LiVE Group
视觉智能与学习研究中心

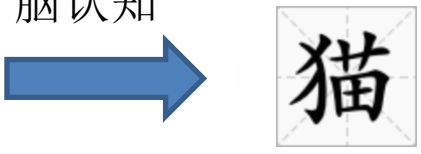
机器学习（第6讲）

主讲：张 磊

E-mail: leizhang@cqu.edu.cn
Lab Website: http://www.leizhang.tk

第六章：概率建模

脑认知



□产生式思考

对于任意一个样本，我们定义模型如下：

$$t_n = \mathbf{w}^T \mathbf{x}_n + \varepsilon_n$$

其中，ε_n是一个随机变量。所谓产生式考虑，是指第n次的观测响应，是通过$\mathbf{w}^T \mathbf{x}_n$生成。同时，伴随着一个随机变量，可正可负。

ε_n是一个随机变量，存在某种概率分布。

我们假设，$\varepsilon_n \sim \mathcal{N}(\mu, \sigma^2)$高斯分布（正态分布）

$$p(\varepsilon_n) = \mathcal{N}(\mu, \sigma^2)$$

该假设已在前面的课程给出解释（中心极限定理、易于分析）。

□产生式思考

对于任意一个样本，我们定义模型如下：

$$t_n = \mathbf{w}^T \mathbf{x}_n + \varepsilon_n$$

现在，模型由两部分组成：

1. 数据生成的决定性部分，$\mathbf{w}^T \mathbf{x}_n$，表示一种趋势或倾向；
2. 随机组成部分ε_n，可以理解成噪声。

注：这里的随机变量ε_n（或噪声）并非限定为高斯分布，也并非限定为可加性噪声。但为了便于解释和理解，选择可加性高斯分布噪声，可使我们获得最优参数$\hat{\mathbf{w}}$的精确表达式。

□产生式思考

将模型重新定义如下：

$$t_n = \mathbf{w}^T\mathbf{x}_n + \varepsilon_n, \text{其中，} \varepsilon_n \sim \mathcal{N}(\mu, \sigma^2)$$

这里令μ=0，即$\varepsilon_n \sim \mathcal{N}(0, \sigma^2)$

由于$\mathbf{w}^T\mathbf{x}_n$是一个常量，因此加上一个随机变量ε_n后，t_n也是一个随机变量（**这是赋予随机变量的原因**）。

而且，$t_n \sim \mathcal{N}(\mathbf{w}^T\mathbf{x}_n, \sigma^2)$（为什么？高斯随机变量加上一个常数，依然是高斯，仅均值发生变化），从而概率密度函数可以写成：

$$p(t_n|\mathbf{x}_n, \mathbf{w}, \sigma^2) = \mathcal{N}(\mathbf{w}^T\mathbf{x}_n, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(t_n - \mathbf{w}^T\mathbf{x}_n)^2}{2\sigma^2}}$$

可以看出，预测响应t_n的密度依赖于特定的$\mathbf{x}_n$,$\mathbf{w}$ (均值)和σ^2(方差)的值。

问题

我们看下面这个概率密度函数

$$p\left(t_n|\mathbf{x}_n,\mathbf{w},\sigma^2\right)=\mathcal{N}\left(\mathbf{w}^T\mathbf{x}_n,\sigma^2\right)$$

对于给定的$\mathbf{w},\sigma^2$，我们可以计算出在观测$\mathbf{x}_n$下，使得$p\left(t_n|\mathbf{x}_n,\mathbf{w},\sigma^2\right)$最大的$t_n$，即预测值。

对于已知的高斯分布来说，只要均值和方差确定，即可绘制出连续或离散的概率密度。

例：设第10个苹果的直径为0.5，即$x_{10}=[0.5,1]^T$；设$w=[0.6,-0.1]^T$, $\sigma^2=0.1$；另：该苹果实际重量$t_{10}=2$.

