# 统计学习方法

## 感知机、逻辑回归、朴素贝叶斯

杨浩哲

山软智库讨论班

2019 年 10 月 19 日

# 大纲 I

# 大纲 II

- 朴素贝叶斯的策略和算法

4 补充

# 前情提要

- 统计学习方法的三要素
- 统计学习方法能解决的三类问题
- 感知机、逻辑回归、朴素贝叶斯分别是什么模型

# 提前预复习

- 观察二项分布、泊松分布、正态分布，思考将其抽象为一个函数的可行性。
- 尽可能把所有引入的部份自学完成
- 尽可能把所有其他的部份自学完成

# 引入：平面表示方法

对于二维空间，平面是一条直线：

$$ax + by + c = 0 \tag{1}$$

上式等价于

$$w_1 x_1 + w_2 x_2 + b = 0 \tag{2}$$

此时，$w \in R^2, x \in R^2$。同时，注意到 $w$ 是法向量。这种表示方法可以推广到任意维的欧式空间（$R^n$）。因此，任意维度的空间中，一个平面可以表示为两个向量点积等于 0 的形式。

# 引入：距离和范数 I

**距离** 距离的定义是一个宽泛的概念，只要满足非负、自反、三角不等式就可以称之为距离。

**范数** 范数是一种**强化了的距离概念**，它在定义上比距离多了一条数乘的运算法则。**有时候**为了便于理解，我们可以把范数当作距离来理解。

其中，最常用的是 $L_1$ 范数（一范数）和 $L_2$ 范数（二范数）。

# 引入：距离和范数 II

一范数如下，其表示向量 $x$ 中非零元素的绝对值之和：

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i| \tag{3}$$

二范数如下，表示向量元素的平方和再开平方，实际上就是一般情况下的距离[1]：

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} \tag{4}$$

[1]表述清楚的情况下，下标 2 可以被省略，即默认 $\|\mathbf{x}\| = \|\mathbf{x}\|_2$

# 引入：点到平面的距离

对于二维空间上的任意一个点 $x$，均可表示为 $[x_1, x_2]^T$ 的形式[2]。

首先考虑到高中学到的点 $[x_0, y_0]$ 到线段（即二维中的平面）的距离：

$$\frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}} \tag{5}$$

使用向量表示时，该表示方式变为：

$$\frac{1}{\|w\|} |w \cdot x + b| \tag{6}$$

[2]向量一般均指列向量，因此如果横向表示，一般添加转置符号

# 感知机的模型

感知机的模型由一个平面和一个二值映射函数组成：

$$f(x) = sign(wx + b) \tag{7}$$

其中，

$$sign(x) = \begin{cases} +1, & x \geq 0 \\ -1 & x < 0 \end{cases} \tag{8}$$

# 感知机的策略

感知机学习的目标是求得一个能够将训练集的正例和负例**完全正确分开**的超平面。

感知机的损失函数定义**为误分类点到超平面 $S$ 的总距离。**，其策略为在假设空间中选取令损失函数最小的模型参数。

根据之前引入的距离，假设误分类点集合为 $M^3$，那么损失函数就可以定义为：

$$L(w,b)=\sum_{x_i\in M}|w\cdot x_i+b| \tag{9}$$

由于对于误分类点 $x_i$，可以得到

$$-y_i\left(w\cdot x_i+b\right)>0 \tag{10}$$

所以，可以用上式将损失函数的绝对值去掉，得最终的损失函数形式：

$$L(w,b)=-\sum_{x_i\in M}y_i\left(w\cdot x_i+b\right) \tag{11}$$

# 引入：关于损失函数

损失函数由三要素中的算法来优化，其表明了主观人为认定的模型的**某种缺陷**，因此，其**一般**有以下几个特征[4]：

- 恒大于等于 0，且仅当模型是理想模型不存在缺陷时，损失函数为 0[5]
- 模型的缺陷越多，损失函数值越大。
- 为了方便算法进行优化，损失函数应该尽可能保证连续可导。

