统计学习方法习题解答

X

2017 年 8 月 16 日

前言

本文是作者对李航*统计学习方法*中的每章习题所做的解答合集。本文的章节顺序，数学符号等都尽量与该书保持一致，解答中也参考了网络资源或者其他书籍，均在对应习题序号下列出。由于水平所限，文中谬误在所难免，欢迎指正。

目录

1 统计学习方法概论

用$\mathcal{X},\mathcal{Y}$表示输入空间和输出空间，X和Y分别是空间$\mathcal{X}$和$\mathcal{Y}$上的随机变量。假设训练集有N个样本$T=\{(x_1,y_1)...(x_N,y_N)\}$ 由联合概率分布$P(X,Y)$独立同分布产生。

1.1 伯努利模型的统计学习方法三要素

Exercise 1.1. 伯努利模型的极大似然估计和贝叶斯估计统计学习方法三要素

Solution 1.1.

伯努利模型的极大似然估计统计学习方法三要素

1. 模型：条件概率$\mathbb{P}(Y|X)$服从参数为θ的伯努利分布

$$\mathbb{P}(Y=1|X)=\theta,\theta\in[0,1]$$

2. 策略：最大化似然函数

$$\max L(\theta)=\max\prod_{i=1}^{N}\theta^{y_i}\,(1-\theta)^{1-y_i}$$

3. 算法：对似然函数取对数求导得到驻点即可得到θ最优解

令$G(\theta)=\log L(\theta)$, 则有

$$\begin{aligned}G^{'}(\theta) &= \sum_{i=1}^{N}\left(\frac{y_i}{\theta}-\frac{1-y_i}{1-\theta}\right)\\ &= \sum_{i=1}^{N}\frac{y_i-\theta}{\theta(1-\theta)}\end{aligned}$$

令$G^{'}(\theta)=0$ 得到

$$\sum_{i=1}^{N}(y_i-\theta)=0\Rightarrow\theta=\frac{\sum_{i=1}^{N}y_i}{N}$$

伯努利模型的贝叶斯估计统计学习方法三要素

1. 模型：条件概率$\mathbb{P}(Y|X)$服从参数为θ的伯努利分布

$$\mathbb{P}(Y=1|X)=\theta,\theta\in[0,1]$$

其中θ也是一个变量，假设θ服从Beta分布$\theta\sim\mathcal{B}(\alpha,\beta)$，那么其概率密度函数为

$$p(\theta)=\frac{1}{B(\alpha,\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$$

其中$B(\alpha,\beta)=\int_0^1\theta^{\alpha-1}(1-\theta)^{\beta-1}d\theta=\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$，

2. 策略：求θ的后验概率分布的期望$\mathbb{E}(\theta|y_1, y_2, ...y_N)$

3. 算法：用p表示概率密度函数，根据贝叶斯公式

$$\begin{aligned}
p(\theta|y_1, y_2, ...y_N) &= \frac{p(\theta, y_1, y_2, ...y_N)p(\theta)}{p(y_1, y_2, ...y_N)} \\
&= \frac{p(\theta, y_1, y_2, ...y_N)p(\theta)}{\int_0^1 p(\theta, y_1, y_2, ...y_N)d\theta} \\
&= \frac{\prod_{i=1}^N \theta^{y_i}(1-\theta)^{1-y_i} \frac{1}{B(\alpha,\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}}{\int_0^1 \prod_{i=1}^N \theta^{y_i}(1-\theta)^{1-y_i} \frac{1}{B(\alpha,\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}d\theta} \\
&= \frac{\theta^{\alpha+k-1}(1-\theta)^{\beta+N-k-1}}{\int_0^1 \theta^{\alpha+k-1}(1-\theta)^{\beta+N-k-1}d\theta} \quad \left(\text{令}\sum_{i=1}^N y_i = k\text{，则有}\sum_{i=1}^N 1-y_i = N-k\right) \\
&= \frac{\theta^{\alpha+k-1}(1-\theta)^{\beta+N-k-1}}{B(\alpha+k, \beta+N-k)} \quad \left(\text{根据定义}B(\alpha,\beta) = \int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1}d\theta\right)
\end{aligned}$$

故$\theta|y_1, y_2, ...y_N \sim \mathcal{B}(\alpha+k, \beta+N-k)$

根据Beta分布的性质可得$\mathbb{E}(\theta|y_1, y_2, ...y_N) = \frac{\alpha+k}{N+\alpha+\beta}$

1.2 经验风险最小化与极大似然估计

Exercise 1.2. 当模型是条件概率分布，损失函数是对数损失时，经验风险最小化等价于极大似然估计

Proof.

$$\begin{aligned}
\min R_{\text{emp}} &= \min_f \frac{1}{N}\sum_{i=1}^N L(y_i, f(x_i)) \\
&= \min_f \sum_{i=1}^N L(y_i, f(x_i)) \quad \text{(去掉常数乘积项)} \\
&= -\min_\theta \sum_{i=1}^N \log \mathbb{P}_\theta(y_i|x_i) \quad \text{(条件概率模型，对数损失)} \\
&= \max_\theta \sum_{i=1}^N \log \mathbb{P}_\theta(y_i|x_i) \\
&= \max_\theta \log \prod_{i=1}^N \mathbb{P}_\theta(y_i|x_i) \\
&= \max_\theta \prod_{i=1}^N \mathbb{P}_\theta(y_i|x_i)
\end{aligned}$$

□

2 感知机

假设模型的输入空间$\mathcal{X} = R^n$，输出空间$\mathcal{Y} = \{-1, 1\}$，训练集有N个样本$T = \{(x_1, y_1)...(x_N, y_N)\}$由联合概率分布$P(X, Y)$独立同分布产生。

2.1 感知机不能表示异或

Exercise 2.1. 验证感知机为什么不能表示异或

Proof. 设数据集$T = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$

根据异或的真值表，假设$(x_1, y_1) = ((0, 0), -1), (x_2, y_2) = ((0, 1), 1), (x_3, y_3) = ((1, 0), 1), (x_4, y_4) = ((1, 1), -1)$

假设存在超平面$\omega^T x + b = 0$将T正确分类，其中$\omega = (\omega_1, \omega_2)^T$，则有

$$y_i \left(\omega^T x + b\right) > 0, i = 1, 2, 3, 4$$

$$\Rightarrow \begin{cases} b < 0 \\ w_2 + b > 0 \\ w_1 + b > 0 \\ w_1 + w_2 + b < 0 \end{cases} \tag{2.1}$$

公式(2.1)中前三个不等式约束与第四个明显矛盾，故不存在分割超平面，从而感知机不能表示异或

□

2.2 求解感知机

Exercise 2.2. 构建从训练数据集求解感知机模型的例子

Solution 2.1.