分析：针对该苹果，可以绘制其重量的概率密度函数曲线

问题

✓ 如果$\mathbf{w}, \sigma^2$已经最优

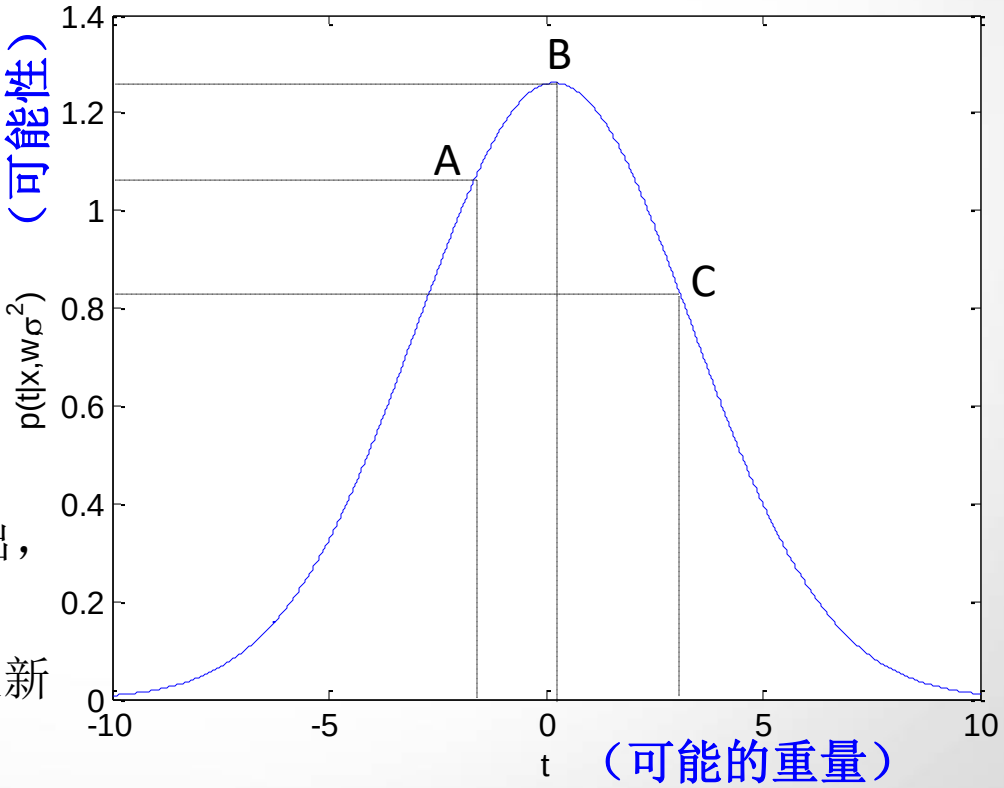
通过右边的概率密度函数可以得出：

1. B点的可能性最大，即为苹果重量的最佳估计值$\mathbf{w}^T\mathbf{x}_{10}$
2. C点的可能性最小。

✓ 如果$\mathbf{w}, \sigma^2$还不是最优

实际重量t_{10}=2，根据图可以看出，最佳的可能值是在B和C之间。

但是**数据不可能变**，因此只能更新模型参数。

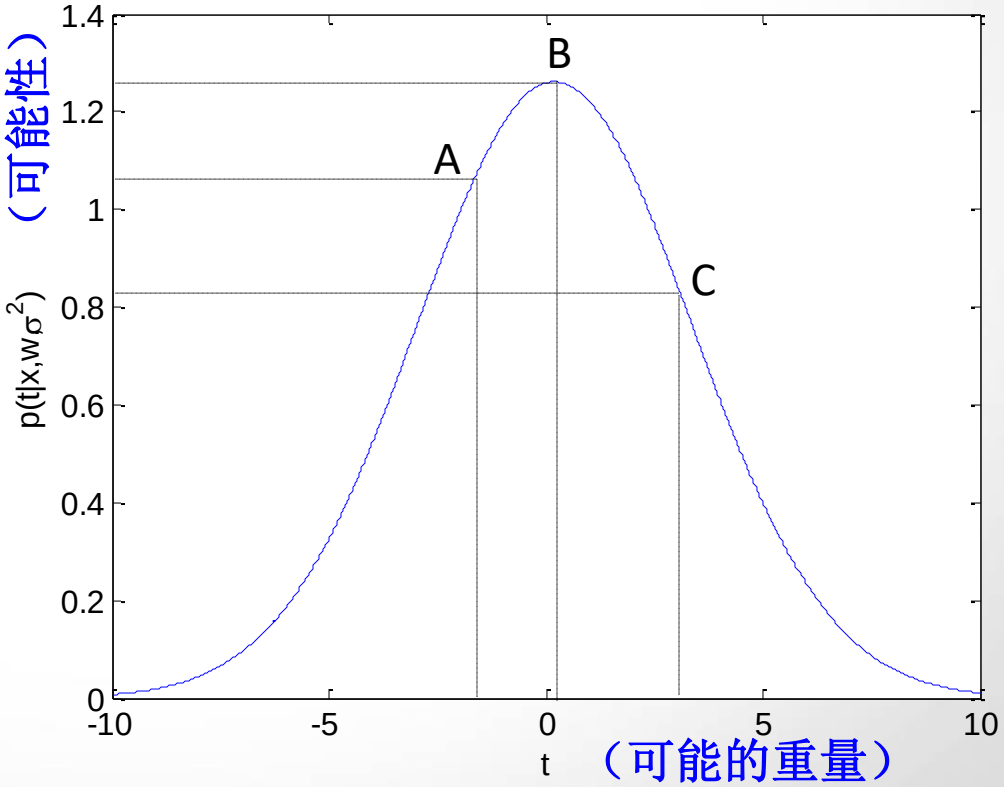


定义

这种通过寻找参数，来最大化似然值的方式，称为基于似然函数最大化的线性建模——**概率建模**，是机器学习中的一种重要方法。

数据集的似然值

对机器学习，我们感兴趣的是整个数据集的**所有样本的似然值**，而并非单个样本的似然值。



数据集的似然值

- 对于单个样本的概率密度表达式为

$$p\left(t_n|\mathbf{x}_n,\mathbf{w},\sigma^2\right)=\mathcal{N}\left(\mathbf{w}^T\mathbf{x}_n,\sigma^2\right)$$

- 如果有N个数据样本点，那么我们感兴趣的是它们的联合条件概率密度

$$p\left(t_1,t_2,\ldots,t_N|\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N,\mathbf{w},\sigma^2\right)$$

