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

Expectation

期望：根据总体计算
均值：根据样本数据计算

离散的

- If X is a discrete random variable

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

- If X is a continuous random variable having probability density function f

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

$$E[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} E[X_i]$$

Expectation

- If rolling one die (6-sided) and X is the value on its face, then: $E[X]$?

Expectation

- If rolling one die (6-sided) and X is the value on its face, then: $E[X]$?

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

Median

- Sort n variables
  - $X(1) <= X(2) <= \dots <= X(n)$
- If n is odd number
  - $X((n+1)/2)$
- If n is even number
  - $(X(n/2)+X(1+n/2))/2$

Mode

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

Variance 方差

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

X	$E(X)$	$(X\text{-}E(X))^2$	X^2
1	2	1	1
2	2	0	4
3	2	1	9

https://blog.csdn.net/hearthougan/article/details/77859173

Covariance

- $\mathrm{Cov}(X,Y) = E[(X-E(X))(Y-E(Y))]$

$$= E[XY - E(X)Y - XE(Y) + E(X)E(Y)]$$

$$= E[XY] - E(X)E[Y] - E[X]E(Y) + E(X)E(Y)$$

$$= E[XY] - E[X]E[Y]$$

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

Pearson correlation coefficient

Linear Regression

$y = w_0 + w_1 x + z$(扰动项)

- Least-squares solutions

迭代法求w_0, w_1

初始化：$w_0^{(0)}, w_1^{(0)}$

迭代：$w_0^{(i)} = w_0^{(i-1)} - \eta \frac{\partial Q(w_0, w_1)}{w_0}$

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

E(z) = 0 扰动项的期望为0

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

COV(x,z) = E(xz) - E(x)E(z) = E(xz) = 0

x与z线性不相关

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

找到w_0和w_1使得z^2取最小值

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

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

Linear Regression

- Least-squares solutions

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

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

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

使用回归模型预测分类的缺点：回归趋向于预测为整个数据集的均值

Logistic Regression 用于预测分类

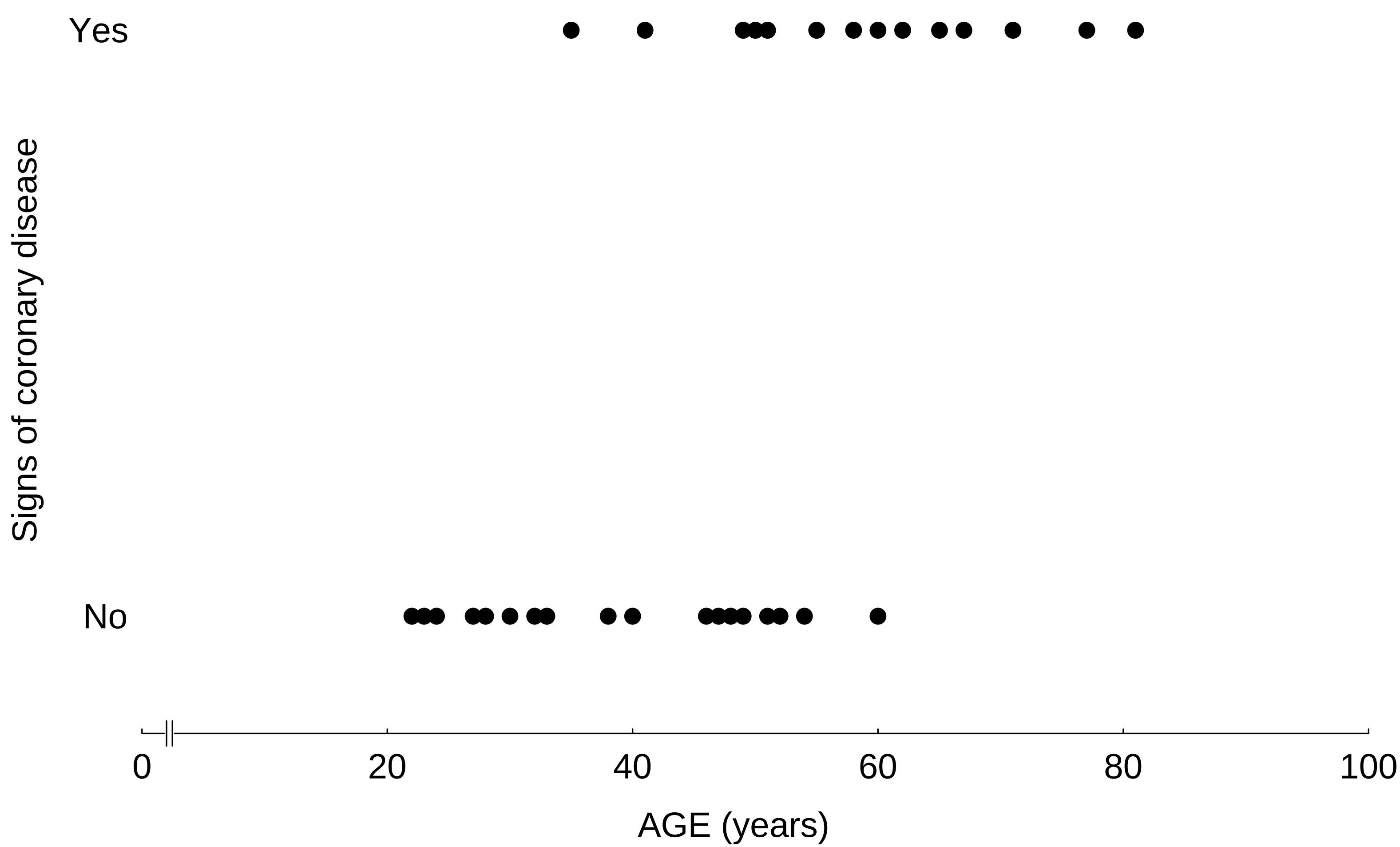
- We may use the linear regression model for binary classification 将线性回归模型用于二元分类