自制一个非常简单的例子。

假设正实例点$x_1 = (6, 2)^T, x_2 = (5, 1)^T$，负实例点$x_3 = (2, 5)^T$

1. 初值$w_0 = 0, b_0 = 0$
2. x_1未被正确分类，计算$w_1 = w_0 + y_1 x_1 = (6, 2)^T$，$b_1 = b_0 + y_1 = 2$，得到新的超平面为$w_1 x + b_1 = (6, 2)^T x + 2$
3. 在新的超平面下，x_3未被正确分类，计算$w_2 = w_1 + y_3 x_3 = (4, -3)^T$，$b_2 = b_1 + y_1 = 1$，得到新的超平面为$w_2 x + b_2 = (4, -3)^T x + 1$
4. x_1, x_2, x_3都被正确分类，故感知机模型为$f(x) = \text{sign}(4x^1 - 3x^2 + 1)$

2.3 样本线性可分的充要条件

[凸集分离定理，https://wenku.baidu.com/view/23497261a45177232f60a26f.html]

Exercise 2.3. 样本线性可分的充要条件是正实例点构成的凸壳与负实例点构成的凸壳互不相交

Proof. 充分性：凸壳不相交⇒ 线性可分

1. 证明凸壳conv(S)是凸集。

   假设任意两点$x, y \in \text{conv}(S)$，根据凸壳conv(S)定义可知

$$\begin{aligned} x &= \sum_{i=1}^{N} \alpha_i x_i \\ y &= \sum_{i=1}^{N} \beta_i x_i \\ \sum_{i=1}^{N} \alpha_i &= \sum_{i=1}^{N} \beta_i = 1, \alpha_i \geq 0, \beta_i \geq 0 \end{aligned}$$

   对于$\forall \lambda \in [0,1]$，都有

$$\begin{aligned} \lambda x + (1-\lambda) y &= \lambda \sum_{i=1}^{N} \alpha_i x_i + (1-\lambda) \sum_{i=1}^{N} \beta_i x_i \\ &= \sum_{i=1}^{N} \left(\lambda \alpha_i + (1-\lambda)\beta_i\right) x_i \end{aligned}$$

   其中

$$\sum_{i=1}^{N} \left(\lambda \alpha_i + (1-\lambda)\beta_i\right) = \lambda \sum_{i=1}^{N} \alpha_i + (1-\lambda) \sum_{i=1}^{N} \beta_i = \lambda + (1-\lambda) = 1$$

   $\lambda x + (1-\lambda) y$满足conv($S$)的定义，故还是属于此凸壳中，故凸壳是凸集

2. 证明凸壳conv(S)是闭集。

   假设凸壳中的某数列$\{z_n \in \text{conv}(S)\}$ 收敛于z。

   根据凸壳的定义，$z_n = \sum_{i=1}^{N} \lambda_i^n x_i, \lambda_i^n \in [0,1]$。对于给定的分量$i$，$\{\lambda_i^n\}, n = 1, 2, ...$ 构成了一个有界序列。

   对于第一个分量$\{\lambda_1^n\}$，其必然存在一个收敛子序列$\{\lambda_1^{a_1^n}\}$，假设其收敛于λ_1。由上标序列$\{a_1^n\}$可以构建一个第二个分量的子序列$\{\lambda_2^{a_1^n}\}$，其依然是有界序列，则必然存在一个收敛子序列，不妨假设为$\{\lambda_2^{a_2^n}\}$，收敛于λ_2。以此类推，找到最后一个分量的收敛子序列$\{\lambda_N^{a_N^n}\}$，收敛于λ_N。

   这样$\{a_N^n\}$就是所有收敛子序列下标的交集，为简化表示下文用$\{a_n\}$代替。因为收敛子序列的子序列依然是收敛的，且收敛于相同极限，故以相同下标$\{a_n\}$构成的子序列集$\{\lambda_i^{a_n}\}, n = 1, 2, ...$均为收敛序列，且

$$\{\lambda_i^{a_n}\} \to \lambda_i, \forall i \tag{2.2}$$

另外由于$\sum_{i=1}^{N}\lambda_i^n=1$，根据极限可加性可到

$$\sum_{i=1}^{N}\lambda_i=1 \tag{2.3}$$

根据关系式(2.2)可以得到，$z_{a_n}=\sum_{i=1}^{N}\lambda_i^{a_n}x_i\to z^*=\sum_{i=1}^{N}\lambda_i x_i$。根据公式(2.3)可知$z^*\in \mathrm{conv}(S)$。

根据假设$\{z_n\}$收敛于z，那么其任意收敛子序列必同样收敛于z。于是可以得到$z=z^*\in\mathrm{conv}(S)$。故凸壳是闭集。

3. 证明凸集分离定理的一个引申结论

Theorem 2.1. 假设S_1和S_2是$\mathbb{R}^n$的两个非空闭凸集，S_1有界，且$S_1\cap S_2=\emptyset$，则存在非零向量ω和$\xi>0$，使得

$$\inf\{\omega^T x|x\in S_1\}\geq\xi+\sup\{\omega^T x|x\in S_2\} \tag{2.4}$$

Proof. 设$S=S_1-S_2=\{y-x|x\in S_1,y\in S_2\}$

(a) 因为$S_1\cap S_2=\emptyset$，所以$0\notin S$。

(b) S_1,S_2为凸集，所以S为凸集且$S\neq\emptyset$

(c) 假设存在序列$\{z_n\in S\}$收敛于z。根据z_n的定义存在$\{x_n\in S_1\},\{y_n\in S_2\}$，使得$z_n=y_n-x_n,n=1,2...$。

