密度函数是均匀分布的密度函数。也就是说，当x不在区间[a,b]上的时候，函数值等于0；而在区间[a,b]上的时候，函数值等于这个函数 。这个函数并不是完全的连续函数，但是是可积函数。

- 正态分布是重要的概率分布。它的概率密度函数是：
- 随着参数μ和σ变化，概率分布也产生变化

$$f(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

把公式中的误差带入概率密度函数

$$y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}$$

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right)$$

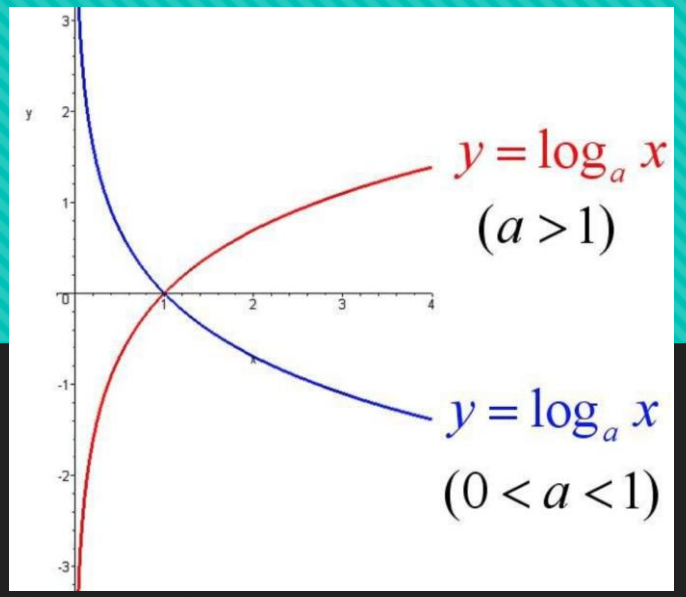
$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{m} p(y^{(i)} \mid x^{(i)};\theta) \\ &= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \end{aligned}$$

似然函数与概率密度函数

- $L(\theta|x)=f(x|\theta)$
- 这个等式表示的是对于事件发生的两种角度的看法。其实等式两边都是表示的这个事件发生的概率或者说可能性。再给定一个样本x后，我们去想这个样本出现的可能性到底是多大。统计学的观点始终是认为样本的出现是基于一个分布的。那么我们去假设这个分布为f，里面有参数theta。对于不同的theta，样本的分布不一样。$f(x|\theta)$表示的就是在给定参数theta的情况下，x出现的可能性多大。$L(\theta|x)$表示的是在给定样本x的时候，哪个参数theta使得x出现的可能性多大。所以其实这个等式要表示的核心意思都是在给一个theta和一个样本x的时候，整个事件发生的可能性多大。

对数似然函数



- 形式化就是：在一定条件下，最大化对数似然目标函数等价于最大化 P(θ | X)，而 P(θ | X) 就是我们的学习目标：给定训练数据 X 的条件下，找到最可能出现的参数 θ
- 大致推理过程如下：argmax_θ P(θ | X) = argmax_θ P(X | θ)P(θ)/P(X), 对于每个 θ，P(X) 都相等，且我们假设 P(θ) 服从均匀分布，因此有 argmax_θ P(θ | X) = argmax_θ P(X | θ)。加上对数只是为了求解方便（将乘积转换为求和，去掉 exp 项等），但是并不改变最终结果。
- ln 即自然对数 ln a=$\log_e$ a

为什么是最小二乘法

- 我们要的是对数似然函数取最大，所以是max(logL(W))

目标函数

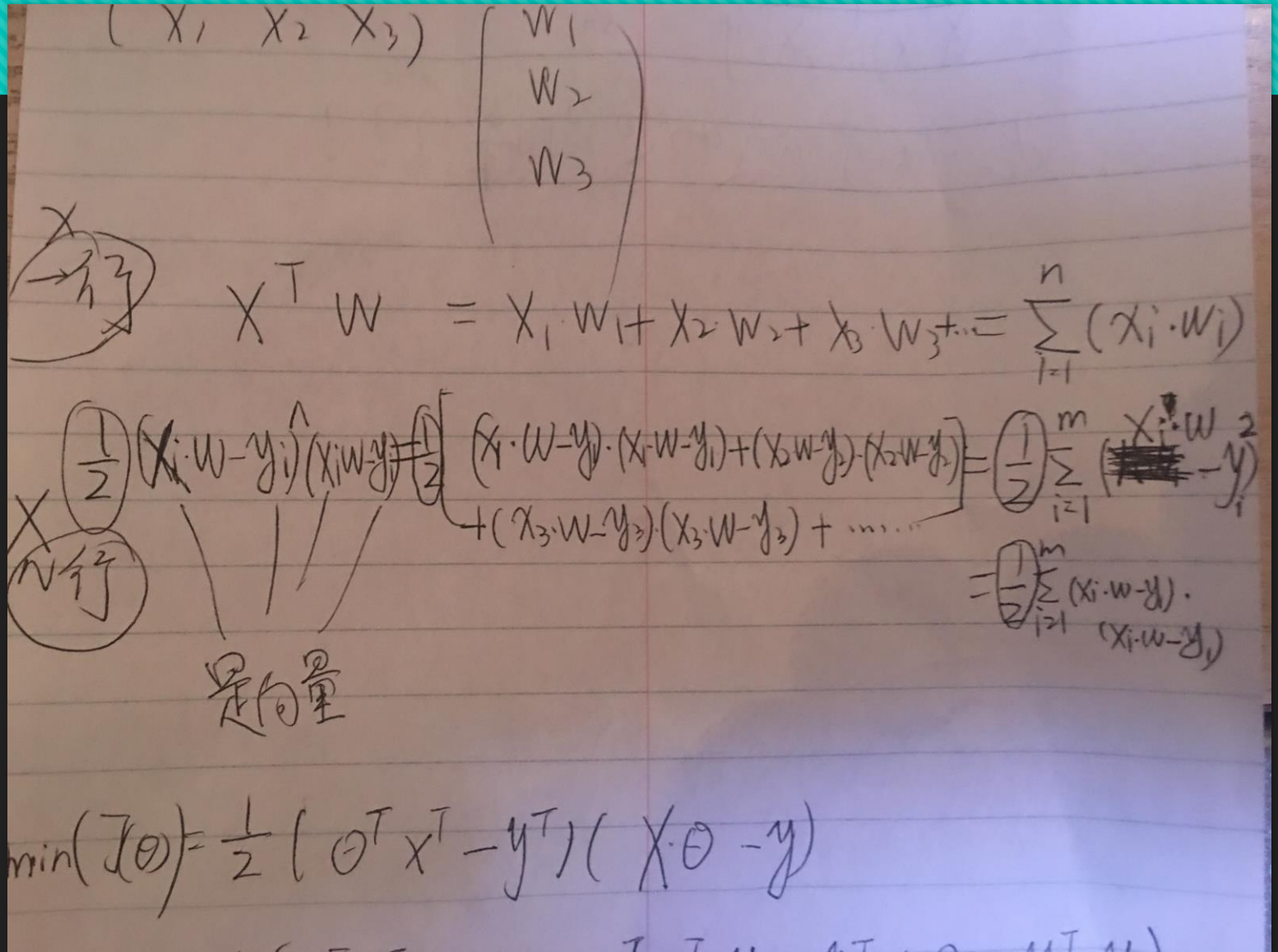
$$
\begin{aligned}
\ell(\theta) &= \log L(\theta) \\
&= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\
&= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\
&= m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^{m} (y^{(i)} - \theta^T x^{(i)})^2
\end{aligned}
$$