[4]注意是一般，一般，一般

[5]当然可以反过来恒小于等于 0，但这没有必要

# 感知机算法的原始形式

对给定的数据集：

$$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\} \tag{12}$$

感知机算法的优化目标的形式化表示：

$$\min_{w,b} L(w, b) = -\sum_{x_i \in M} y_i (w \cdot x_i + b) \tag{13}$$

感知机算法的优化方法：首先，任意选取一个超平面 $w_0, b_0$，然后用随机梯度下降法不断地极小化目标函数13。在极小化的过程中，一次随机选取**一个误分类点使其梯度下降。**损失函数 $L(w, b)$ 的总梯度为：

$$\begin{aligned} \nabla_w L(w, b) &= -\sum_{x_i \in M} y_i x_i \\ \nabla_b L(w, b) &= -\sum_{x_i \in M} y_i \end{aligned} \tag{14}$$

# 引入：梯度 I

**梯度** 是一个向量（矢量），表示某一函数在该点处的方向导数沿着该方向取得最大值，即函数在该点处沿着该方向（此梯度的方向）变化最快，变化率最大。

# 引入：梯度 II

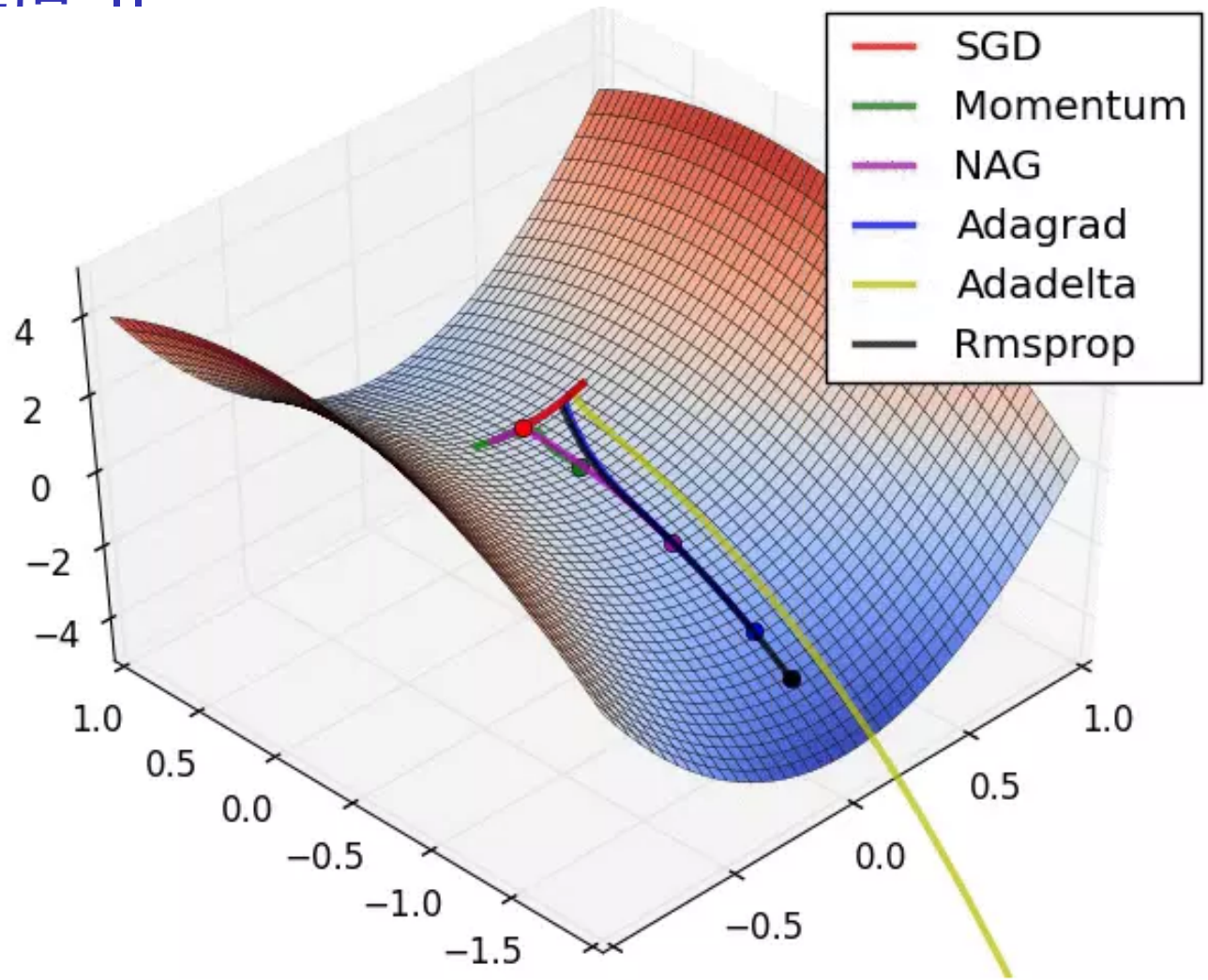