因为S_1为有界闭集，则$\{x_n\}$存在收敛子序列$\{x_{a_n}\}\to x\in S_1$，另外收敛序列的任何子序列收敛于同一个极限，$\{z_{a_n}\}\to z$，所以可知$\{y_{a_n}\}\to x+z$。因为S_2为闭集，故$x+z\in S_2$。由此根据S的定义可知$z=(x+z)-x\in S$，故所以S 为闭集。

综上所述，S为非空闭凸集，$0\notin S$，根据凸集分离定理，存在非零向量ω和$\xi>0$，使得$\forall z\in S$，有

$$\begin{aligned}
&\omega^T*0\geq\xi+\omega^T z\\
\Rightarrow\quad &0\geq\xi+\omega^T(y-x),y\in S_2,x\in S_1\\
\Rightarrow\quad &\omega^T x\geq\xi+\omega^T y\\
\Rightarrow\quad &\inf\{\omega^T x|x\in S_1\}\geq\xi+\sup\{\omega^T x|x\in S_2\}
\end{aligned}$$

□

4. 用$\mathrm{conv}(S_+),\mathrm{conv}(S_-)$分别表示由正负实例点构成的凸壳。由上可知$\mathrm{conv}(S_+),\mathrm{conv}(S_-)$为非空闭凸集，其有有限个样本点构成，均为有界集。如果二者不相交，那么根据定理(2.1)，存在非零向量$\omega\in\mathbb{R}^N$及$\xi>0$，使得$\alpha\geq\xi+\beta$，其中$\alpha=\inf\{\omega^T x|\forall x\in\mathrm{conv}(S_+)\},\beta=\sup\{\omega^T y|\forall y\in\mathrm{conv}(S_-)\}$。

定义超平面$\omega^T x-\frac{\xi}{2}-\beta=0$，那么对于$\forall x\in\mathrm{conv}(S_+)$，都有

$$\omega^T x-\frac{\xi}{2}-\beta\geq\inf\{\omega^T x\}-\frac{\xi}{2}-\beta\geq\xi+\beta-\frac{\xi}{2}-\beta\geq\frac{\xi}{2}>0$$

对于$\forall y \in \text{conv}(S_-)$，都有

$$\omega^T y - \frac{\xi}{2} - \beta \leq \sup\{\omega^T y\} - \frac{\xi}{2} - \beta \leq -\frac{\xi}{2} < 0$$

所以超平面$\omega^T x - \frac{\xi}{2} - \beta = 0$将正负实例点构成的凸壳线性分割。

必要性：线性可分⇒ 凸壳不相交

假设存在分割超平面$\omega^T x + b = 0$，对于正实例集$\{x_i\}$，$\omega^T x_i + b > 0$，对于负实例集$\{y_i\}$，$\omega^T y_i + b < 0$。用反证法证明正负实例集构成的凸壳$\text{conv}(S_+), \text{conv}(S_-)$没有交集。

假设$\text{conv}(S_+), \text{conv}(S_-)$有交集，用$z$表示其中一点。

因为$z \in \text{conv}(S_+)$，那么

$$\omega^T z + b = \omega^T \left(\sum_{i=1}^{N} \lambda_i x_i \right) + b = \sum_{i=1}^{N} \lambda_i \left(\omega^T x_i + b \right) > 0$$

另外$z \in \text{conv}(S_-)$，同理可得

$$\omega^T z + b = \omega^T \left(\sum_{i=1}^{N} \lambda_i y_i \right) + b = \sum_{i=1}^{N} \lambda_i \left(\omega^T y_i + b \right) < 0$$

$\omega^T z + b > 0, \omega^T z + b < 0$不可能同时成立，那么便不可能存在同属于$\text{conv}(S_+), \text{conv}(S_-)$的点，故凸壳$\text{conv}(S_+), \text{conv}(S_-)$没有交集。

□

3 k近邻法

3.1 k值与模型复杂度以及预测准确率的关系

3.2 求最近邻点

Exercise 3.1. *利用例题3.2构造的kd树求点$x = (3, 4.5)$的最近邻点*

1. 包含x 的叶节点为$(4, 7)$，以$(4, 7)$为当前最近点。用O表示以x为圆心，半径$d = \sqrt{(4-3)^2 + (7-4.5)^2}$的圆
2. $(4, 7)$的父节点为$(5, 4)$，其另一子节点$(2, 3)$对应的区域与O_1相交，且$(2, 3)$到x的距离更近，则以$(2, 3)$为当前最近点。用O_2 表示以x为圆心，半径$d = \sqrt{(2-3)^2 + (3-4.5)^2}$的圆
3. 继续返回上一父节点$(7, 2)$, 其另一子节点$(9, 6)$与O_2没有交集，且$(7, 2)$是根节点，故停止搜索，$(2, 3)$为x的最近邻点

3.3 k近邻的算法

Exercise 3.2. *输出为x的k近邻的算法*

输入:已构造的kd树，目标点x
输出:x的k近邻点
初始化:长度为k的空优先队列Q，Q中元素q的优先级定义为$d = \|q - x\|_2$

1. 在kd树中找出包含目标点x的叶节点：从根节点出发，递归地向下访问kd树。若x当前维的坐标小于切分点的坐标，则移动到左节点，否则移动到右节点，直至到叶节点为止。
2. 将此叶节点加入优先队列Q
3. 递归的向上回退
   (a) 如果Q中元素未满k个，则进入父节点的另一个子节点搜索，将其区域内的节点按照与x的距离顺序(按照从近到远)加入Q，如果还未满，则回退到父节点的父节点继续搜索，直至Q元素满k个
   (b) 如果Q中元素已满，则以队列顶端的元素的优先级数值为半径，以x为圆心，画圆，用O表示。如果父节点的另一子节点的区域与O 相交，那么搜索该区域，检查是否存在到x更近的点，如果存在，则将这些点进行加入Q(从Q中原来优先级最高的开始替换)。递归的进行搜索；如果不相交，则向上回退
4. 当回退到根节点时，搜索结束，最后的Q中保存的是x的k近邻