$$
J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2
$$

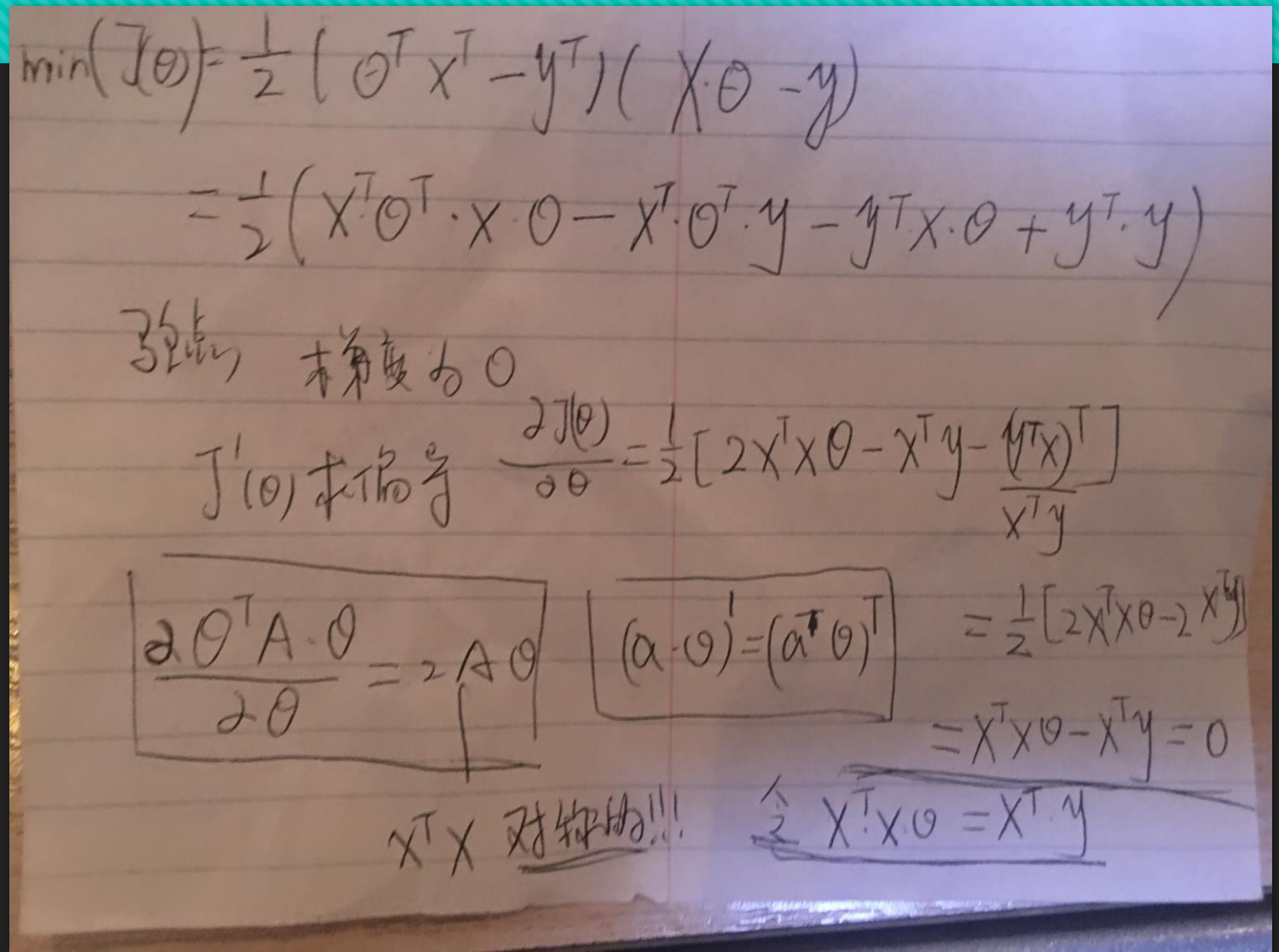
带有假设

- 这种最小二乘估计，其实我们就可以认为，假定了误差服从正太分布，认为样本是独立的，使用最大似然估计，就能得出结论！
- ML学习特点，不强调模型100%正确，是有价值的，堪用的！

误差函数的另一种表达


$(x_1\ x_2\ x_3)\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}$
$X^T w = x_1 w_1 + x_2 w_2 + x_3 w_3 + \cdots = \sum_{i=1}^{n}(x_i \cdot w_i)$
$\frac{1}{2}(x_i \cdot w - y_i)^2 = \frac{1}{2}[(x_1 \cdot w - y_1)\cdot(x_1 w - y_1) + (x_2 w - y_2)\cdot(x_2 w - y_2) + (x_3 \cdot w - y_3)(x_3 \cdot w - y_3) + \cdots] = \frac{1}{2}\sum_{i=1}^{m}(x_i \cdot w - y_i)^2$
$= \frac{1}{2}\sum_{i=1}^{m}(x_i \cdot w - y)\cdot(x_i \cdot w - y)$
是向量
$\min(J(\theta)) = \frac{1}{2}(\theta^T x^T - y^T)(X\cdot\theta - y)$

$$J(\theta)=\frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2=\frac{1}{2}(X\theta-y)^T(X\theta-y)$$