- 根据向量和矩阵表示法，可以写成紧缩的表达形式：

$$p\left(\mathbf{t}|\mathbf{X},\mathbf{w},\sigma^2\right)$$

- 假设数据的噪声是独立的，即噪声的联合概率密度可表达为

$$p(\boldsymbol{\varepsilon})=p(\varepsilon_1,\varepsilon_2,\ldots,\varepsilon_N)=\prod_{n=1}^{N}p(\varepsilon_n)$$

这一独立性假设，可以使联合密度进行分解为更易操作的对象，使问题求解更简单

数据集的似然值

- 基于噪声的独立性假设，对N个数据样本点，它们的联合条件概率密度

$$p(\mathbf{t}|\mathbf{X},\mathbf{w},\sigma^2)=p(t_1,t_2,\dots,t_N|\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_N,\mathbf{w},\sigma^2)=\prod_{n=1}^{N}p(t_n|\mathbf{x}_n,\mathbf{w},\sigma^2)$$

$$=\prod_{n=1}^{N}\mathcal{N}(\mathbf{w}^T\mathbf{x}_n,\sigma^2)$$

注：这里不是说$\mathbf{t}$是独立的，否则不存在对数据的建模。而是条件独立，可以根据模型$t_n=\mathbf{w}^T\mathbf{x}_n+\varepsilon_n$理解。当$\mathbf{w}^T\mathbf{x}_n$给定时，$t_n$独立，即条件独立。
或者说，只有当模型确定了，t_n才是一个独立的随机变量。

另：$\prod_{n=1}^{N}$ 表示连乘符号。

最大似然建模

- 最大似然建模，就是为了寻找$\mathbf{w}$和σ^2，使得似然值最大化。也就是最大化下列似然函数。

$$\max_{\mathbf{w},\sigma^2} p\left(\mathbf{t}|\mathbf{X},\mathbf{w},\sigma^2\right) = p\left(t_1,t_2,\dots,t_N|\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_N,\mathbf{w},\sigma^2\right)$$

$$\propto \max_{\mathbf{w},\sigma^2} \prod_{n=1}^{N} p(t_n|\mathbf{x}_n,\mathbf{w},\sigma^2) = \prod_{n=1}^{N} \mathcal{N}\left(\mathbf{w}^T\mathbf{x}_n,\sigma^2\right)$$

换句话说，当给定$\mathbf{w}$和$\boldsymbol{\sigma^2}$时，可以获得数据集的似然值$L = p\left(\mathbf{t}|\mathbf{X},\mathbf{w},\sigma^2\right)$，但是，该似然值并非最大，因此，我们要改变$\mathbf{w}$和$\boldsymbol{\sigma^2}$使得$L$最大。

当然，为什么不改变X呢？数据！！

因为数据集是固定的，不同的参数值会产生不同似然值，建模的目的是要求出最佳的参数值。

最大似然建模

- 最大似然建模求解：即求解$\mathbf{w}$和σ^2，使得似然值最大化。

$$\max_{\mathbf{w},\sigma^2} \prod_{n=1}^{N} \mathcal{N}\left(\mathbf{w}^T\mathbf{x}_n, \sigma^2\right)$$

很显然，要求解上面的模型，可以最大化似然值的自然对数，用log表示，即，

$$\max_{\mathbf{w},\sigma^2} \prod_{n=1}^{N} \mathcal{N}\left(\mathbf{w}^T\mathbf{x}_n, \sigma^2\right) \propto \max_{\mathbf{w},\sigma^2} log \prod_{n=1}^{N} \mathcal{N}\left(\mathbf{w}^T\mathbf{x}_n, \sigma^2\right)$$

- 似然函数的自然对数可以简化

$$log \prod_{n=1}^{N} \mathcal{N}\left(\mathbf{w}^T\mathbf{x}_n, \sigma^2\right) = log\mathcal{N}\left(\mathbf{w}^T\mathbf{x}_1, \sigma^2\right) + \cdots + log\mathcal{N}\left(\mathbf{w}^T\mathbf{x}_N, \sigma^2\right)$$

$$= \sum_{n=1}^{N} log\mathcal{N}\left(\mathbf{w}^T\mathbf{x}_n, \sigma^2\right)$$

- 最大似然建模
  - 似然函数的自然对数可以简化

$$logL = log\prod_{n=1}^{N}\mathcal{N}\left(\mathbf{w}^T\mathbf{x}_n,\sigma^2\right) = \sum_{n=1}^{N} log\mathcal{N}\left(\mathbf{w}^T\mathbf{x}_n,\sigma^2\right)$$

$$= \sum_{n=1}^{N} log\left(\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{\left(t_n-\mathbf{w}^T\mathbf{x}_n\right)^2}{2\sigma^2}}\right)$$

$$= \sum_{n=1}^{N}\left(-\frac{1}{2}log(2\pi) - log\sigma - \frac{1}{2\sigma^2}\left(t_n-\mathbf{w}^T\mathbf{x}_n\right)^2\right)$$

$$= -\frac{N}{2}log(2\pi) - Nlog\sigma - \frac{1}{2\sigma^2}\sum_{n=1}^{N}\left(t_n-\mathbf{w}^T\mathbf{x}_n\right)^2$$