4 朴素贝叶斯法

[极大似然估计法推出朴素贝叶斯法中的先验概率公式，https://www.zhihu.com/question/33959624]

假设输入空间$\mathcal{X}$为n维向量的集合，输出空间为分类标记集合$\mathcal{Y} = \{c_1, c_2, ...c_k\}$。$X$和$Y$分别是空间$\mathcal{X}$和$\mathcal{Y}$上的随机变量。训练集有$N$个样本$T = \{(x_1, y_1)...(x_N, y_N)\}$ 由联合概率分布$P(X, Y)$独立同分布产生。

朴素贝叶斯是通过训练集学习联合概率分布$P(X, Y)$: 具体来说是学习先验概率分布$P(Y = c_k), k = 1...K$，以及条件概率分布$P(X = x|Y = c_k)$, 其中x为n维向量，进而学习到联合概率分布$P(X, Y)$。

4.1 参数的极大似然估计

Exercise 4.1. *先验概率$P(Y = c_k)$的极大似然估计是*

$$P(Y = c_k) = \frac{\sum_{i=1}^{N} \mathbb{1}_{y_i = c_k}}{N}, k = 1...K \tag{4.1}$$

Proof. 假设$P(Y = c_k) = \theta_k, k = 1...K$，那么$Y$的概率密度函数可以表示成$\mathbb{P}(Y) = \prod_{k=1}^{K} \theta_k^{\mathbb{1}_{Y=c_k}}$。

进一步可得似然函数

$$L(\theta_k; y_1...y_N) = \prod_{i=1}^{N} P(y_i) = \prod_{k=1}^{K} \theta_k^{N_k}$$

其中$N_k = \sum_{i=1}^{N} \mathbb{1}_{y_i = c_k}$

对数似然函数为

$$\log L(\theta_k; y_1...y_N) = \sum_{k=1}^{K} N_k \log \theta_k$$

存在一个约束条件

$$\sum_{k=1}^{K} \theta_k = 1$$

利用拉格朗日乘子法可构造函数

$$G(\theta_k, \lambda) = \sum_{k=1}^{K} N_k \log \theta_k + \lambda \left(\sum_{k=1}^{K} \theta_k - 1 \right)$$

对G求偏导，令其为零，可得

$$\begin{aligned} \frac{\partial G}{\partial \lambda} &= \sum_{k=1}^{K} \theta_k - 1 = 0 \\ \frac{\partial G}{\partial \theta_k} &= \frac{N_k}{\theta_k} + \lambda = 0 \end{aligned} \tag{4.2}$$

解方程组(4.2)可得

$$\begin{aligned} & \frac{\theta_1}{N_1} = .. = \frac{\theta_K}{N_K} = -\frac{1}{\lambda} \\ \Rightarrow \quad & \theta_1 + ... + \theta_K = -\frac{1}{\lambda}(N_1 + ...N_k) = -\frac{N}{\lambda} = 1 \\ \Rightarrow \quad & \lambda = -N \\ \Rightarrow \quad & \theta_k = \frac{N_k}{N} \end{aligned}$$

□

Exercise 4.2. 假设第j个特征x^j可能取值的集合为$\{a_{j1}..a_{jS_j}\}$，则条件概率$P(X^j = a_{jl}|Y = c_k)$的极大似然估计是

$$P(X^j = a_{jl}|Y = c_k) = \frac{\sum\limits_{i=1}^{N} \mathbb{1}_{x_i^j = a_{jl}, y_i = c_k}}{\sum\limits_{i=1}^{N} \mathbb{1}_{y_i = c_k}}$$

Proof. 假设$\mathbb{P}(X^j = a_{jl}|Y = c_k) = \mu_{lk}, l = 1, 2..S_j, k = 1, 2..K$，那么条件概率的概率密度函数可以表示成

$$\mathbb{P}(X^j|Y) = \prod_{l=1}^{s_j} \prod_{k=1}^{K} \mu_{lk}^{\mathbb{1}_{\{X^j = a_{jl}, Y = c_k\}}}$$

对于给定的j, 观测数据的似然函数为

$$\begin{aligned} & L(\mu_{lk}; (x_1^j, y_1), (x_2^j, y_2), ..., (x_N^j, y_N)) \\ = & \prod_{i=1}^{N} \mathbb{P}(x_i^j, y_i) \\ = & \prod_{i=1}^{N} \mathbb{P}(x_i^j | y_i) \mathbb{P}(y_i) \\ = & \prod_{i=1}^{N} \prod_{l=1}^{s_j} \prod_{k=1}^{K} (\mu_{lk}\theta_k)^{\mathbb{1}_{\{X^j = a_{jl}, Y = c_k\}}} \\ = & \prod_{l=1}^{s_j} \prod_{k=1}^{K} (\mu_{lk}\theta_k)^{N_{lk}} \end{aligned}$$

其中

$$N_{lk} = \sum_{i=1}^{N} \mathbb{1}_{\{x_i^j = a_{jl}, y_i = c_k\}}$$

对数似然函数

$$\log L = \sum_{l=1}^{s_j} \sum_{k=1}^{K} N_{lk} \left(\log \mu_{lk} + \log \theta_k \right)$$