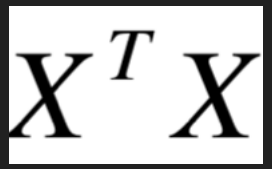
$$\theta=\left(X^TX\right)^{-1}X^Ty$$

判定为凸函数

- 凸函数有许多判定方法，比如：定义，一阶条件，二阶条件等等。你这里说利用正定性判定，应该是指使用二阶条件判定。
- 半正定一定是凸函数，开口朝上的，半正定一定有极小值
- 在用二阶条件进行判定时，需要得到Hessian矩阵，然后根据Hessian的正定性判定函数的凹凸性，比如Hessian矩阵半正定，函数为凸函数；Hessian矩阵正定，严格凸函数
- Hessian矩阵：
- 黑塞矩阵（Hessian Matrix），又译作海森矩阵、海瑟矩阵、海塞矩阵等，是一个多元函数的二阶偏导数构成的方阵，描述了函数的局部曲率。

黑塞矩阵

- 黑塞矩阵是由目标函数 在点X处的二阶偏导数组成的对称矩阵
- 正定：
- 对A的特征值全为正数，那么是正定的。
- 不正定，那么就非正定或半正定。若A的特征值大于等于，则半正定。否则非正定。
-
- 对J损失函数求二阶导，之后得到的一定是半正定的，自己和自己做点乘嘛！



解析解

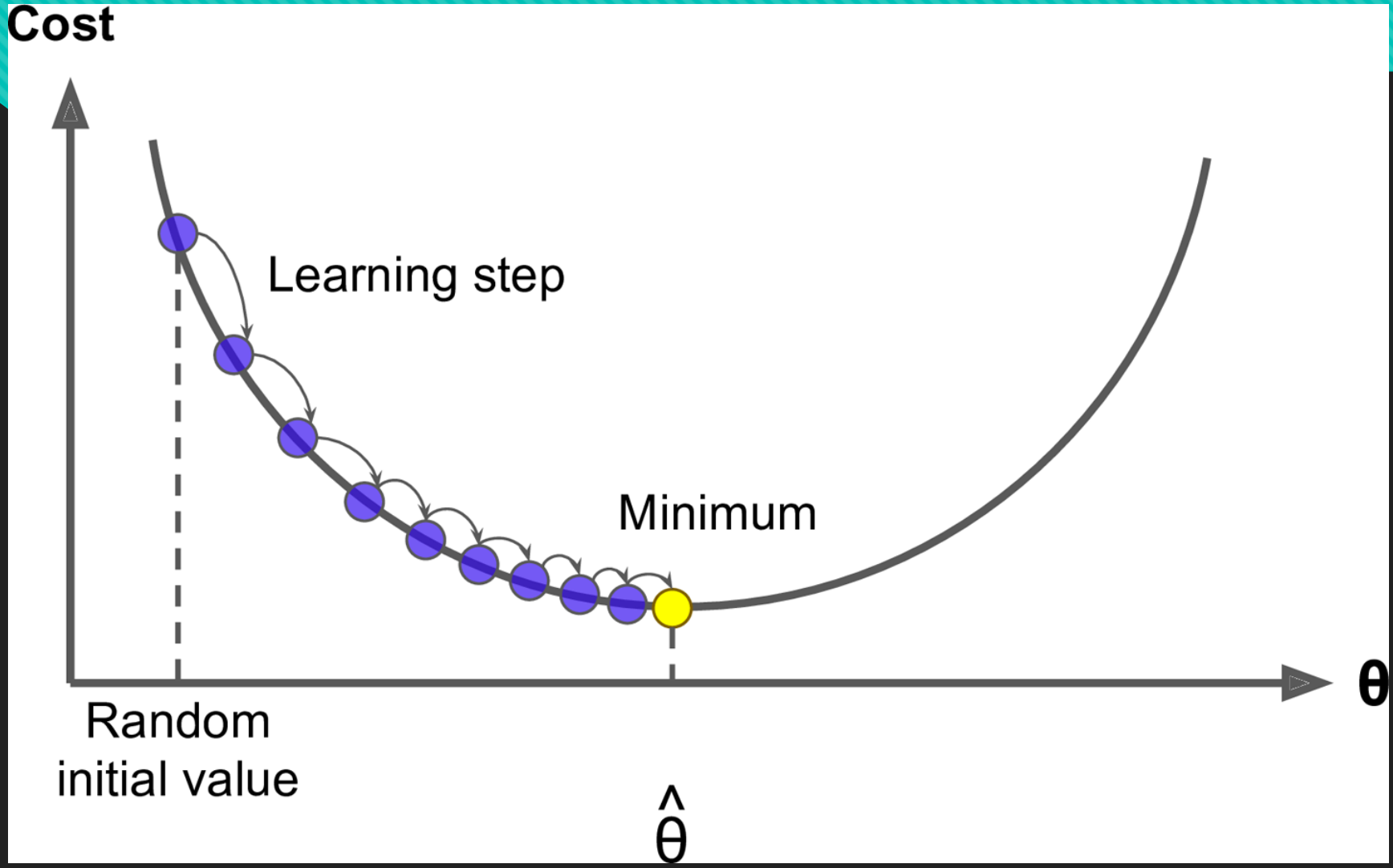
$$\theta = \left(X^T X\right)^{-1} X^T y$$

- 数值解是在一定条件下通过某种近似计算得出来的一个数值，能在给定的精度条件下满足方程 解析解为方程的解析式（比如求根公式之类的），是方程的精确解，能在任意精度下满足方程