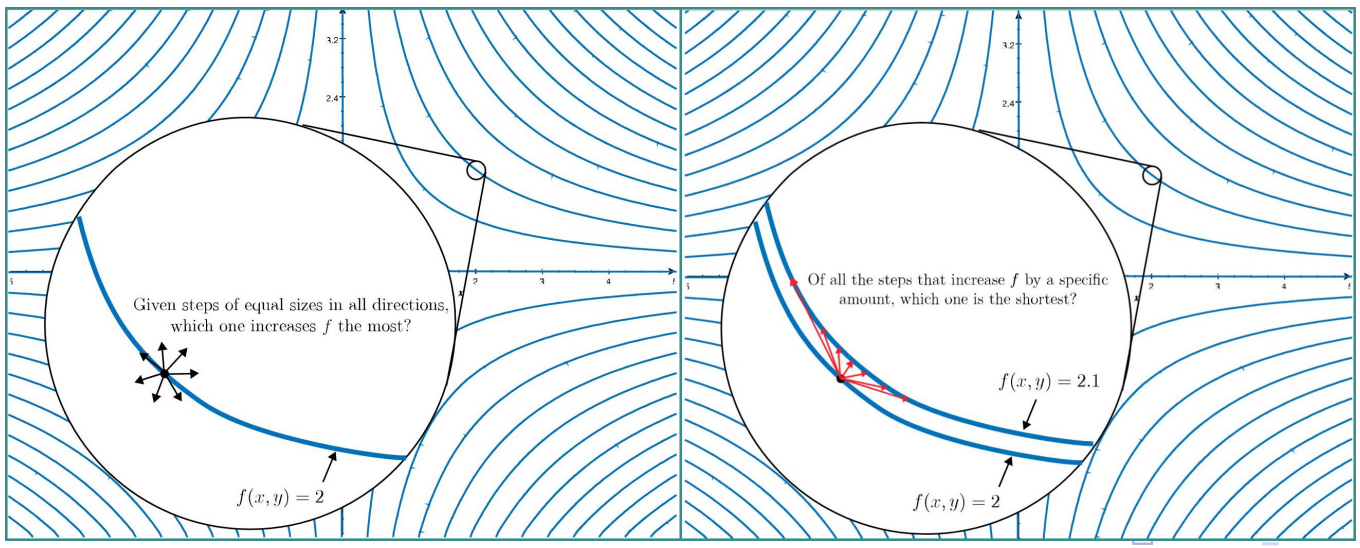

图: 图来自对梯度概念的直观理解

# 引入：梯度下降

**梯度下降** 为了求得一个函数的最大或者最小值（一般是最小值），算法努力的求出在当前值下的梯度，并朝着梯度的方向下降的方法。

为什么要选择朝着梯度下降可以从以下两个角度考虑：

- 在特定函数点，固定每次移动的步长，向那个方向移动函数值增长最快
- 固定需要增加的函数值，向哪个方向需要移动的步长最短？



# 最常见的三种梯度下降法

- 梯度下降法：选择所有的误分类点进行更新
- 随机梯度下降法：随机选择一个误分类点进行更新，这也是感知机算法所采用的方法。
- 批量梯度下降法：随机选择一批误分类点进行更新，是梯度下降法和随机梯度下降法的折中。

实际上，这三者在蒙特卡洛方法的角度下具有一定的等价性。

# 感知机算法的学习策略

**算法 2.1（感知机学习算法的原始形式）**

输入：训练数据集 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$，其中 $x_i\in\mathcal{X}=\mathbf{R}^n$，$y_i\in\mathcal{Y}=\{-1,+1\}$，$i=1,2,\cdots,N$；学习率 $\eta\ (0<\eta\leqslant 1)$；

输出：$w,b$；感知机模型 $f(x)=\mathrm{sign}(w\cdot x+b)$.