同时存在K个关于μ_{lk}的约束条件

$$\sum_{l=1}^{s_j} \mu_{lk} = 1, k = 1, 2...K$$

以及1个关于θ_k的约束条件

$$\sum_{k=1}^{K} \theta_k = 1$$

利用拉格朗日乘子法可构造函数

$$\begin{aligned} G(\mu_{jk}, \lambda_1, ...\lambda_k) = & \sum_{l=1}^{s_j} \sum_{k=1}^{K} N_{lk} \left(\log \mu_{lk} + \log \theta_k \right) + \sum_{k=1}^{K} \lambda_k \left(\sum_{l=1}^{s_j} \mu_{lk} - 1 \right) \\ + & \gamma \left(\sum_{k=1}^{K} \theta_k - 1 \right) \end{aligned}$$

给定k, 求G对于λ_k和μ_{jk}的偏导，令其为零，可得

$$\begin{aligned} \frac{\partial G}{\partial \lambda_k} &= \sum_{l=1}^{s_j} \mu_{lk} - 1 = 0 \\ \frac{\partial G}{\partial \mu_{lk}} &= \frac{N_{lk}}{\mu_{lk}} + \lambda_k = 0 \end{aligned} \tag{4.3}$$

解方程组(4.3)可得

$$\begin{aligned} & \frac{N_{1k}}{\mu_{1k}} = .. = \frac{N_{S_jk}}{\mu_{S_jk}} = -\lambda_k \\ \Rightarrow \quad & \mu_{1k} + .. + \mu_{S_jk} = -\frac{1}{\lambda_k}(N_{1k} + ...N_{S_jk}) = 1 \\ \Rightarrow \quad & \lambda_k = -(N_{1k} + ...N_{S_jk}) \\ \Rightarrow \quad & \mu_{lk} = \frac{N_{lk}}{N_{1k} + ...N_{S_jk}} \end{aligned}$$

根据N_{ik}的定义可得

$$\begin{aligned} \mu_{lk} &= \frac{\sum_{i=1}^{N} \mathbb{1}_{\{x_i=a_{jl}, y_i=c_k\}}}{\sum_{i=1}^{N} \mathbb{1}_{\{x_i=a_{j1}, y_i=c_k\}} + ... + \sum_{i=1}^{N} \mathbb{1}_{\{x_i=a_{jS_j}, y_i=c_k\}}} \\ &= \frac{\sum_{i=1}^{N} \mathbb{1}_{\{x_i=a_{jl}, y_i=c_k\}}}{\sum_{i=1}^{N} \mathbb{1}_{\{y_i=c_k\}}} \end{aligned}$$

□

4.2 参数的贝叶斯估计

Exercise 4.3. $P(Y = c_k)$的贝叶斯估计是

$$P(Y = c_k) = \frac{\sum\limits_{i=1}^{N} \mathbb{1}_{y_i=c_k} + \lambda}{N + K\lambda}, k = 1...K \tag{4.4}$$

Proof. 假设$P(Y = c_k) = \theta_k$，θ_k服从Beta分布$\theta_k \sim \mathcal{B}(\alpha, \beta)$。

根据习题(1.1)的结论，$\theta_k|y_1, y_2, ...y_N \sim \mathcal{B}(\alpha + \gamma, \beta + N - \gamma)$，其中$\gamma = \sum\limits_{i=1}^{N} \mathbb{1}_{y_i=c_k}$。

$P(Y = c_k)$的贝叶斯估计为$\theta_k|y_1, y_2, ...y_N$的期望

$$\begin{aligned} \mathbb{E}(\theta_k \sim \mathcal{B}(\alpha, \beta)) &= \frac{\sum\limits_{i=1}^{N} \mathbb{1}_{y_i=c_k} + \alpha}{N + \alpha + \beta} \\ &= \frac{\sum\limits_{i=1}^{N} \mathbb{1}_{y_i=c_k} + \lambda}{N + K\lambda} \quad (\text{取}\alpha = \lambda, \beta = (K-1)\lambda) \end{aligned}$$

□

Exercise 4.4. $P(X^j = a_{jl}|Y = c_k)$的贝叶斯估计是

$$P(X^j = a_{jl}|Y = c_k) = \frac{\sum\limits_{i=1}^{N} \mathbb{1}_{x_i^j=a_{jl}, y_i=c_k} + \lambda}{\sum\limits_{i=1}^{N} \mathbb{1}_{y_i=c_k} + S_j\lambda}$$

同理可证，此处略去

5 决策树

5.1 利用C4.5算法生成决策树

Exercise 5.1. 书中表*5.1*是一个由*15*个样本组成的贷款申请训练集，用信息增益比来生成决策树

Solution 5.1.

用D表示训练集，A_1, A_2, A_3, A_4表示年龄、有工作、有自己的房子和信贷情况4个特征。那么

1. 经验熵$H(D) = -\frac{9}{15}\log_2\frac{9}{15} - \frac{6}{15}\log_2\frac{6}{15} = 0.971$
2. - $g(D, A_1) = H(D) - \left[\frac{5}{15}H(D_1) + \frac{5}{15}H(D_2) + \frac{5}{15}H(D_3)\right] = 0.083$
   - $H_{A_1}(D) = -\frac{5}{15}\log_2\frac{5}{15} - \frac{5}{15}\log_2\frac{5}{15} - \frac{5}{15}\log_2\frac{5}{15} = 1.585$
   - $g_R(D, A_1) = \frac{g(D,A_1)}{H_{A_1}(D)} = 0.052$
3. - $g(D, A_2) = H(D) - \left[\frac{5}{15}H(D_1) + \frac{10}{15}H(D_2)\right] = 0.324$
   - $H_{A_2}(D) = -\frac{5}{15}\log_2\frac{5}{15} - \frac{10}{15}\log_2\frac{10}{15} = 0.918$
   - $g_R(D, A_2) = \frac{g(D,A_2)}{H_{A_2}(D)} = 0.353$
4. - $g(D, A_3) = H(D) - \left[\frac{6}{15}H(D_1) + \frac{9}{15}H(D_2)\right] = 0.42$
   - $H_{A_3}(D) = -\frac{6}{15}\log_2\frac{6}{15} - \frac{9}{15}\log_2\frac{9}{15} = 0.971$
   - $g_R(D, A_3) = \frac{g(D,A_3)}{H_{A_3}(D)} = 0.432$
5. - $g(D, A_4) = H(D) - \left[\frac{5}{15}H(D_1) + \frac{6}{15}H(D_2) + \frac{4}{15}H(D_3)\right] = 0.363$
   - $H_{A_4}(D) = -\frac{5}{15}\log_2\frac{5}{15} - \frac{6}{15}\log_2\frac{6}{15} - \frac{4}{15}\log_2\frac{4}{15} = 1.499$
   - $g_R(D, A_4) = \frac{g(D,A_4)}{H_{A_4}(D)} = 0.242$

