

人工智能

——样例学习 II



饶洋辉

数据科学与计算机学院,

中山大学

raoyangh@mail.sysu.edu.cn

期望 (Expectation)

- 如果 X 是一个离散随机变量，则：

$$E[X] = \sum_i x_i P\{X = x_i\}$$

- 如果 X 是一个服从概率密度函数 f 的连续随机变量

$$E[X] = \int_{-\infty}^{\infty} xf(x)dx$$

期望

- 假设 X 是抛一个六面的骰子朝上的那面的数值，那么： $E[X]$?

$$E[X] = \sum_{x=1}^6 xp(x) = \frac{1}{6} \sum_{x=1}^6 x = \frac{21}{6}$$

中位数 (Median)

- 对 n 个变量排序
 - $X(1) \leq X(2) \leq \dots \leq X(n)$
- 如果 n 为奇数
 - $X((n+1)/2)$
- 如果 n 为偶数
 - $(X(n/2) + X(1+n/2))/2$

众数 (Mode)

- 10 5 9 12
- 6 5 9 8 5
- 25 28 28 36 25 42

方差 (Variance)

- $\text{Var}(X) = E[(X-E[X])^2] = E[X^2] - (E[X])^2$

X	$E(X)$	$(X-E(X))^2$	X^2
1	2	1	1
2	2	0	4
3	2	1	9

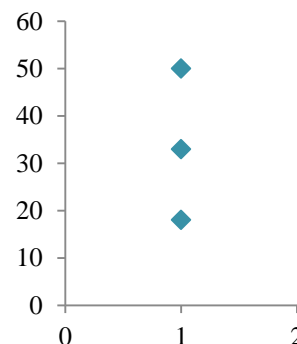
协方差 (Covariance)

- $$\begin{aligned}\text{Cov}(X, Y) &= E[(X - E(X))(Y - E(Y))] \\ &= E[XY - E(X)Y - XE(Y) + E(X)E(Y)] \\ &= E[XY] - E(X)E[Y] - E[X]E(Y) + E(X)E(Y) \\ &= E[XY] - E[X]E[Y]\end{aligned}$$

相关系数 (Correlation)

- 如果 X 和 Y 是两个独立随机变量，那么 $\text{Cov}(X, Y) = 0$

性别	年龄
1	18
1	50
1	33



- 随机变量 X 与 Y 之间的相关系数是：

$$\text{Corr}(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X) \text{Var}(Y)}}$$

线性回归

- 最小二乘法

$$y_i = w_1 x_i + w_0 + \varepsilon$$

$$n^{-1} \sum_{i=1}^n (y_i - w_0 - w_1 x_i) = 0 \quad \text{令}\varepsilon\text{均值为}0$$

$$n^{-1} \sum_{i=1}^n x_i (y_i - w_0 - w_1 x_i) = 0$$

x 和 ε 协方差为 0, 即线性独立,
干扰项 ε 与 x 变化无关.

$$Q(w_0, w_1) = \min_{w_0, w_1} \sum_{i=1}^n (y_i - w_0 - w_1 x_i)^2$$

令偏差和最小化

$$\partial Q(w_0, w_1) / \partial w_0 = 0$$

$$\partial Q(w_0, w_1) / \partial w_1 = 0$$

$$-2 \sum_{i=1}^n (y_i - w_0 - w_1 x_i) = 0$$

$$-2 \sum_{i=1}^n x_i (y_i - w_0 - w_1 x_i) = 0$$

线性回归

多元: $Q: y = w_0 + w_1 x_1 + w_2 x_2 + \dots$

梯度下降法: $w^{\text{new}} = w^{\text{old}} - \eta \frac{\partial Q}{\partial w}$

- 最小二乘法

$$w_0 = \bar{y} - w_1 \bar{x}$$

$$w_1 = \frac{\sum_{i=1}^n x_i (y_i - \bar{y})}{\sum_{i=1}^n x_i (x_i - \bar{x})}$$

$$= \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$\frac{x, y \text{ 的协方差}}{x \text{ 的方差}}$