（1）选取初值 $w_0,b_0$

（2）在训练集中选取数据 $(x_i,y_i)$

（3）如果 $y_i(w\cdot x_i+b)\leqslant 0$

$$w\leftarrow w+\eta y_i x_i$$

$$b\leftarrow b+\eta y_i$$

（4）转至（2），直至训练集中没有误分类点. ■

# 算法的收敛性

参考《统计学习方法》P31，推公式的过程中注意：

- $y^2 = 1$
- $w_{opt}$ 已经被归一化，即 $\|\hat{w}_{opt}\| = 1$
- 需要用到施瓦茨不等式

由于收敛性的证明实际上不会对我们理解模型或者对后续的学习有帮助，所以不再做详细推导。

# 感知机学习算法的对偶形式 I

很多时候，模型的学习都存在一个对偶形式，对偶的存在能够减少模型学习的计算量。

根据之前的更新策略

$$\begin{aligned} w &\leftarrow w + \eta y_i x_i \\ b &\leftarrow b + \eta y_i \end{aligned} \tag{15}$$

假设该更新一共更新了 $n$ 次，那么 $w, b$ 关于 $(x_i, y_i)$ 的增量分别是 $\alpha_i y_i x_i$ 和 $\alpha_i y_i$，这里 $\alpha_i = n_i \eta$，表示一组 $(x_i, y_i)$ 可以在这 $n$ 次中贡献了 $n_i$ 次。

$$\begin{aligned} w &= \sum_{i=1}^{N} \alpha_i y_i x_i \\ b &= \sum_{i=1}^{N} \alpha_i y_i \end{aligned} \tag{16}$$

这里，实例点更新次数越多（$n_i$ 越大），意味着它距离分离超平面越近，也就越难正确分类。

# 感知机学习算法的对偶形式 II

由于感知机的模型是由误分类点的更新驱动的，所以根据上述公式，感知机的形式也可以表示为：

$$f(x) = \text{sign}\left(\sum_{j=1}^{N} \alpha_j y_j x_j \cdot x + b\right) \tag{17}$$

# 感知机对偶形式的学习算法 I

**算法 2.2（感知机学习算法的对偶形式）**

输入：线性可分的数据集 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$，其中 $x_i\in\mathbf{R}^n$，$y_i\in\{-1,+1\}$，$i=1,2,\cdots,N$；学习率 $\eta$（$0<\eta\leqslant 1$）；

输出：$\alpha,b$；感知机模型 $f(x)=\operatorname{sign}\left(\sum_{j=1}^{N}\alpha_j y_j x_j\cdot x+b\right)$.

其中 $\alpha=(\alpha_1,\alpha_2,\cdots,\alpha_N)^{\mathrm{T}}$.

（1）$\alpha\leftarrow 0$，$b\leftarrow 0$

（2）在训练集中选取数据 $(x_i,y_i)$

（3）如果 $y_i\left(\sum_{j=1}^{N}\alpha_j y_j x_j\cdot x_i+b\right)\leqslant 0$

$$\alpha_i\leftarrow\alpha_i+\eta$$
$$b\leftarrow b+\eta y_i$$

（4）转至（2）直到没有误分类数据. ■

# 感知机对偶形式的学习算法 II

可以看到训练中的训练示例仅以内积的形式出现：

$$y_i \left( \sum_{j=1}^{N} \alpha_j y_j x_j \cdot x_i + b \right) \leqslant 0 \tag{18}$$

因此，可以将内积提前计算，用于减少实际训练时候的计算开销，该提前计算的内积就构成了一个 $N \times N$ 的矩阵（$N$ 是数据集样本个数），称为 Gram 矩阵。

$$G = [x_i \cdot x_j]_{N \times N} \tag{19}$$

# 逻辑回归回顾

- 逻辑回归模型是感知机与概率模型的融合
- 感知机是通过 $sign(\cdot)$ 函数将 $x$ 映射到二值空间
- 逻辑回归是通过 logistic 函数将 $x$ 映射到概率空间