   $g_R(D, A_3)$最大，以A_3划分特征空间。

表 1: 无房子的信贷申请样本数据

ID	年龄	有工作	信贷	类别
1	青	否	一般	否
2	青	否	好	否
3	青	是	好	是
5	青	否	一般	否
6	中	否	一般	否
7	中	否	好	否
13	老	是	好	是
14	老	是	非常好	是
15	老	否	一般	否

有房子的数据集所有类别都相同，已经完成了分类。对于无房子的情况(见表(1))

1. 经验熵$H(D)=-\frac{3}{9}\log_2\frac{3}{9}-\frac{6}{9}\log_2\frac{6}{9}=0.918$
2. - $g(D,A_1)=H(D)-\left[\frac{4}{9}H(D_1)+\frac{2}{9}H(D_2)+\frac{3}{9}H(D_3)\right]=0.251$
   - $H_{A_1}(D)=-\frac{4}{9}\log_2\frac{4}{9}-\frac{2}{9}\log_2\frac{2}{9}-\frac{3}{9}\log_2\frac{3}{9}=1.530$
   - $g_R(D,A_1)=\frac{g(D,A_1)}{H_{A_1}(D)}=0.164$
3. - $g(D,A_2)=H(D)-\left[\frac{3}{9}H(D_1)+\frac{6}{9}H(D_2)\right]=0.918$
   - $H_{A_2}(D)=-\frac{3}{9}\log_2\frac{3}{9}-\frac{6}{9}\log_2\frac{6}{9}=0.918$
   - $g_R(D,A_2)=\frac{g(D,A_2)}{H_{A_2}(D)}=1.0$
4. - $g(D,A_4)=H(D)-\left[\frac{4}{9}H(D_1)+\frac{4}{9}H(D_2)+\frac{1}{9}H(D_3)\right]=0.473$
   - $H_{A_4}(D)=-\frac{4}{9}\log_2\frac{4}{9}-\frac{4}{9}\log_2\frac{4}{9}-\frac{1}{9}\log_2\frac{1}{9}=1.392$
   - $g_R(D,A_4)=\frac{g(D,A_4)}{H_{A_4}(D)}=0.34$

$g_R(D,A_2)$最大，以A_2进一步划分特征空间，结果(见表(2)(3))，所有数据都已经得到了正确的分类，故决策树构建完成。

表 2: 无工作的信贷申请样本数据

ID	年龄	信贷	类别
1	青	一般	否
2	青	好	否
5	青	一般	否
6	中	一般	否
7	中	好	否
15	老	一般	否

表 3: 有工作的信贷申请样本数据

ID	年龄	信贷	类别
3	青	好	是
13	老	好	是
14	老	非常好	是

5.2 二叉回归树

Exercise 5.2. 根据如下训练数据，用平方误差准则生成一个二叉回归树

Solution 5.2.

表 4: 训练数据表

x_i	1	2	3	4	5	6	7	8	9	10
y_i	4.5	4.75	4.91	5.34	5.8	7.05	7.9	8.23	8.7	9.0

最小二乘回归树的损失函数是

$$L(j,s)=\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2$$

此处算法以变量x_i的值作为切分点。

1. 遍历$x_1\sim x_9$得到对应的损失函数值列表，见表(5)：使得损失函数最小的点是x_5，$c_1=5.06, c_2=8.176$，得到第一个回归树

$$T_1(x)=5.06\mathbb{1}_{x\leq 5}+8.176\mathbb{1}_{x>5}$$

表 5: 遍历$x_1\sim x_9$得到的损失函数值

切分点x_i	1	2	3	4	5	6	7	8	9
损失函数	22.65	17.70	12.19	7.37	3.35	5.07	10.05	15.17	21.32

2. 对左右子树分别再次进行遍历计算，得到两张子表

表 6: 遍历$x_1\sim x_4$得到的损失函数值

切分点x_i	1	2	3	4
损失函数	0.66	0.42	0.19	0.37

表 7: 遍历$x_6\sim x_9$得到的损失函数值

切分点x_i	6	7	8	9
损失函数	0.715	0.662	0.786	1.451

使得左右两个子树损失函数最小的点分别是x_3, x_7，对应的y_i均值分别为$c_1 = 4.72, c_2 = 5.57$和$c_1 = 7.47, c_2 = 8.64$。由此得到两个子回归树

$$T_2(x) = 4.72\mathbb{1}_{x\leq 3} + 5.57\mathbb{1}_{5\geq x>3}$$
$$T_3(x) = 7.47\mathbb{1}_{5<x\leq 7} + 8.64\mathbb{1}_{10\geq x>7}$$