梯度下降法

- 上面利用公式求解里面对称阵是N维乘以N维的，复杂度是是O N的三次方，换句话说，就是如果你的特征数量翻倍，你的计算时间大致上要2的三次方，8倍的慢

梯度下降法



求导

$$\theta = \theta - \alpha \cdot \frac{\partial J(\theta)}{\partial \theta}$$

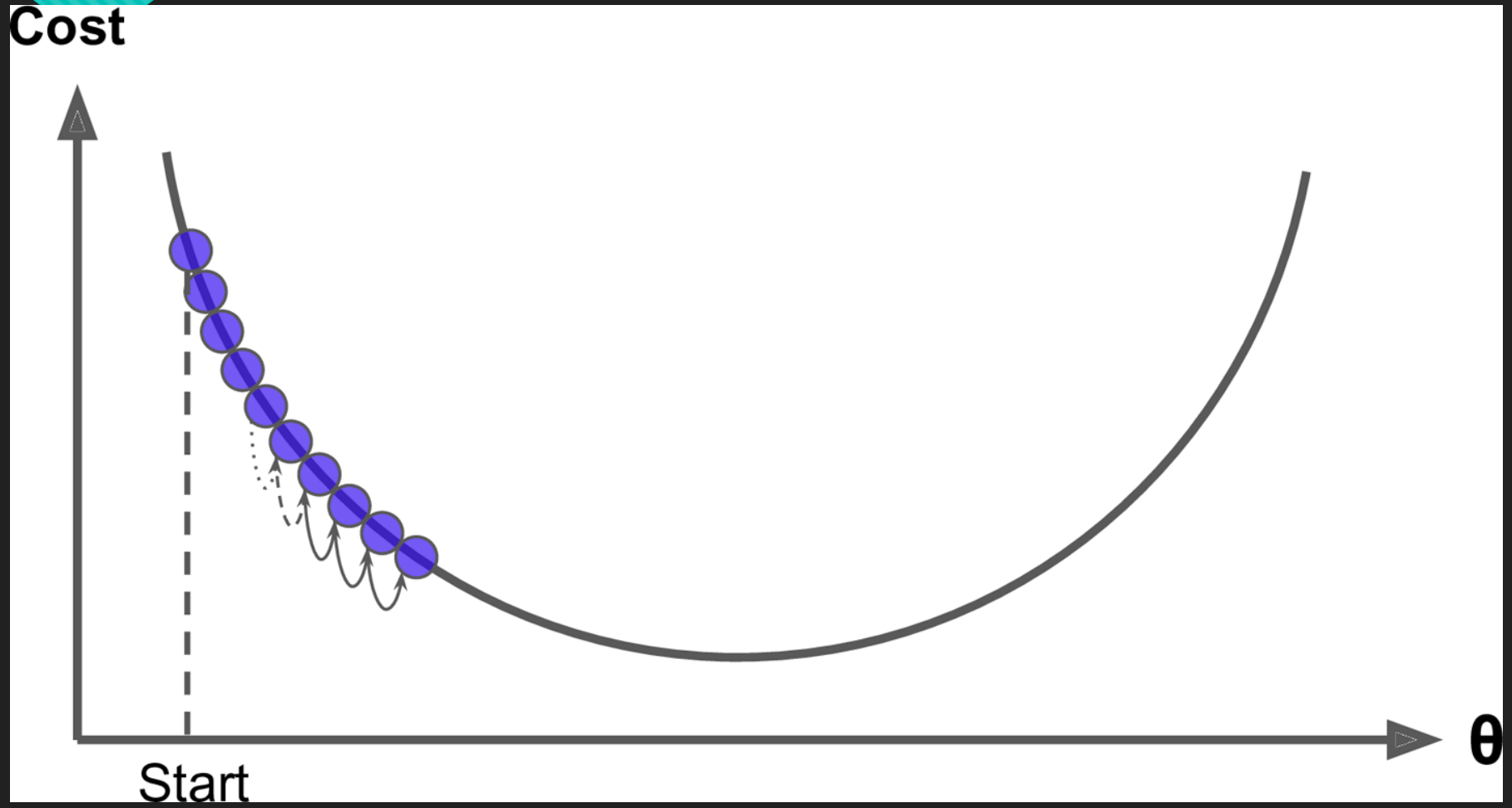
$$\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} (h_\theta(x) - y)^2 \\
&= 2 \cdot \frac{1}{2} (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} (h_\theta(x) - y) \\
&= (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^{n} \theta_i x_i - y \right) \\
&= (h_\theta(x) - y) \, x_j
\end{aligned}$$

批量梯度下降

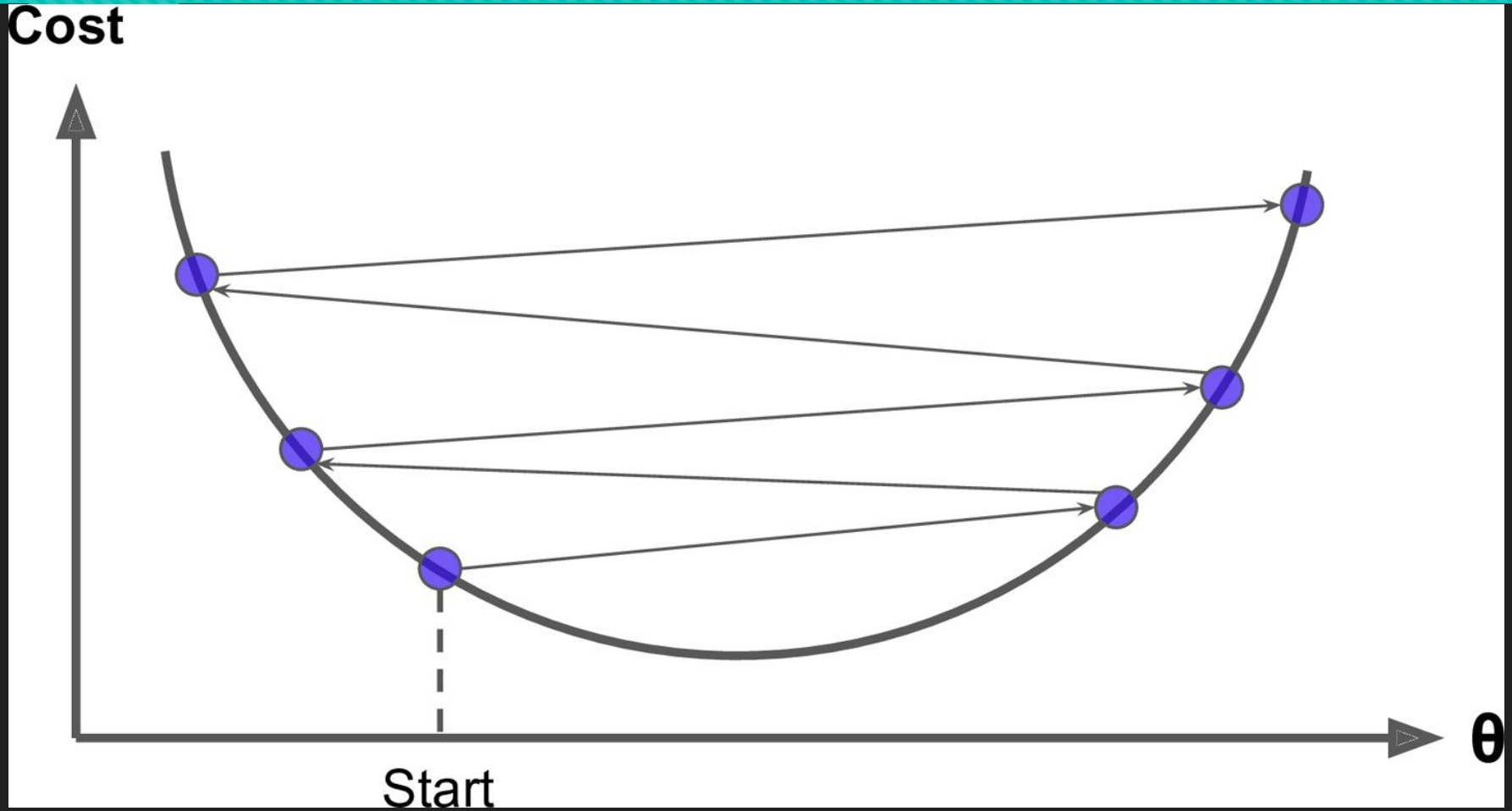
$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)}) \right) x_j^{(i)}$$