◘ 最大似然建模

似然函数的自然对数最大化模型可以简化

$$\max_{\mathbf{w},\sigma^2} \log L = -\frac{N}{2} - \log(2\pi) - N\log\sigma - \frac{1}{2\sigma^2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\mathbf{x}_n\right)^2$$

➢ 利用前面学习的内容，对w求偏导，

$$\frac{\partial \log L}{\partial \mathbf{w}} = \frac{1}{\sigma^2}\sum_{n=1}^{N}\mathbf{x}_n\left(t_n - \mathbf{x}_n^T\mathbf{w}\right) = \frac{1}{\sigma^2}\sum_{n=1}^{N}\mathbf{x}_n t_n - \frac{1}{\sigma^2}\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^T\mathbf{w}$$

$$= \frac{1}{\sigma^2}\left(\mathbf{X}^T\mathbf{t} - \mathbf{X}^T\mathbf{X}\mathbf{w}\right)$$

令$\frac{\partial \log L}{\partial \mathbf{w}} = \mathbf{0}$，可以得出$\hat{\mathbf{w}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{t}$（w的最大似然解） *似曾相识！？*

将噪声假设为高斯分布，最小化平方损失等同于最大似然解。

◘ 最大似然建模

似然函数的自然对数最大化模型可以简化

$$\max_{\mathbf{w},\sigma^2} \log L = -\frac{N}{2} - \log(2\pi) - N\log\sigma - \frac{1}{2\sigma^2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\mathbf{x}_n\right)^2$$

➢ 对σ求偏导，

$$\frac{\partial \log L}{\partial \sigma} = -\frac{N}{\sigma} + \frac{1}{\sigma^3}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\mathbf{x}_n\right)^2$$

令$\frac{\partial \log L}{\partial \sigma} = 0$，可以得出$\widehat{\sigma^2} = \frac{1}{N}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\mathbf{x}_n\right)^2 = \frac{1}{N}(\mathbf{t} - \mathbf{X}\mathbf{w})^T(\mathbf{t} - \mathbf{X}\mathbf{w})$

最终，$\widehat{\sigma^2} = \frac{1}{N}\left(\boldsymbol{t}^{\mathrm{T}}\boldsymbol{t} - \boldsymbol{t}^{\mathrm{T}}\mathbf{X}\mathbf{w}\right)$

课外任务：

根据前面的内容，请自行推导最大似然建模方法的整个过程。

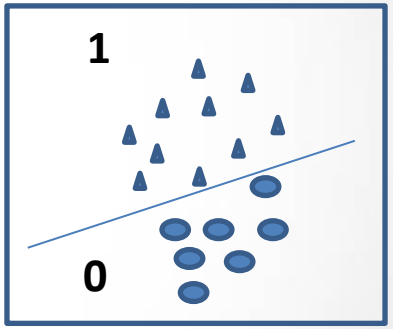
Logistic回归

回顾线性回归算法中的二类分类问题

$$h(\mathbf{x}) = \mathbf{w_0} + \mathbf{w_1 x_1} + \cdots + \mathbf{w}_d \mathbf{x_d}$$

分类边界为

$$h(\mathbf{x}) = 0$$



- 当$h(\mathbf{x}) > 0$, x的属性为Y=1
- 当$h(\mathbf{x}) < 0$, x的属性为Y=0

也就是说，对于一次观测x，其仅仅给出“是”或者“不是”的猜测（*简单、粗暴*），给出的预测是一个“肯定发生”或“肯定不发生”的事件。

如果提供一个关于可能性大小的结果岂不是更好？？换句话说，“可能性”是关于概率的一个概念，如果对于一个观测，能够计算出该观测的事件发生的概率（即Y=1发生的概率），通常比直接给出“是”或者“不是”更加有用。

Logistic回归

几个特点：

- Logistic回归模型作为一种概率模型，可用于预测某事件发生的概率；
- Logistic回归模型不要求因变量在正态分布的前提假设下完成。
- 该模型在疾病诊断、预测中应用较广，主要用于分析不同变量因素对疾病的影响。在医学领域，一些随机事件只具有两种互斥结果的离散性随机事件。这类只具有两种互斥结果的离散型随机事件的规律性进行描述的一种概率分布，叫二项分布。
- 二项分布概率公式：如果进行n次伯努利试验，取得成功次数(事件发生)为k(k=0,1,…,n)的概率，表示为

$$P(\xi = k) = C_n^k p^k (1-p)^{n-k}$$

其中，事件发生的概率为p, 事件不发生的概率为1-p。$\xi \sim B(n,p)$，当n=1时，二项分布也可称为伯努利分布或0-1分布（1次伯努利试验）

Logistic回归

条件概率：

对于观测输入变量X, 其响应Y的条件概率分布表达为

$$\boldsymbol{P}(Y|X)$$

这个概率表示模型的准确性。

设想，如果对于“今天是否下雪”的预测结果是51%，但是并没有下雪。那么这个预测也要比预测为下雪的可能性为99%要好！

设$P(Y=1|X=x)=p(x;\boldsymbol{w})$,其中w为参数，那么有

$$P(Y=0|X=x)=1-p(\boldsymbol{x};\boldsymbol{w})$$

Y的取值为0或1

◘ Logistic回归

条件概率：

对于一个观测x_i,其响应$Y = y_i$的条件概率可以写成

$$P(Y = y_i|X = x_i) = p(x_i; \boldsymbol{w})^{y_i}(1 - p(x_i; \boldsymbol{w}))^{1-y_i}$$