3. 对上一步中左子树以x_3为根节点分别得到新的左右子树，新的右子树仅包含两个数据点，无需再划分，新的左子树利用同样算法可得到损失函数表

表 8: 遍历$x_1 \sim x_2$得到的损失函数值

切分点x_i	1	2
损失函数	0.012	0.031

同理对上一步中右子树以x_7为根节点分别得到新的左右子树，新的左子树仅包含两个节点，无需再划分，新的右子树对应的损失函数表为

表 9: 遍历$x_8 \sim x_9$得到的损失函数值

切分点x_i	8	9
损失函数	0.045	0.011

根据表(8)和(11)可知新的划分点为x_1和x_8，以此两点得到新的左右子树的均值分别为$c_1 = 4.5, c_2 = 4.83$和$c_1 = 8.23, c_2 = 8.85$，加上两棵无需处理的子树，一共得到四棵回归子树

$$\begin{aligned} T_4(x) &= 4.51\mathbb{1}_{x\leq 1} + 4.83\mathbb{1}_{1<x\leq 7} \\ T_5(x) &= 5.57\mathbb{1}_{3<x\leq 5} \\ T_6(x) &= 7.47\mathbb{1}_{5<x\leq 7} \\ T_7(x) &= 8.23\mathbb{1}_{7<x\leq 8} + 8.85\mathbb{1}_{8<x\leq 10} \end{aligned}$$

综上所述，最终得到的回归树为

$$\begin{aligned} T_(x) &= T_4(x) + T_5(x) + T_6(x) + T_7(x) \\ &= 4.51\mathbb{1}_{x\leq 1} + 4.83\mathbb{1}_{1<x\leq 7} + 5.57\mathbb{1}_{3<x\leq 5} + 7.47\mathbb{1}_{5<x\leq 7} + 8.23\mathbb{1}_{7<x\leq 8} + 8.85\mathbb{1}_{8<x\leq 10} \end{aligned}$$

5.3 最小子树的唯一性

Exercise 5.3. *求证在CART剪枝算法中，当α确定时，存在唯一的最小子树T_α使得损失函数$C_\alpha(T)$最小。*

Solution 5.3.

Proof. 1. 存在性：对于任意一棵树，子树的数目是有限的，每个子树T_i对应一个损失函数值$C_\alpha(T_i) = C(T_i) + \alpha|T_i|$，对于固定的$\alpha$，总能找到至少一个最小值，对应的子树就是最小子树。

2. 唯一性：假设最小子树不唯一，至少存在两个子树，不妨假设为T_1和T_2，同时使得损失函数最小。

   T_1和T_2必存在一些不相同的叶节点。假设叶节点$t \in T_1, t \notin T_2$, 那么必存在以某结点t^* 为根节点，以t为某一叶结点的子树T_t*在T_1而不在T_2 中。

   在T_1的剪枝过程中T_t*未被剪除，说明在t^*节点，以结点t^* 为根节点的子树的损失函数$C(T_t^*)$小于等于以t^*为单节点的树的损失函数$C(t^*)$，即$C(T_t^*) \leq C(t^*)$，而在T_2剪枝的过程中T_t*被剪除，则有$C(T_t^*) > C(t^*)$。存在矛盾。故最小子树不可能出现不唯一的情况。

□

5.4 最优子树序列

Exercise 5.4. *求证在$CART$剪枝算法中求出的子树序列$\{T_0, T_1, ..., T_n\}$分别是区间$\alpha \in [\alpha_i, \alpha_{i+1})$的最优子树$T_\alpha$，这里$i = 0, 1, ...n, 0 = \alpha_0 < \alpha_1 < ... < \alpha_n < \infty$*

Solution 5.4.

Proof. 用数学归纳法证明本题：

1. 根据剪枝算法，$\alpha_1 = \min\{g(t_1), g(t_2), ...g(t_n)\}$，即$\alpha_1$等于原始树$T$的所有内部节点对应的$g(t)$的最小值。那么当$\alpha \in [0, \alpha_1)$时，对于任意内部节点$t$都有

$$\begin{aligned} & g(t) > \alpha \\ \Rightarrow \quad & \frac{C(t) - C(T_t)}{|T_t| - 1} > \alpha \\ \Rightarrow \quad & C(t) + \alpha > C(T_t) + \alpha|T_t| \end{aligned} \tag{5.1}$$

   由公式(5.1)可以看到，以t为单节点树的损失函数比以t为根节点的子树损失函数要大，故对于任意内部节点t都不满足剪枝的条件。故$T_0 = T$就是当$\alpha \in [0, \alpha_1)$时对应的最优子树。

2. 假设T_i为区间$\alpha \in [\alpha_i, \alpha_{i+1})$上的最优子树。根据剪枝算法，可知

   - 当$\alpha = \alpha_{i+1}$，在T_i上剪枝生成子树T_{i+1}
   - 当$\alpha \in (\alpha_{i+1}, \alpha_{i+2})$时，因为$\alpha_{i+2}$就是$T_{i+1}$上任意内部节点$t$求得的$g(t)$的最小值，所以对于$T_{i+1}$上的任意内部节点$t$ 都有$g(t) > \alpha$，根据公式(5.1)同理可知，对于T_{i+1}上任意内部节点t都有$C(t)+\alpha > C(T_t) + \alpha|T_t|$，即不满足剪枝条件

   所以T_{i+1}为区间$\alpha \in [\alpha_{i+1}, \alpha_{i+2})$上的最优子树。

3. 综上所知，对于$\forall i = 0, 1, ..., n$，T_i为区间$\alpha \in [\alpha_i, \alpha_{i+1})$上的最优子树。

□

6 逻辑斯蒂回归与最大熵模型

6.1 指数分布族

Exercise 6.1. 确认逻辑斯蒂分布属于指数分布族

无法证明，可能题目有误。

6.2 逻辑斯蒂模型学习的梯度下降算法

Exercise 6.2. 写出逻辑斯蒂模型学习的梯度下降算法

Solution 6.1.