逻辑回归模型

- 如果使用最小二乘法的回归模型来做二分类任务：^{0, 1}

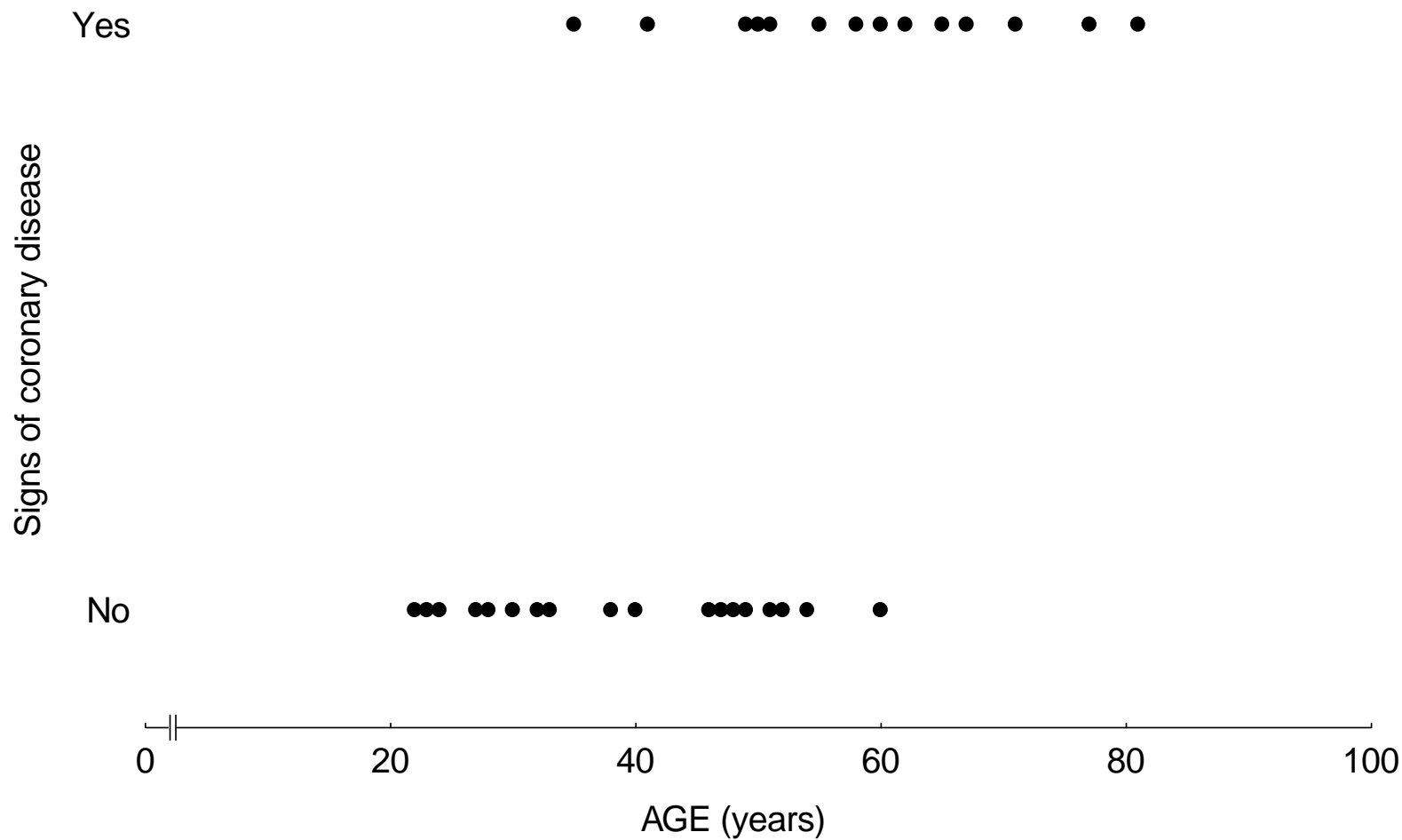
$$y = w_0 + \sum_{j=1}^d w_j x_j + u$$

$$= \tilde{\mathbf{W}}^T \tilde{\mathbf{X}}$$

$$\mathbf{w}^T = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{pmatrix} \quad \tilde{\mathbf{X}} = \begin{pmatrix} 1 \\ x_{11} \\ \vdots \\ x_{1d} \end{pmatrix}$$