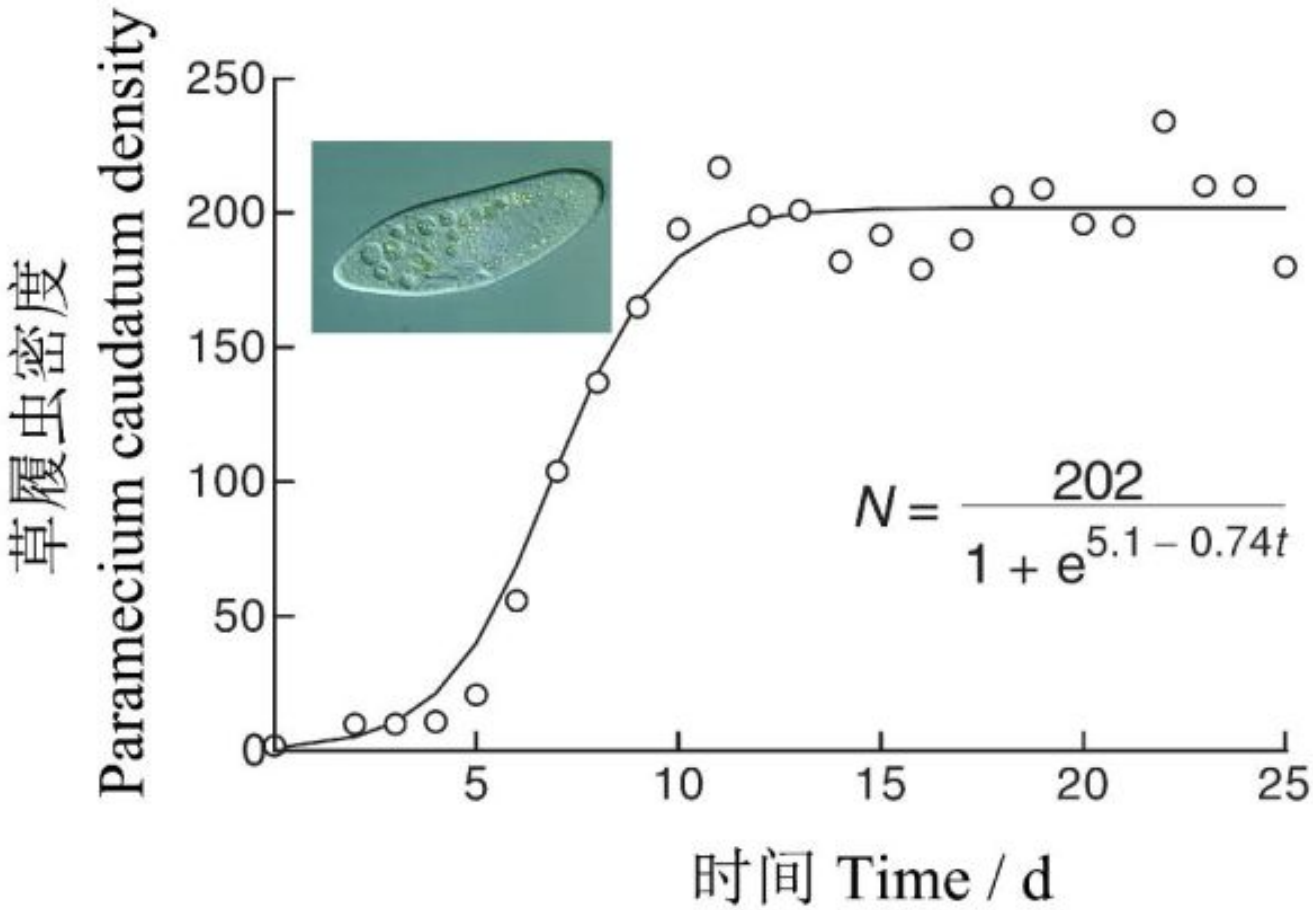
逻辑回归的模型表示：

$$P(Y=1|x) = \frac{exp(wx+b)}{1+exp(wx+b)} \tag{20}$$

$$P(Y=0|x) = \frac{1}{1+exp(wx+b)} \tag{21}$$

# 生长曲线

种群生长曲线能够很好的描述一个种群随着时间的数量变化，如草履虫的生长曲线为：



# 寻找曲线

我们需要寻找一个性质良好的满足将映射关系（从线性空间映射到概率空间）的函数，其定义域是 $(-\infty, +\infty)$，值域是 $[0, +1]$ 的函数，为了寻找这样的函数，我们可以先寻找该函数的逆函数（定义域 $[0, +1]$，值域 $(-\infty, +\infty)$）

- $log(P)$ 函数在 $[0, 1]$ 上的值域为 $(-\infty, 0]$，此时直接逼近了一半。而 $log(P)$ 只有在 $(0, +\infty)$ 的时候才符合其值域。
- 考虑 $log(g(P))$，$g(P)$ 是一个将 $[0, 1]$ 映射到 $(0, +\infty)$ 的函数。
- 考虑倒数，相反数... 最终 $g(x)$ 被确定为 $\frac{P}{1-P}$
- 最终，我们找到了一个这样的变换，被称为 logistic 变换：$log(\frac{P}{1-P}) = w \cdot x$