由于N次观测的独立性，联合条件概率分布(似然)可以写为

$$\prod_{i=1}^{N} P(Y = y_i|X = x_i) = \prod_{i=1}^{N} p(x_i; \boldsymbol{w})^{y_i}(1 - p(x_i; \boldsymbol{w}))^{1-y_i}$$

N个观测的似然函数

Y的取值为0或1

Logistic回归—与线性回归的关系

主要思想：logistic回归模型的输出是某次观测事件发生的概率，通过观测样本x的特征与最佳估计参数相乘、求和后，然后利用Logistic函数（sigmoid）进行非线性计算（概率），然后与阈值进行比较，得出该观测x的输出。

相乘、求和：

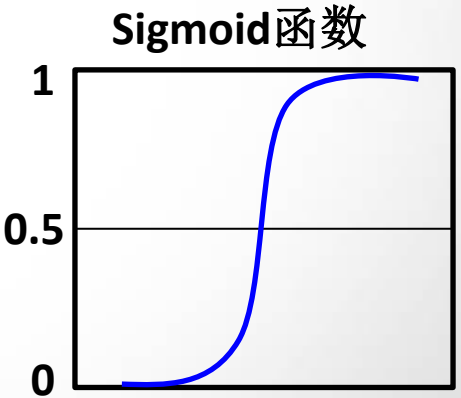
$$h(\mathbf{x}) = \mathbf{w_0} + \mathbf{w_1 x_1} + \cdots + \mathbf{w}_d \mathbf{x_d}$$

条件概率$\boldsymbol{P}(\mathrm{y} = \mathbf{1}|\boldsymbol{x})$的计算表达式为

$$\boldsymbol{P}(\mathrm{y} = \mathbf{1}|\boldsymbol{x}) = \boldsymbol{p}(\boldsymbol{x}) = \frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{e}^{-h(x)}}$$

条件概率$\boldsymbol{P}(\mathrm{y} = \mathbf{0}|\boldsymbol{x})$的计算表达式为

$$\boldsymbol{P}(\mathrm{y} = \mathbf{0}|\boldsymbol{x}) = \mathbf{1} - \boldsymbol{P}(\boldsymbol{y} = \mathbf{1}|\boldsymbol{x}) = \mathbf{1} - \boldsymbol{p}(\boldsymbol{x}) = \frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{e}^{h(x)}}$$



定义域（-inf, inf）；值域为（0，1）

Logistic回归—与线性回归的关系

$P(y=1|x)=\frac{1}{1+e^{-h(x)}}$为事件发生的概率;

$P(y=0|x)=\frac{1}{1+e^{h(x)}}$为事件不发生的概率;

那么，事件发生与事件不发生的概率比值为

$$\frac{P(y=1|x)}{P(y=0|x)}=\frac{p(x)}{1-p(x)}=\frac{1+e^{h(x)}}{1+e^{-h(x)}}=e^{h(x)}$$

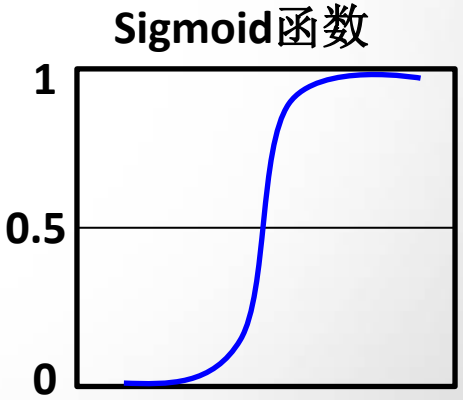
两边同时取对数，

$$log\frac{p(x)}{1-p(x)}=h(x)=w_0+w_1x_1+\cdots+w_dx_d$$

得到线性函数（即Logistic 回归模型）。

利用上式，可以求出$p(x)=\frac{e^{h(x)}}{1+e^{h(x)}}=\frac{1}{1+e^{-h(x)}}$介于（0,1）。

采用Logit变换，建立与属性变量之间的关系。当0<p<1时，logit变换可取任意值



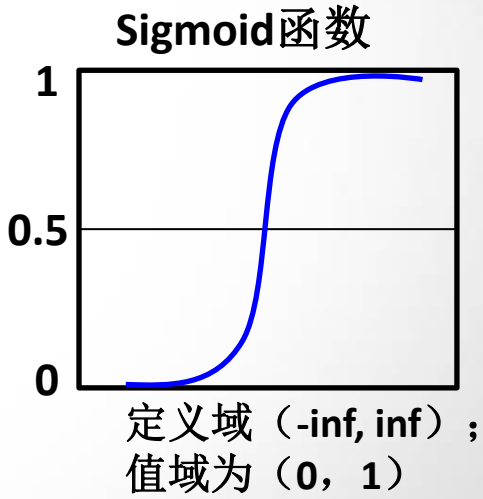
- Logistic回归—与线性回归的关系
- $p(x)=\frac{e^{h(x)}}{1+e^{h(x)}}=\frac{1}{1+e^{-h(x)}}$ 介于（0,1）。