- 基于上述模型预测的y值，即样本属于某个类的概率，会超出0到1的范围。

逻辑回归模型



逻辑回归模型

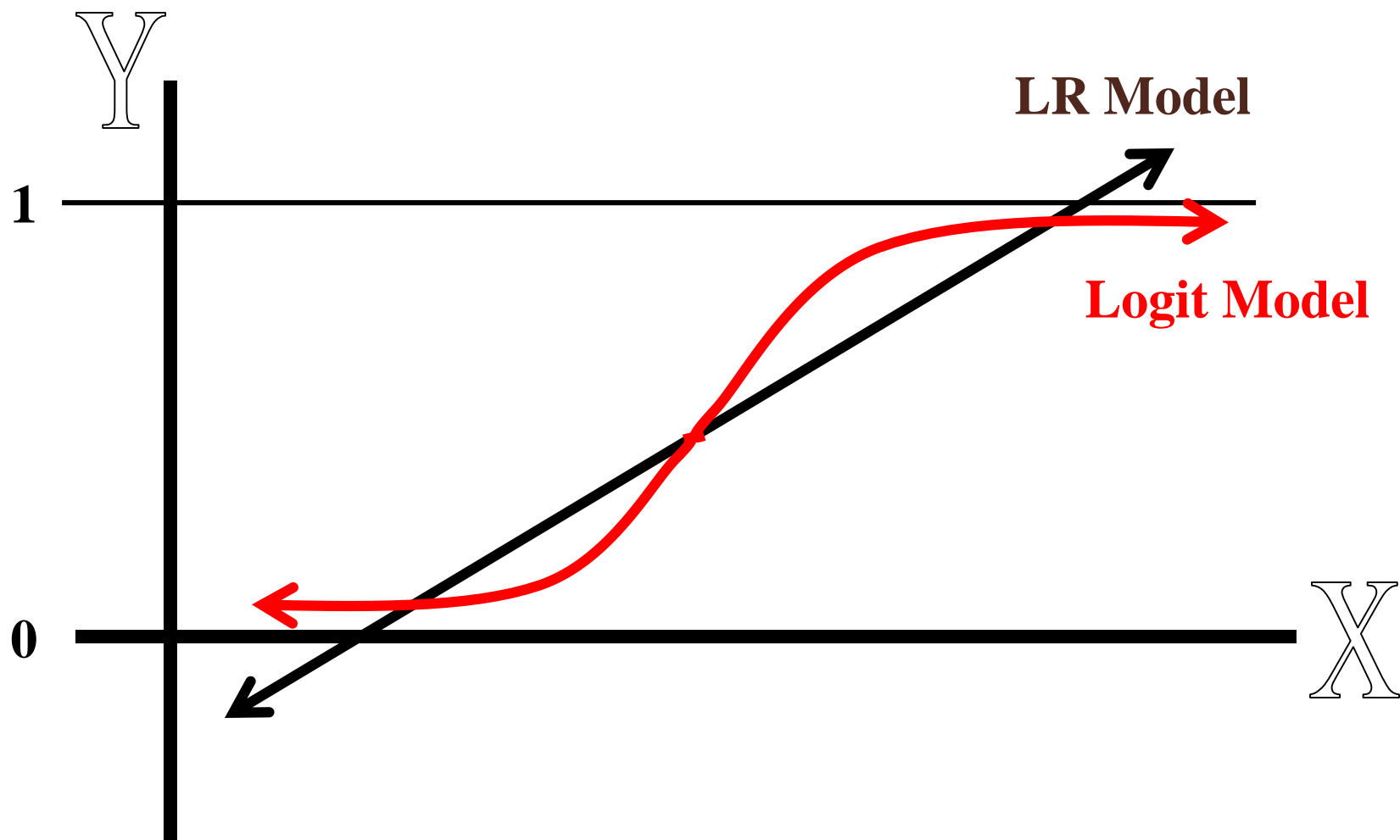
- “logit” 变换可以解决上述问题:

值域为 $[-\infty, +\infty]$
反推 p 值

$$\log\left(\frac{p}{1-p}\right) = w_0 + \sum_{j=1}^d w_j x_j + u$$
$$= \tilde{\mathbf{W}}^T \tilde{\mathbf{X}}$$