求解 $P$ 可得：

$$P(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{1 + e^{-(\boldsymbol{w} \cdot \boldsymbol{x})}} \tag{22}$$

逻辑回归的模型表示：

$$P(Y=1|x)=\frac{exp(wx+b)}{1+exp(wx+b)} \tag{23}$$

$$P(Y=0|x)=\frac{1}{1+exp(wx+b)} \tag{24}$$

注意两个公式实际上是等价的

# 引入：极大似然估计 I

对于我们获得的一组样本，既然样本对应的观测值已经出现，那么实验的条件（或者说原概率分布）应该有利于对应样本的发生。

极大似然估计就是**利用已知的样本结果**，反推**最有可能（最大概率）导致这样结果的参数值。**

**当确认了样本的分布时**[6]，一个确定的参数就会确定一个样本发生的概率。此时，参数的所有可能取值就构成了一个参数空间，极大似然估计就是需要在这个参数空间中选择一个参数，让样本整体发生的概率最大。

# 引入：极大似然估计 II

极大似然估计的求解步骤为：

1. 列出样本整体发生的概率（即所谓的似然函数），这个概率是一个与参数 $\theta$ 有关的函数：

$$L(\theta) = P(X_1 = x_1; X_2 = x_2; ...) = \prod P(X_i = x_i) \tag{25}$$

2. 最大化该函数值，因为求其最大值需要求导，为了求导方便，一般通过取对数（注意对数的单调性）将乘积化和，随后求导求极值，得到相应的解。

[6]注意一定要确认了分布才能使用极大似然估计

# 逻辑回归的策略 I

逻辑回归可以看做是概率模型，从这个角度看，求其模型的解就需要使用极大似然估计法来求解。

其形式为：

$$L(\theta) = \prod_{i=1}^{n} P(\boldsymbol{x}_i; \boldsymbol{\theta})^{y_i} \left(1 - P(\boldsymbol{x}_i; \boldsymbol{\theta})\right)^{1-y_i} \tag{26}$$

# 逻辑回归的策略 II

求其对数似然函数，化简过程为：

$$
\begin{aligned}
\ell(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \left(y_i \log P(\boldsymbol{x}_i;\boldsymbol{\theta}) + (1-y_i)\log\left(1-P(\boldsymbol{x}_i;\boldsymbol{\theta})\right)\right) \\
&= \sum_{i=1}^{n} \log\left(1-P(\boldsymbol{x}_i;\boldsymbol{\theta})\right) + \sum_{i=1}^{n} y_i \log \frac{P(\boldsymbol{x}_i;\boldsymbol{\theta})}{1-P(\boldsymbol{x}_i;\boldsymbol{\theta})} \\
&= \sum_{i=1}^{n} \log\left(1-P(\boldsymbol{x}_i;\boldsymbol{\theta})\right) + \sum_{i=1}^{n} y_i \boldsymbol{\theta}\boldsymbol{x}_i \\
&= \sum_{i=1}^{n} -\log\left(1+e^{\theta x_i}\right) + \sum_{i=1}^{n} y_i \boldsymbol{\theta}\boldsymbol{x}_i
\end{aligned}
\tag{27}
$$

# 逻辑回归的策略 III

由于 $x$ 是一个 $n$ 维向量，因此求导需要对其每一个维度进行求导，令其中 $x_{ij}$ 表示样本 $x_i$ 的第 $j$ 个分量，可得：

$$\begin{aligned}\frac{\partial \ell}{\partial \theta_i} &= -\sum_{i=1}^{n} \frac{1}{1+e^{\theta x_i}} e^{\theta x_i} \boldsymbol{x}_{ij} + \sum_{i=1}^{n} y_i \boldsymbol{x}_{ij} \\ &= \sum_{i=1}^{n} \left(y_i - P\left(\boldsymbol{x}_i; \boldsymbol{\theta}\right)\right) \boldsymbol{x}_{ij}\end{aligned} \tag{28}$$

该求导没有解析解，只存在数值解，可以可以使用之前提到的梯度下降法进行求解。

# 引入：一大堆概率定律

这里自己学习，上一次讨论班讲过了

- 单变量下的贝叶斯公式表达式?
- 多维变量下，具有相关性的贝叶斯公式的展开式?
- 多维变量下，各维之间不存在相关性的贝叶斯公式的展开式?