根据观测样本x的概率表示可以看出，当$p(x)>0.5$, Y=1;
当$p(x)<0.5$, Y=0。

- 这也同时意味着，$h(x)>0$时，Y=1；$h(x)<0$时，Y=0。

即，将两类分开的判决边界为$h(x)=0$。

- 但Logistic回归不同于线性分类器，不是为了找分类边界，而是找出一个概率值，对“可能性”进行建模。



在Logistic回归模型中，若要计算p(x;w)，关键是要求出参数w。

Logistic回归

最大似然函数：

对于N次观测，似然函数可以写为

$$\prod_{i=1}^{N} P(Y = y_i|X = x_i) = \prod_{i=1}^{N} p(x_i; \boldsymbol{w})^{y_i}(1 - p(x_i; \boldsymbol{w}))^{1-y_i}$$

注：N次观测的独立性。w是模型参数。

我们的目标是能够求出使这一似然函数取最大值时的参数估计w。

$$\hat{\boldsymbol{w}} = \max_{\boldsymbol{w}} \prod_{i=1}^{N} P(Y = y_i|X = x_i) = \prod_{i=1}^{N} p(x_i; \boldsymbol{w})^{y_i}(1 - p(x_i; \boldsymbol{w}))^{1-y_i}$$

接下来，如何解这个模型是关键。

◘ Logistic回归

最大似然函数：

对似然函数取自然对数，可得出对数似然函数L(w)，

$$L(\boldsymbol{w}) = \sum_{i=1}^{N}[y_i \log p(x_i;\boldsymbol{w}) + (1-y_i)\log(1-p(x_i;\boldsymbol{w}))]$$

$$= \sum_{i=1}^{N}\left[\log(1-p(x_i;\boldsymbol{w})) + y_i \log\frac{p(x_i;\boldsymbol{w})}{1-p(x_i;\boldsymbol{w})}\right]$$

$$= \sum_{i=1}^{N}\left[\log\left(1-\frac{1}{1+e^{-h(x)}}\right) + y_i h(x)\right]$$

$$= \sum_{i=1}^{N}\left[-\log\left(1+e^{h(x)}\right) + y_i h(x)\right]$$

$$= \sum_{i=1}^{N}\left[-\log\left(1+e^{w_0+\mathbf{w}^T\mathbf{x}_i}\right)\right] + \sum_{i=1}^{N} y_i\left(w_0+\mathbf{w}^T\mathbf{x}_i\right)$$

为了估计能使L(w)取得最大值的参数w=[w_0,w_1,…,w_d],对函数求导。

Logistic回归

最大似然函数：

化简后的对数似然函数L(w)，

$$L(\boldsymbol{w}) = \sum_{i=1}^{N}\left[-\log\left(1+e^{w_0+\mathbf{w}^T\mathbf{x}_i}\right)\right] + \sum_{i=1}^{N} y_i\left(w_0+\mathbf{w}^T\mathbf{x}_i\right)$$

为了估计能使L(w)取得最大值的参数$w=[w_0,w_1,\ldots,w_d]$,分别求函数对$w_0,w_1,\ldots,w_d$的导数，显然存在d+1个方程。

$$\frac{\partial L(w)}{\partial w_j} = -\sum_{i=1}^{N}\frac{1}{1+e^{w_0+\mathbf{w}^T\mathbf{x}_i}}\cdot e^{w_0+\mathbf{w}^T\mathbf{x}_i}\cdot x_{ij} + \sum_{i=1}^{N} y_i x_{ij}$$

$$= \sum_{i=1}^{N}(y_i - p(x_i;\boldsymbol{w}))x_{ij}$$

注：此时导数表达式是一个超越方程，不存在闭合解（解析解、精确解）。*怎么办？*

超越方程：非多项式函数，如指数函数、对数函数、三角函数等

Logistic回归

Newton-Raphson（牛顿-拉斐森）优化算法：

我们考虑一个单变量函数的情况，即$f(\beta)$,求该函数的最优解β^*。我们假设函数f(.)是平滑的，那么意味着该函数在β^*处的导数为0，二阶导数大于0（极值理论）。

那么根据泰勒展开，$f(\beta)$的表达式可以写成

$$f(\beta) \approx f(\beta^*) + \frac{1}{2}(\beta - \beta^*)^2 \frac{d^2 f}{d\beta^2}|_{\beta=\beta^*}$$

既然β^*是最小值点，那么$f(\beta) > f(\beta^*)$,因此，$\frac{d^2 f}{d\beta^2}|_{\beta=\beta^*} > 0$。

而且，根据上式，$\frac{df}{d\beta}|_{\beta=\beta^*} = 0$

换句话说，在最优解附近，$f(\beta)$是一个近二次的函数表达式。

Logistic回归

Newton-Raphson（牛顿-拉斐森）优化算法：

既然在最优解附近，$f(\beta)$是一个近二次的函数表达式，而我们的目的是求出这个最优解，因此，可以将$f(\beta)$进行二次泰勒展开（为什么不更高次？？极值定理）