- p 是事件 y 发生的概率, 比如: $p = p(y=1 | \mathbf{X})$
- $p/(1-p)$ 称为机率比或优势比 (odds ratio)
- $\log[p/(1-p)]$ 是机率比的对数, 或称为 "logit"

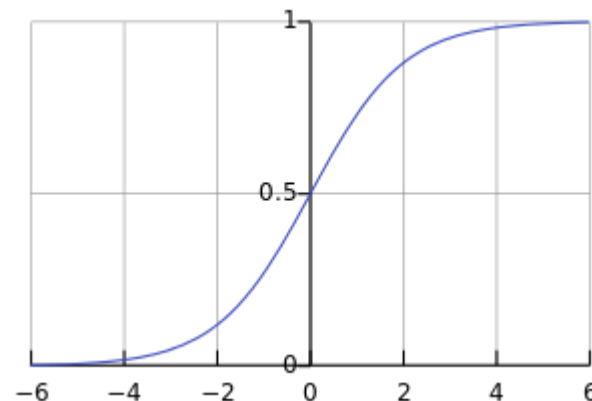
逻辑回归模型



逻辑回归模型

- logistic 函数使得输出的概率值在0到1的范围内.
- 样本 \mathbf{X} 标签为正的的概率 $p(y=1|\mathbf{X})$ 是:

$$p = \frac{1}{1 + e^{-w_0 - \sum_{j=1}^d w_j x_j}} = \frac{e^{w_0 + \sum_{j=1}^d w_j x_j}}{1 + e^{w_0 + \sum_{j=1}^d w_j x_j}}$$
$$= \frac{1}{1 + e^{-\tilde{\mathbf{w}}^T \tilde{\mathbf{x}}}} = \frac{e^{\tilde{\mathbf{w}}^T \tilde{\mathbf{x}}}}{1 + e^{\tilde{\mathbf{w}}^T \tilde{\mathbf{x}}}}$$



- 如果 $w_0 + \sum_{j=1}^d w_j x_j = 0$, 那么 $p = 0.5$
- 当 $w_0 + \sum_{j=1}^d w_j x_j$ 很大时, p 趋近于 1
- 当 $w_0 + \sum_{j=1}^d w_j x_j$ 很小时, p 趋近于 0

逻辑回归模型

- 最小二乘法回归模型，使用了最小二乘的公式，直接得到了最终的模型。
- 对于逻辑回归，可以使用极大似然估计，配合以一种迭代式的方法，计算出最终的模型。
- 算法：
 - 首先，随机初始化权重，并对某个样本进行预测；
 - 接着，计算这个模型在这次预测上的误差，改变权重，以提高模型在这个样本上的似然度；
 - 重复这个过程，直到模型收敛，即当前模型和上一步的模型的表现相差无几。
- 这个想法的本质是：找到一个最有可能产生你观察到的数据的参数。

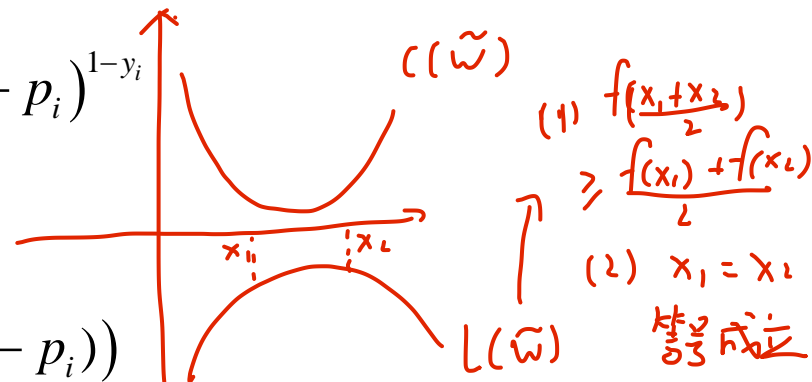
逻辑回归模型

- 似然函数: $\prod_{i=1}^n (p_i)^{y_i} (1-p_i)^{1-y_i}$
- 极大似然法:

$$L(\tilde{\mathbf{W}}) = \sum_{i=1}^n (y_i \log p_i + (1-y_i) \log(1-p_i))$$

$$= \sum_{i=1}^n \left(y_i \log \frac{p_i}{1-p_i} + \log(1-p_i) \right)$$

$$= \sum_{i=1}^n \left(y_i \tilde{\mathbf{W}}^T \tilde{\mathbf{X}}_i - \log(1 + e^{\tilde{\mathbf{W}}^T \tilde{\mathbf{X}}_i}) \right)$$