# 朴素贝叶斯回顾

- 贝叶斯概率的“朴素”形式

朴素贝叶斯的模型表示：

$$P(Y = C_k|X = x) = \frac{P(X = x, Y = C_k)}{P(X = x)} \tag{29}$$

当输入空间 $X$ 为一维空间（及 $x$ 为一维数据）时，该函数和贝叶斯概率求解的方法相同。当输入空间高于一维时，不同维度之间的相关性极大的提高了该公式求解的复杂度，因此，假设各变量独立，由此得到的计算公式，即为**朴素**贝叶斯。

朴素贝叶斯的模型表示：

$$P(Y = C_k|X = x) = \frac{P(X = x, Y = C_k)}{P(X = x)} \tag{30}$$

其中，$P(X = x)$ 由已得到的数据集求得，在**使用**时是常数。而

$$\begin{aligned} P(X = x, Y = C_k) &= P(Y = C_k)P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n|Y = C_k) \\ &= P(Y = C_k)\prod_j P(X_j = x_j|Y = C_k) \end{aligned} \tag{31}$$

求解时，对得到的 $\hat{x}$，对每一个 $Y \in C_K$，求解对应类别的条件概率 $P(Y = C_k|X = x)$，选择值最大的类作为最后的分类结果，其形式化表示为：

$$\underset{C_k}{argmax} \quad P(Y = C_k|X = x) \tag{32}$$

朴素贝叶斯中策略和算法区分的并不明显，因为其求解过程是通过数据集一步到位的。

在求解朴素贝叶斯模型的过程中，我们要求得的参数有两类，分别是 $P(X)$ 和 $P(X=x|Y=C_k)$

$$P(X)=\sum_k\left(P(Y=C_k)\prod_i P(X_i=x_i|Y=C_k)\right) \tag{33}$$

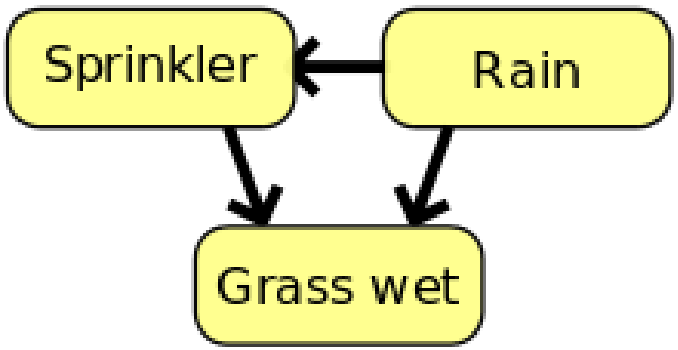
$$P(X=x|Y=C_k)=\prod_j P(X_j=x_j|Y=C_k) \tag{34}$$

对于相应的概率值，最简单的方法是通过统计求得。但如果我们清楚了数据集的分布（如正态分布），那么也可以通过已有数据集来估计分布的参数，得到相应的值[7]。

[7]使用正态分布的朴素贝叶斯可以参考wiki-朴素贝叶斯，一般教程中较为少见

# 附：朴素贝叶斯模型的后续

- 当数据较少的时候，可能存在由于数据缺失导致的条件概率值为零的情况（但实际中不会为零），这个时候就需要用到一些方法来避免这种零值，称为**贝叶斯估计**。
- 如果不假设条件独立性，而是认为条件之间存在概率依存关系，那么模型就变成了**贝叶斯网络**，也称为**信念网络**，这是另外的一门学科了。
- 朴素贝叶斯和贝叶斯网络都遵循贝叶斯定理，其背后的理论叫做**贝叶斯决策理论**。



# 附：适用于分类模型的一个数据集示例

数据来源于wiki-朴素贝叶斯

| 性别 | 身高(英尺) | 体重(磅) | 脚的尺寸(英寸) |
|---|---|---|---|
| 男 | 6 | 180 | 12 |
| 男 | 5.92 (5'11") | 190 | 11 |
| 男 | 5.58 (5'7") | 170 | 12 |
| 男 | 5.92 (5'11") | 165 | 10 |
| 女 | 5 | 100 | 6 |
| 女 | 5.5 (5'6") | 150 | 8 |
| 女 | 5.42 (5'5") | 130 | 7 |
| 女 | 5.75 (5'9") | 150 | 9 |