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

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

  - However, the predicted y values (预测的y值) can be greater than 1 or less than 0

Logistic Regression



Logistic Regression

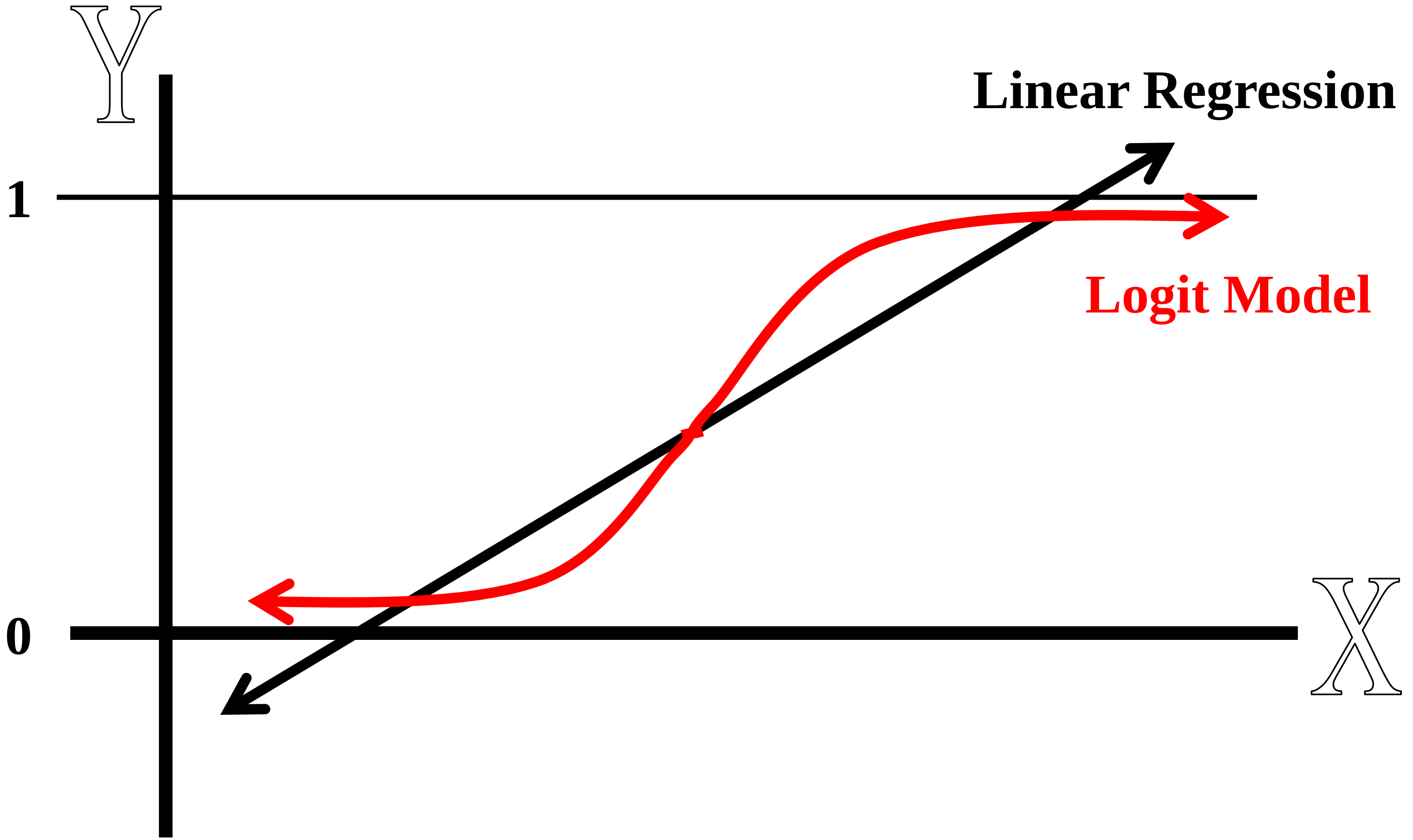
- The "logit" model solves the above problem:

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

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

- p is the probability that the event y occurs, $p(y=1 \mid \mathbf{X})$
- $p/(1-p)$ is the odds ratio (*e.g.*, odds of disease)
- $\log[p/(1-p)]$ is the log odds ratio, or "logit"

Logistic Regression



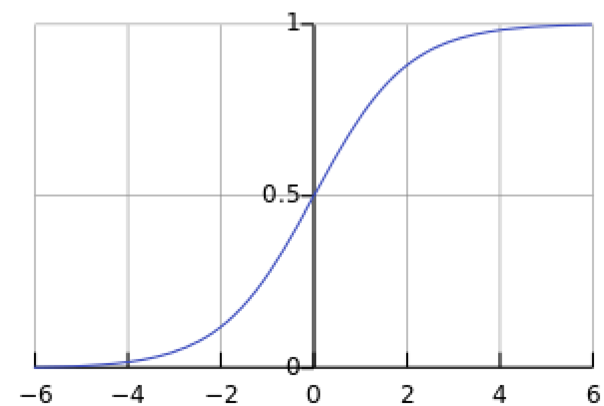
Logistic Regression

逻辑分布将估计概率限制在0和1之间

- The logistic distribution constrains the estimated probabilities to lie between 0 and 1.
- The estimated probability $p(y=1|\mathbf{X})$ is:

$$p=\frac{1}{1+e^{-w_0-\sum_{j=1}^{d}w_jx_j}}=\frac{e^{w_0+\sum_{j=1}^{d}w_jx_j}}{1+e^{w_0+\sum_{j=1}^{d}w_jx_j}}$$

$$=\frac{1}{1+e^{-\tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{x}}}}=\frac{e^{\tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{x}}}}{1+e^{\tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{x}}}}$$



- if you let $w_0+\sum_{j=1}^{d}w_jx_j=0$, then $p=0.5$
- as $w_0+\sum_{j=1}^{d}w_jx_j$ gets really big, p approaches 1
- as $w_0+\sum_{j=1}^{d}w_jx_j$ gets really small, p approaches 0

Logistic Regression

使用迭代极大似然法求解逻辑回归模型

- The Logistic Regression model will be solved by an **iterative maximum likelihood** procedure.
- This is a computer dependent program that:
  从回归系数的任意值开始，建立预测观测数据的初始模型
  - starts with arbitrary values of the regression coefficients and constructs an initial model for predicting the observed data.
  评估预测中的误差并改变回归系数，以便在新模型下使观测数据的似然度更大
  - then evaluates errors in such prediction and changes the regression coefficients so as make the likelihood of the observed data greater under the new model.
  不断重复直到模型收敛
  - repeats until the model converges, meaning the differences between the newest model and the previous model are trivial.
- The idea is that you "find and report as statistics" the parameters that are most likely to have produced your data.