首先，猜测一个初始解$\beta^{(0)}$,在该点，二次泰勒展开：

$$f(\beta) \approx f(\beta^{(0)}) + (\beta - \beta^{(0)}) \frac{df}{d\beta}|_{\beta=\beta^{(0)}} + \frac{1}{2}(\beta - \beta^{(0)})^2 \frac{d^2 f}{d\beta^2}|_{\beta=\beta^{(0)}}$$

为了方便，上式可以重新写成，

$$f(\beta) \approx f(\beta^{(0)}) + (\beta - \beta^{(0)}) f'(\beta^{(0)}) + \frac{1}{2}(\beta - \beta^{(0)})^2 f''(\beta^{(0)})$$

此时，$f(\beta)$是关于β的二次函数，可导，因此便可以利用导数=0求解。

Logistic回归

Newton-Raphson（牛顿-拉斐森）优化算法：

$$f(\beta) \approx f\left(\beta^{(0)}\right) + \left(\beta - \beta^{(0)}\right) f'\left(\beta^{(0)}\right) + \frac{1}{2}\left(\beta - \beta^{(0)}\right)^2 f''\left(\beta^{(0)}\right)$$

若要求函数$f(\beta)$的最小值，我们令

$$\frac{df(\beta)}{d\beta} = f'\left(\beta^{(0)}\right) + f''\left(\beta^{(0)}\right)\left(\beta - \beta^{(0)}\right) = 0$$

从而可以计算出

$$\beta = \beta^{(0)} - \frac{f'\left(\beta^{(0)}\right)}{f''\left(\beta^{(0)}\right)}$$

令此时的最优解为$\beta = \beta^{(1)}$（并非最优）

$$\beta^{(1)} = \beta^{(0)} - \frac{f'\left(\beta^{(0)}\right)}{f''\left(\beta^{(0)}\right)}$$

Logistic回归

Newton-Raphson（牛顿-拉斐森）优化算法：

当计算出新的次优解$\beta^{(1)}$，函数$f(\beta)$的泰勒展开为

$$f(\beta) \approx f\left(\beta^{(1)}\right) + \left(\beta - \beta^{(1)}\right) f'\left(\beta^{(1)}\right) + \frac{1}{2}\left(\beta - \beta^{(1)}\right)^2 f''\left(\beta^{(1)}\right)$$

若要求函数$f(\beta)$的最小值，我们令

$$\frac{df(\beta)}{d\beta} = f'\left(\beta^{(1)}\right) + f''\left(\beta^{(1)}\right)\left(\beta - \beta^{(1)}\right) = 0$$

从而可以计算出

$$\beta = \beta^{(1)} - \frac{f'\left(\beta^{(1)}\right)}{f''\left(\beta^{(1)}\right)}$$

令此时的最优解为$\beta = \beta^{(2)}$（并非最优）

$$\beta^{(2)} = \beta^{(1)} - \frac{f'\left(\beta^{(1)}\right)}{f''\left(\beta^{(1)}\right)}$$

Logistic回归

Newton-Raphson（牛顿-拉斐森）优化算法：

为了找到最优解，通常迭代多次，设为T次。

当计算出新的次优解$\beta^{(t)}$，函数$f(\beta)$的泰勒展开为

$$f(\beta) \approx f\left(\beta^{(t)}\right) + \left(\beta - \beta^{(t)}\right) f'\left(\beta^{(t)}\right) + \frac{1}{2}\left(\beta - \beta^{(t)}\right)^2 f''\left(\beta^{(t)}\right)$$

若要求函数$f(\beta)$的最小值，我们令

$$\frac{df(\beta)}{d\beta} = f'\left(\beta^{(t)}\right) + f''\left(\beta^{(t)}\right)\left(\beta - \beta^{(t)}\right) = 0$$

从而可以计算出

$$\beta = \beta^{(t)} - \frac{f'\left(\beta^{(t)}\right)}{f''\left(\beta^{(t)}\right)}$$

令此时的最优解为$\beta = \beta^{(t+1)}$（并非最优）

解更新的迭代公式

$$\beta^{(t+1)} = \beta^{(t)} - \frac{f'\left(\beta^{(t)}\right)}{f''\left(\beta^{(t)}\right)}$$

Logistic回归

Newton-Raphson（牛顿-拉斐森）优化算法 （多个变量的情况——多维）：

一维时，解更新的迭代公式

$$\beta^{(t+1)} = \beta^{(t)} - \frac{f'(\beta^{(t)})}{f''(\beta^{(t)})}$$

对于多维时，模型由$f(\beta) = \beta_0 + \beta_1 x$转化为$f(\beta) = \beta_0 + \beta_1 x_1 + \cdots + \beta_d x_d$

那么如何计算多维的函数关于变量的一阶导数和二阶导数？？

◘ Logistic回归

Newton-Raphson（牛顿-拉斐森）优化算法 （多个变量的情况——多维）：

对于多维时，模型由$f(\beta)=\beta_0+\beta_1 x$转化为$f(\beta)=\beta_0+\beta_1 x_1+\cdots+\beta_d x_d$