爬山法 $w_{\text{new}} = w_{\text{old}} + \eta \delta$

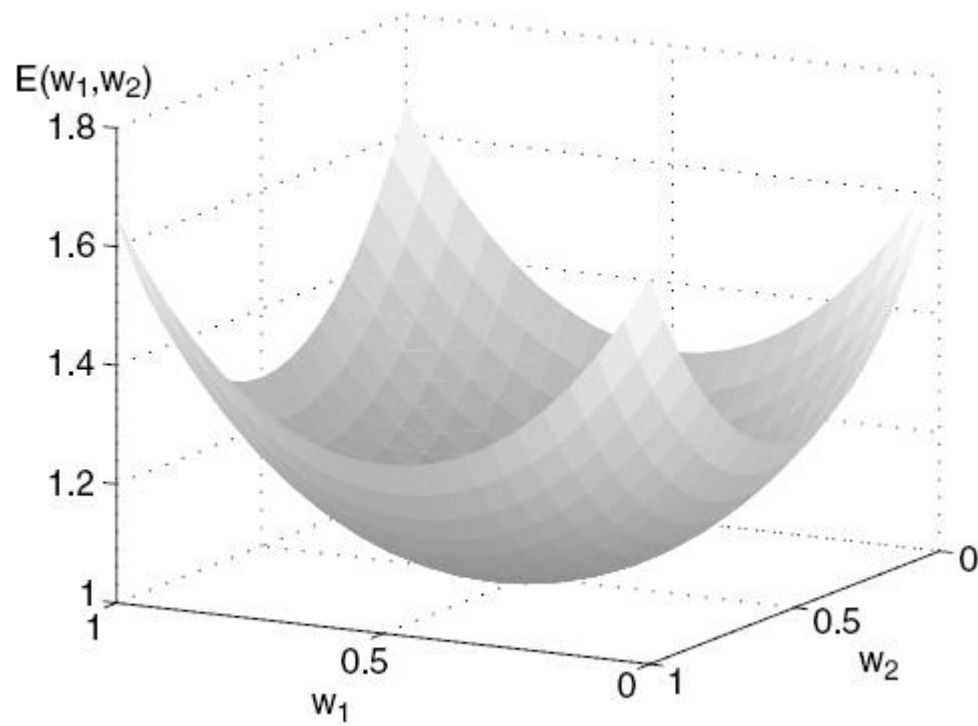
$$\frac{\partial L(\tilde{\mathbf{W}})}{\partial \tilde{\mathbf{W}}} = \sum_{i=1}^n \left[\left(y_i - \frac{e^{\tilde{\mathbf{W}}^T \tilde{\mathbf{X}}_i}}{1 + e^{\tilde{\mathbf{W}}^T \tilde{\mathbf{X}}_i}} \right) \tilde{\mathbf{X}}_i \right]$$

- 等价于最小化代价函数:

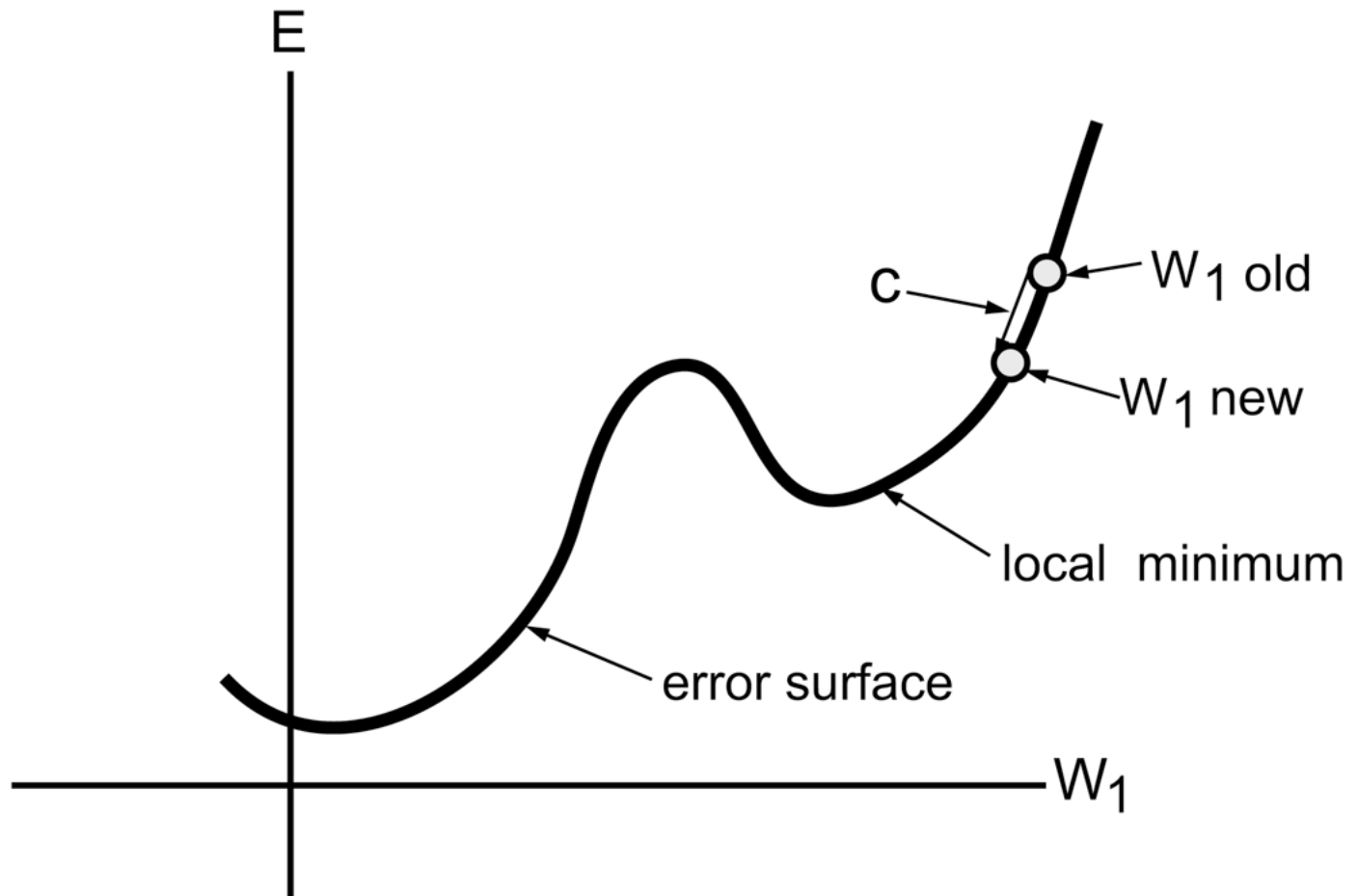
$$C(\tilde{\mathbf{W}}) = -L(\tilde{\mathbf{W}}) = -\sum_{i=1}^n (y_i \log p_i + (1-y_i) \log(1-p_i)) \quad \text{交叉熵}$$

梯度下降法

梯度下降



梯度下降



逻辑回归模型

- 梯度下降

- 计算梯度向量
- 每次用梯度向量的反方向来更新权重

- 重复: $\tilde{\mathbf{W}}_{new}^{(j)} = \tilde{\mathbf{W}}^{(j)} - \eta \frac{\partial C(\tilde{\mathbf{W}})}{\partial \tilde{\mathbf{W}}^{(j)}}$

$$= \tilde{\mathbf{W}}^{(j)} - \eta \sum_{i=1}^n \left[\left(\frac{e^{\tilde{\mathbf{W}}^T \tilde{\mathbf{X}}_i}}{1 + e^{\tilde{\mathbf{W}}^T \tilde{\mathbf{X}}_i}} - y_i \right) \tilde{\mathbf{X}}_i^{(j)} \right]$$

- 直至收敛。