Logistic Regression

- The likelihood function is $\prod_{i=1}^{n}(p_i)^{y_i}(1-p_i)^{1-y_i}$
- We want to maximize the log likelihood:

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

L(W)<0

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

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

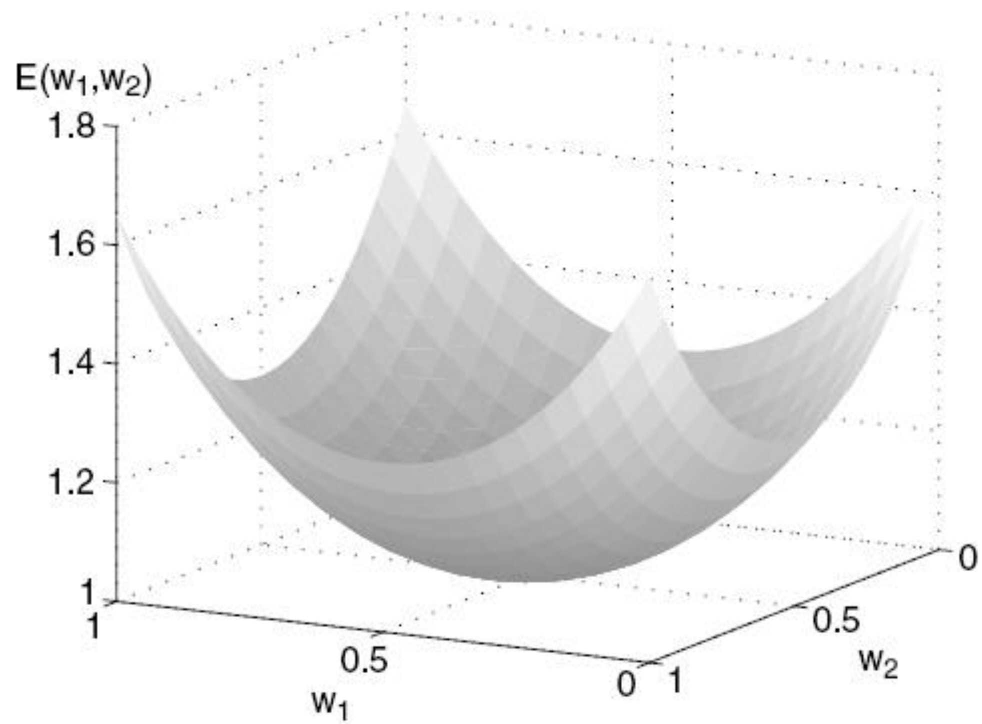
$$\frac{\partial L(\tilde{\mathbf{W}})}{\partial \tilde{\mathbf{W}}} = \sum_{i=1}^{n}\left[\left(y_i - \frac{e^{\tilde{\mathbf{W}}^{\mathrm{T}}\tilde{\mathbf{X}}_i}}{1+e^{\tilde{\mathbf{W}}^{\mathrm{T}}\tilde{\mathbf{X}}_i}}\right)\tilde{\mathbf{X}}_i\right]$$

- It is equal to minimize the cost function

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

Cross-entropy

Gradient Decent



Logistic Regression

n的设置需要根据训练集选择，迭代1次后比较L(W)的变化情况，再调整n的值

- Gradient Decent (梯度下降)
  - Calculate the gradient vector 梯度向量
  - Update the weighting in the opposite direction of the gradient vector at each surface point
- Repeat: $\tilde{\mathbf{W}}_{new}^{(j)} = \tilde{\mathbf{W}}^{(j)} - \eta \dfrac{\partial C(\tilde{\mathbf{W}})}{\partial \tilde{\mathbf{W}}^{(j)}}$ 相当于斜率

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

- Until convergence

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

Gradient Decent