- 初始化W，随机w，给初值
- 沿着负梯度方向迭代，更新后的w使得J(w)更小
- 如果w维度是几百维度，直接算SVD是可以的，几百维度以上一般是梯度下降算法，这个更像是机器学习，学习嘛，埃尔法是学习率、步长
- 上面这个公式其实是把所有样本梯度加起来的结果，因为有个加和符合从1到m，然后更新每个w的
- 卷积神经网络，最常见的算法我们就是用梯度下降

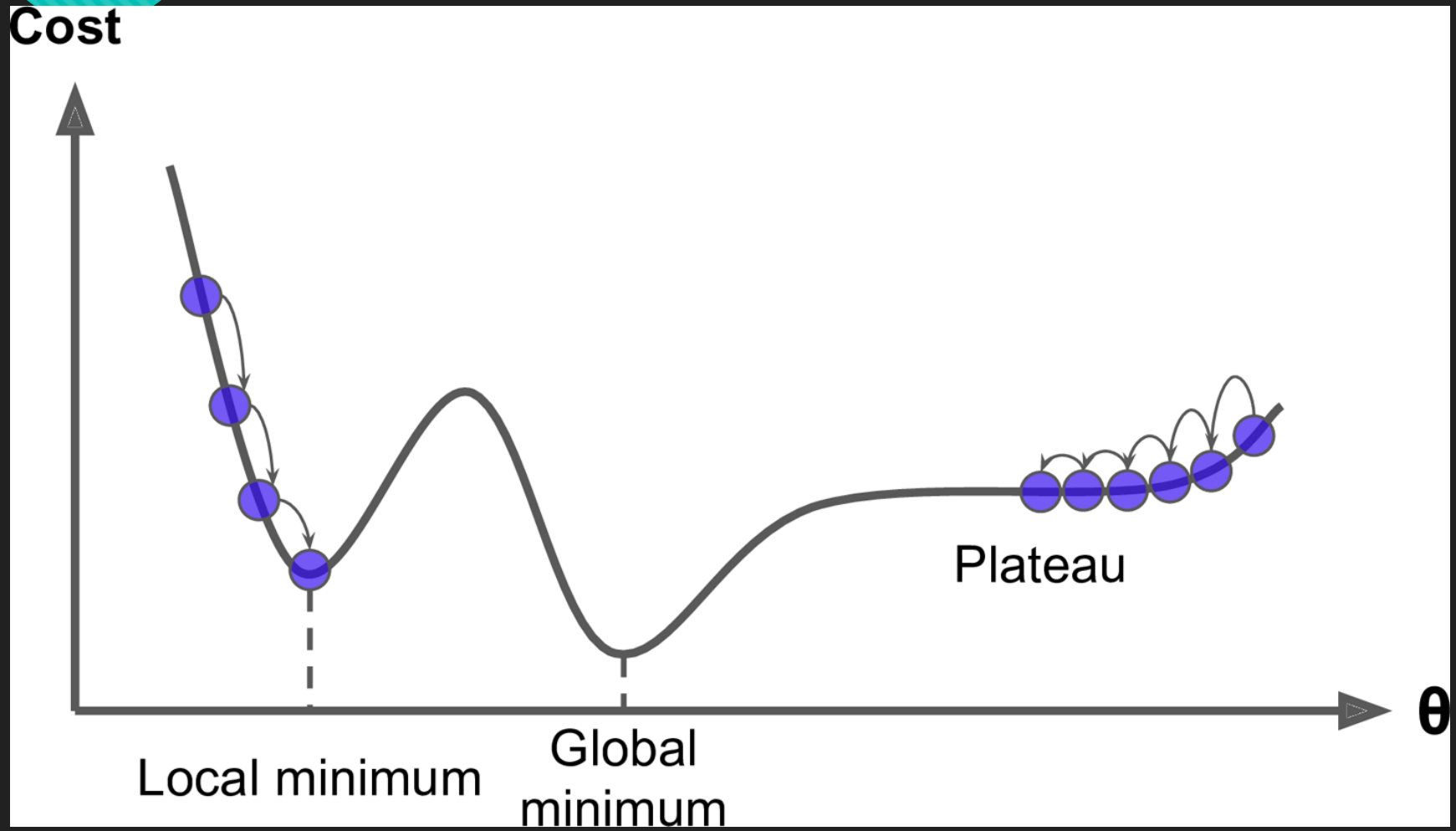
当Step小的时候



当Step大的时候



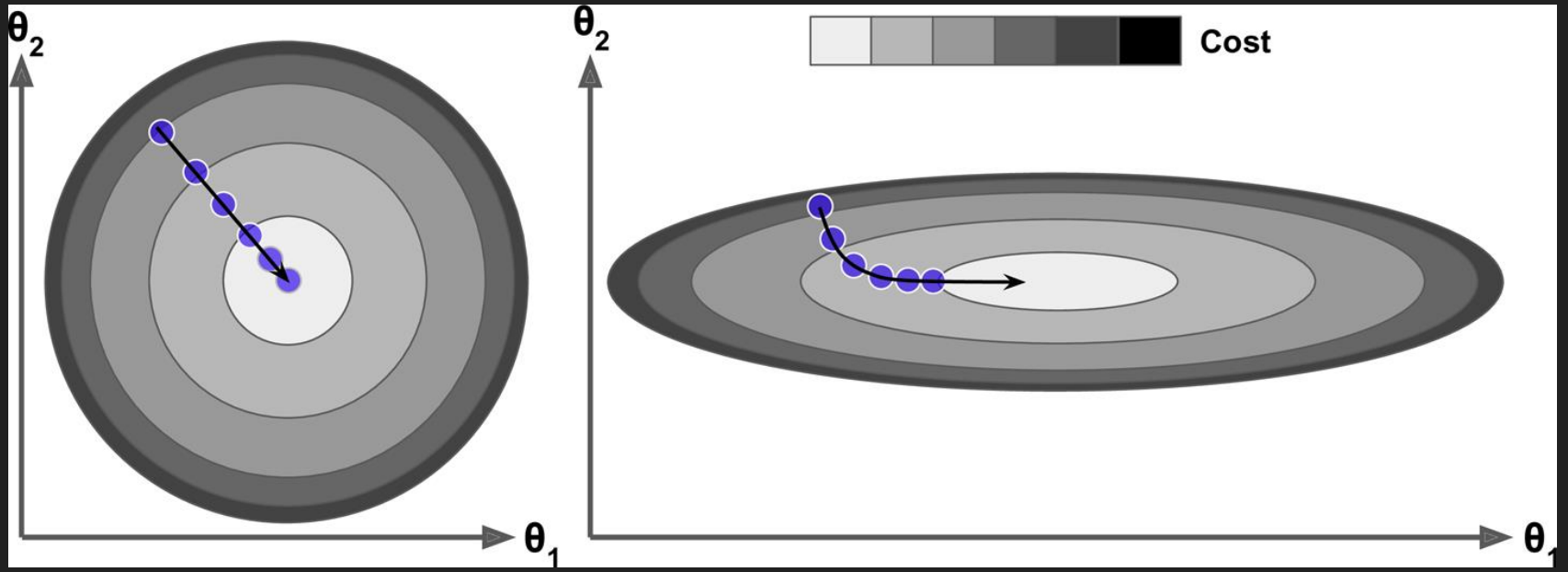
全局最优解



全局最优解