回顾第三章（数学基础）中，关于对矩阵或向量的导数。

$$f'\left(\beta^{(t)}\right)=\nabla f\left(\beta^{(t)}\right)=\left[\frac{\partial f}{\partial \beta_1},\frac{\partial f}{\partial \beta_2},\cdots,\frac{\partial f}{\partial \beta_d}\right]^{\mathrm{T}}$$

$$f''\left(\beta^{(t)}\right)=\nabla^2 f\left(\beta^{(t)}\right)=\begin{bmatrix}\frac{\partial^2 f}{\partial \beta_1^2} & \cdots & \frac{\partial f}{\partial \beta_1 \partial \beta_d}\\ \vdots & \ddots & \vdots\\ \frac{\partial f}{\partial \beta_d \partial \beta_1} & \cdots & \frac{\partial^2 f}{\partial \beta_d^2}\end{bmatrix}\Big|_{\beta=\beta^{(t)}}=H\ (Hessian矩阵)$$

Logistic回归

Newton-Raphson（牛顿-拉斐森）优化算法（多个变量的情况——多维）：

对于多维时，函数$f(\beta)$的解更新优化方法为：

$$\beta^{(t+1)} = \beta^{(t)} - \frac{f'(\beta^{(t)})}{f''(\beta^{(t)})}$$

$$\beta^{(t+1)} = \beta^{(t)} - H^{-1}(\beta^{(t)})\nabla f(\beta^{(t)})$$

迭代方法完全相同，只是在写法上，变成了矩阵向量的方式。

知识回顾（多元函数）

多元函数f(x1, x2, ...,xn)在$X^{(0)}$处的泰勒展开：

$$f(X) = f(X^{(0)}) + \nabla f(X^{(0)})^T \Delta X + \frac{1}{2}\Delta X^T G(X^{(0)})\Delta X + \cdots$$

其中，

（1） $\nabla f\left(X^{(0)}\right) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_n}\right]\Big|_{X^{(0)}}^T$ 一阶梯度向量

（2） $G(X^{(0)}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}_{X^{(0)}}$ 海塞矩阵(Hessian Matrix，二阶梯度)

Logistic回归

最大似然函数：

化简后的对数似然函数L(w)，

$$L(\boldsymbol{w}) = \sum_{i=1}^{N}\left[-\log\left(1+e^{w_0+\mathbf{w}^T\mathbf{x}_i}\right)\right] + \sum_{i=1}^{N} y_i\left(w_0+\mathbf{w}^T\mathbf{x}_i\right)$$

为了估计能使L(w)取得最大值的参数$w=[w_0,w_1,\ldots,w_d]$,分别求函数对$w_0,w_1,\ldots,w_d$的导数，显然存在d+1个方程。

$$\frac{\partial L(w)}{\partial w_j} = -\sum_{i=1}^{N}\frac{1}{1+e^{w_0+\mathbf{w}^T\mathbf{x}_i}}\cdot e^{w_0+\mathbf{w}^T\mathbf{x}_i}\cdot x_{ij} + \sum_{i=1}^{N} y_i x_{ij}$$

$$= \sum_{i=1}^{N}(y_i - p(x_i;\boldsymbol{w}))x_{ij}$$

此时导数表达式是一个超越方程，不存在闭合解。采用牛顿-拉斐森方法求解。

Logistic回归

$$p(x) = \frac{e^{h(x)}}{1+e^{h(x)}}$$

最大似然函数：

Hessian矩阵H的计算：

$$\frac{\partial^2 L(w)}{\partial w_j^2} = -\sum_{i=1}^{N} x_{ij}^2 \cdot p(x_i; \boldsymbol{w}) \cdot (1 - p(x_i; \boldsymbol{w}))$$

$$\frac{\partial^2 L(w)}{\partial w_j \partial w_l} = -\sum_{i=1}^{N} x_{ij} x_{il} \cdot p(x_i; \boldsymbol{w}) \cdot (1 - p(x_i; \boldsymbol{w}))$$

$$w^{(t+1)} = w^{(t)} - H^{-1}\left(w^{(t)}\right)\nabla f\left(w^{(t)}\right)$$

迭代公式

牛顿法存在的问题

- ✓ 经典牛顿法虽然具有二次收敛性，但是要求初始点需要尽量靠近极小点，否则有可能不收敛。
- ✓ 计算过程中需要计算目标函数的二阶偏导数，难度较大。
- ✓ 目标函数的Hessian矩阵无法保持正定，会导致算法产生的方向不能保证是f在Xk处的下降方向，从而令牛顿法失效；
- ✓ 如果Hessian矩阵奇异，牛顿方向可能根本是不存在的。

补充：正定矩阵的定义、性质、判定方法

定义：对于n阶方阵A，如果对任何非零向量z，都有$z^T Az > 0$，就称A为正定矩阵。

性质：正定矩阵一定是非奇异的、可逆的矩阵。

判定：对称阵A为正定的充分必要条件为A的特征值均大于0。

本章小结

- Logistic回归在本质上是线性回归。
- Logistic回归解决的是因变量的离散型问题，即分类问题（二类问题）。
- Logistic回归适合于当因变量属于二项分布这类问题。
- Logistic回归模型的输出是介于0-1之间的数，是一个概率值，基于sigmoid函数。
- 该模型的求解采用最大似然估计和牛顿法。
- 建模过程不需要对属性变量进行高斯分布的假设，但依然需要条件独立假设。

课外任务：

根据前面的内容，请自行推导Logistic回归建模方法的整个过程。