- 其实很多时候大家最后人工智能做多了，就发现很多时候不要去纠结全局最优，就像找终身伴侣怎么找的？你并不是在全球30亿女性中找一个最好的，你一般是在你朋友圈里面找一个，一个模型是堪用的，work的
- 我们只不过在线性回归模型中，目标函数求二阶导，是半正定的，是凸函数的
-
- 批量梯度下降是稳健的，贪心算法，不一定找到最优，但是一定可以找到更好的

归一化和没有归一化的特征



随机梯度下降

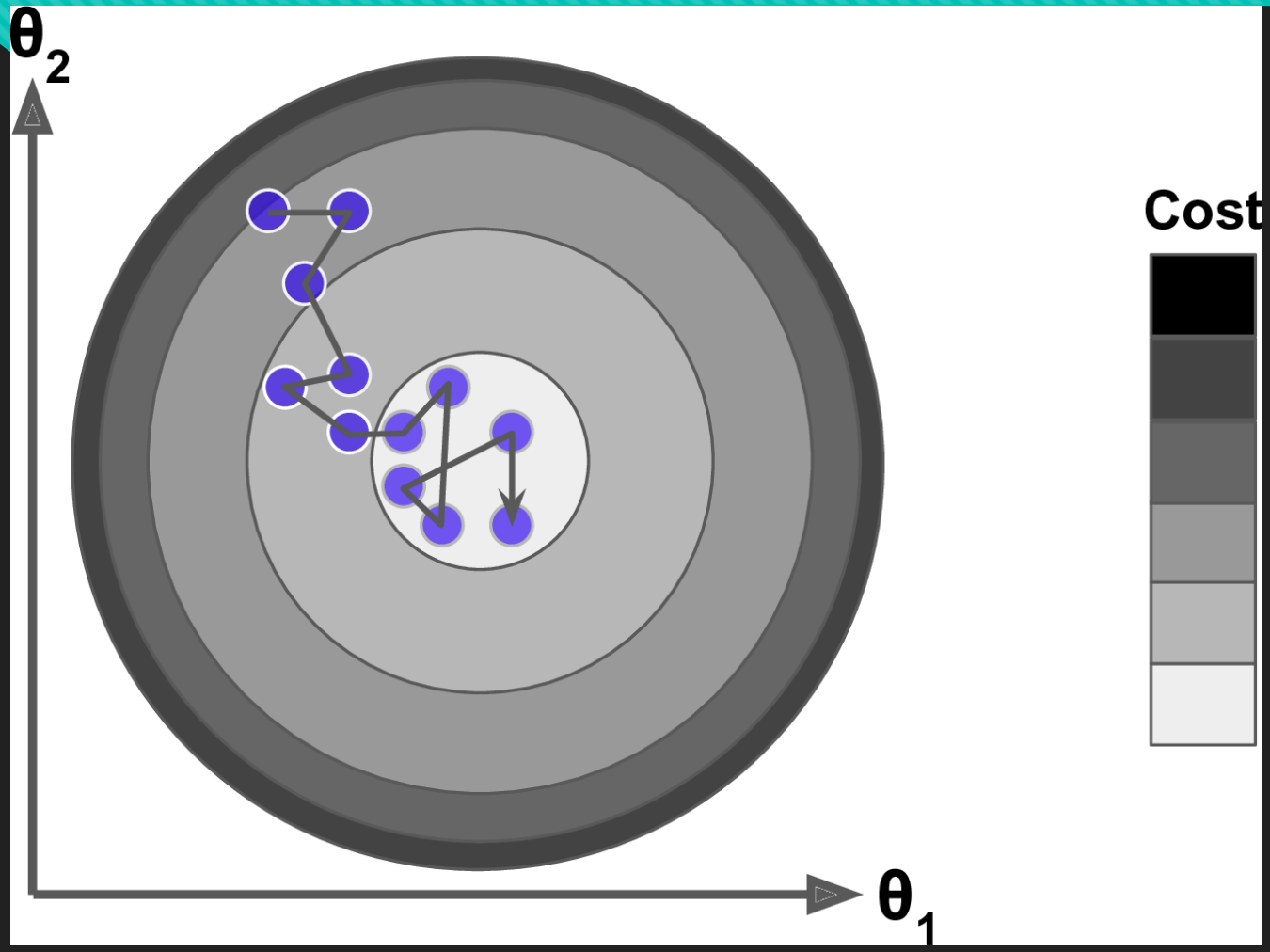
- 优先选择随机梯度下降
- 有些时候随机梯度下降可以跳出局部最小值

for i=1 to m, {

$$\theta_j := \theta_j + \alpha \left(y^{(i)} - h_\theta(x^{(i)}) \right) x_j^{(i)}$$

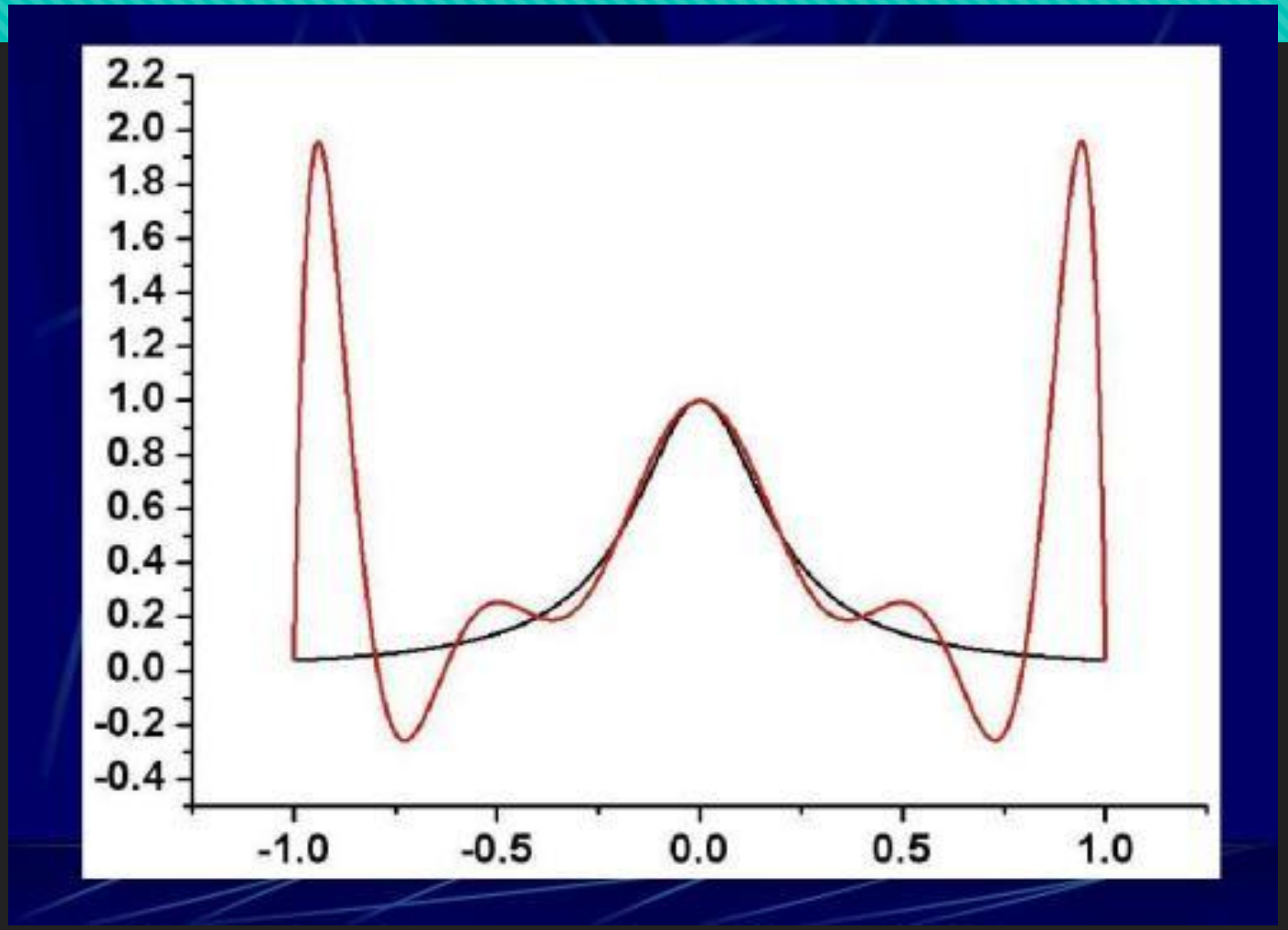
}

随机梯度下降



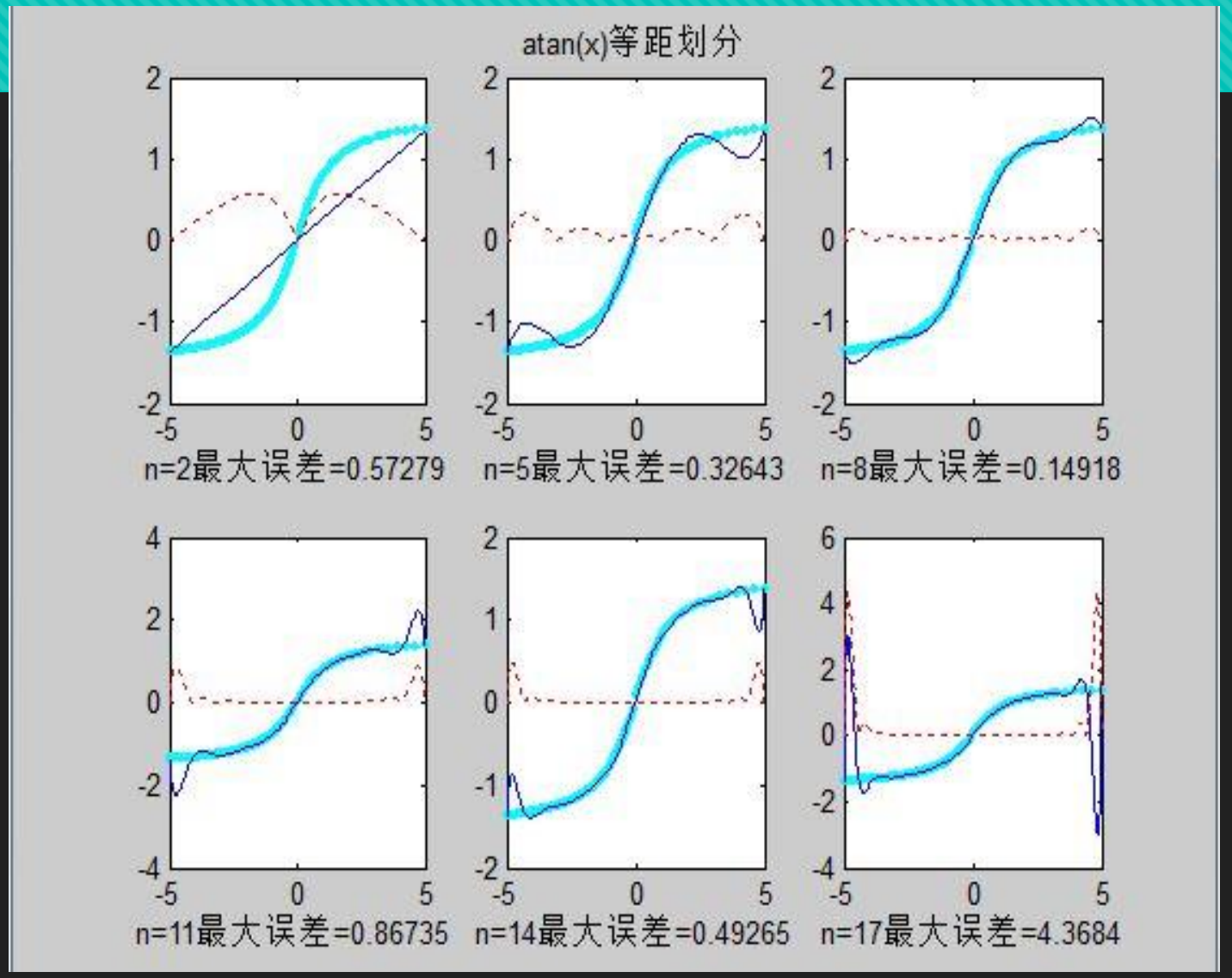
多项式回归

- 龙格现象

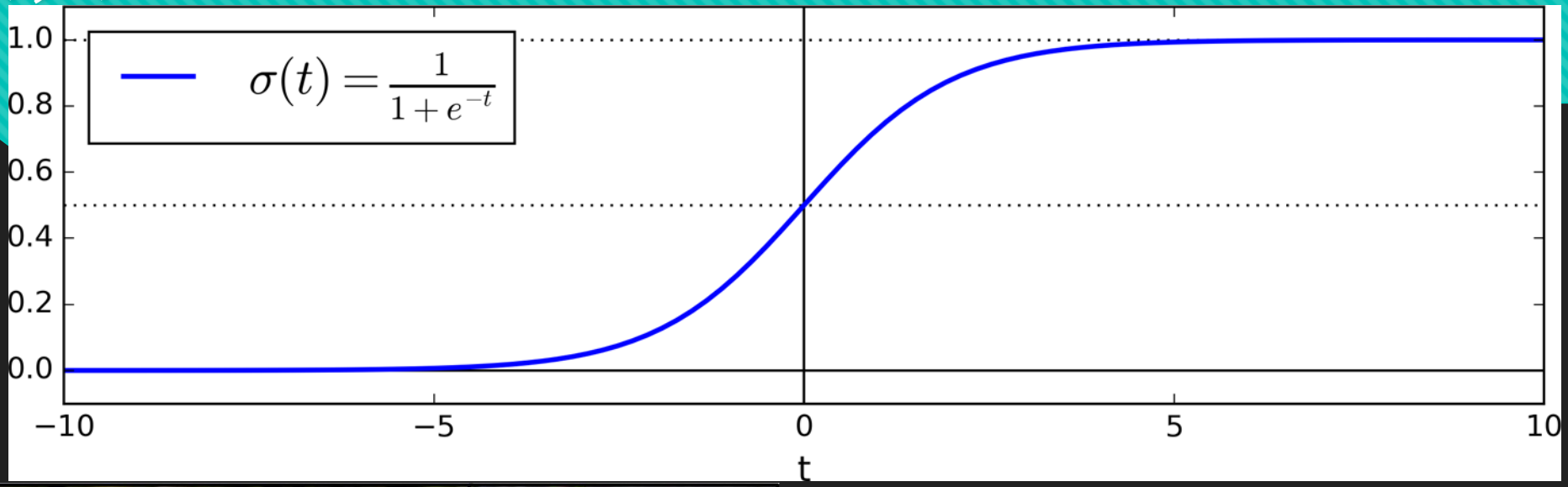


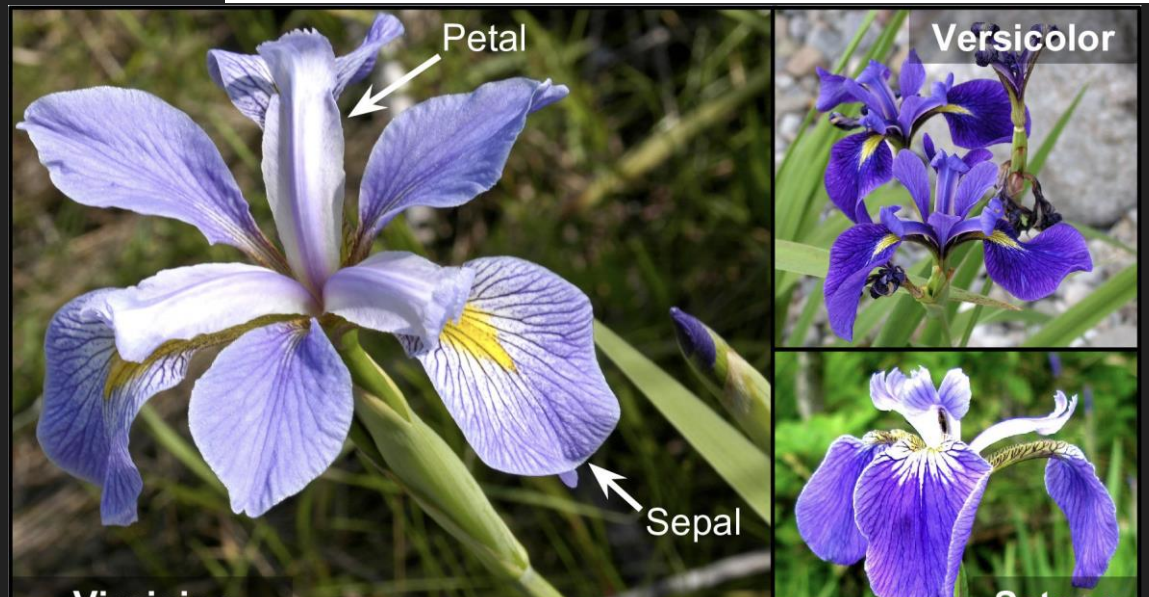
过拟合

- 任何东西都是有噪声的，就像学习，
- 有人不是不学，是没学到点儿上！



逻辑回归





$$\sigma(t) = \frac{1}{1 + \exp(-t)}$$

Soft-max

$$\hat{p}_k = \sigma(\mathbf{s}(\mathbf{x}))_k = \frac{\exp\left(s_k(\mathbf{x})\right)}{\sum_{j=1}^{K} \exp\left(s_j(\mathbf{x})\right)}$$