给定的训练数据集$T=\{(x_1,y_1),...,(x_N,y_N)\}$，$x_i\in\mathcal{X}=\mathbb{R}^n$，$y_i\in\{0,1\}$，逻辑斯蒂回归模型是如下的条件概率分布

$$\mathbb{P}(Y=1|x)=\frac{\exp(w\cdot x+b)}{1+\exp(w\cdot x+b)}$$
$$\mathbb{P}(Y=0|x)=\frac{1}{1+\exp(w\cdot x+b)}$$

其中$w\in\mathbb{R}^n,b\in\mathbb{R}$是参数。为表示方便，将权值向量$w$和输入向量$x$加以扩充，仍记做$w,x$，即$w=\left(w^1,w^2...w^n,b\right)^T,x=\left(x^1,x^2,...x^n,1\right)^T$，那么逻辑斯蒂回归模型可以表示为

$$\mathbb{P}(Y=1|x)=\frac{\exp(w\cdot x)}{1+\exp(w\cdot x)}$$
$$\mathbb{P}(Y=0|x)=\frac{1}{1+\exp(w\cdot x)}$$

令$\pi(x)=\frac{\exp(w\cdot x)}{1+\exp(w\cdot x)}$，那么$P(Y=1|x)=\pi(x),P(Y=0|x)=1-\pi(x)$，似然函数为

$$L(w)=\prod_{i=1}^{N}\left[\pi(x_i)\right]^{y_i}\left[1-\pi(x_i)\right]^{1-y_i}$$

对数似然函数

$$\begin{aligned}\log L(w) &= \sum_{i=1}^{N}\left(y_i\log\pi(x_i)+(1-y_i)\log(1-\pi(x_i))\right)\\ &= \sum_{i=1}^{N}\left(y_i\log\frac{\pi(x_i)}{1-\pi(x_i)}+\log(1-\pi(x_i))\right)\\ &= \sum_{i=1}^{N}\left(y_i(w\cdot x_i)-\log(1+\exp(w\cdot x_i))\right)\end{aligned}\tag{6.1}$$

根据公式(6.1)，对数似然函数对w的偏导为

$$\begin{aligned}\frac{\partial}{\partial w}\log L(w) &= \sum_{i=1}^{N}\left(y_ix_i-\frac{\exp(w\cdot x_i)x_i}{1+\exp(w\cdot x_i)}\right)\\ &= \sum_{i=1}^{N}\left(y_i-\pi(x_i)\right)x_i\end{aligned}$$

由此此处求对数似然函数的最大值，故需要沿着梯度上升的方向进行迭代，迭代公式为

$$\begin{aligned} w_{t+1} &= w_t + \alpha \frac{\partial}{\partial w} \log L(w) \\ &= w_t + \alpha \sum_{i=1}^{N} (y_i - \pi(x_i))\, x_i \end{aligned} \tag{6.2}$$

其中α称为学习率，是一个$n+1$维正常数。
设置合适的学习率α以及初值w_0后，根据公式(6.2)进行迭代直至收敛为止。

6.3 最大熵模型学习的DFP算法

Exercise 6.3. 写出最大熵模型学习的*DFP*算法

Solution 6.2.

算法流程与原书中算法6.2一致，唯一的区别在与步骤(6)的迭代公式更换为

$$B_{k+1} = B_k + \frac{\delta_k \delta_k^T}{\delta_k^T y_k} - \frac{B_k y_k y_k^T B_k}{y_k^T B_k y_k}$$

7 支持向量机

7.1 感知机和支持向量机的比较

Exercise 7.1. 比较感知机的对偶形式与线性可分支持向量机的对偶形式

Solution 7.1.

设训练集$T=\{(x_1,y_1),(x_2,y_2)...(x_N,y_N)\}$，其中$x_i\in\mathcal{X}=\mathbb{R}^n, y_i\in\mathcal{Y}=\{-1,1\}, i=1,2...N$。

1. 感知机模型的决策函数

假设$\alpha^*=(\alpha_1^*,\alpha_2^*,...\alpha_N^*)$是下面最优化问题的解

$$\begin{aligned}&\min_{\alpha}-\sum_{x_i\in M}y_i\left(\sum_{i=1}^{N}\alpha_i y_i x_i\cdot x+\sum_{i=1}^{N}\alpha_i y_i\right)\\&\alpha_i\geq 0, i=1,2,...N\end{aligned}$$

从而得到感知机模型的决策函数为

$$f(x)=\text{sign}\left(\sum_{i=1}^{N}\alpha_i^* y_i x_i\cdot x+\sum_{i=1}^{N}\alpha_i^* y_i\right)$$

2. 线性可分支持向量机

假设$\alpha^*=(\alpha_1^*,\alpha_2^*,...\alpha_N^*)$是下面最优化问题的解

$$\begin{aligned}&\min_{\alpha}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)-\sum_{i=1}^{N}\alpha_i\\&s.t.\quad\sum_{i=1}^{N}\alpha_i y_i=0\\&\alpha_i\geq 0, i=1,2,...N\end{aligned}$$

那么存在下标j，使得$\alpha_j^*>0$，那么线性可分支持向量机的决策函数可以表示为

$$f(x)=\text{sign}\left(\sum_{i=1}^{N}\alpha_i^* y_i x_i\cdot x+y_j-\sum_{i=1}^{N}\alpha_i^* y_i\right)$$

7.2 支持向量机求解

Exercise 7.2. 已知正例点$x_1=(1,2)^T, x_2=(2,3)^T, x_3=(3,3)^T$，负例点$x_4=(2,1)^T, x_5=(3,2)^T$。试求最大间隔分离超平面和分类决策函数。

Solution 7.2.

根据训练集构造最优化问题

$$\begin{aligned}\min_{w,b} \quad & \frac{1}{2}(w_1^2+w_2^2)\\ s.t. \quad & w_1+2w_2+b\geq 1\\ & 2w_1+3w_2+b\geq 1\\ & 2w_1+3w_2+b\geq 1\\ & -2w_1-w_2-b\geq 1\\ & -3w_1-2w_2-b\geq 1\end{aligned}$$

求得此优化问题的解为$w_1=-1, w_2=2, b=-2$，故可得到最大分割超平面为

$$-x^{(1)}+2x^{(2)}-2=0$$

其中支持向量为$(1,2)^T, (3,3)^T, (3,2)^T$

7.3 软间隔支持向量机

Exercise 7.3. 线性支持向量机还可以定义为

$$\begin{aligned}\min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}\xi_i\\ s.t. \quad & y_i\left(w\cdot x_i+b\right)\geq 1-\xi_i, i=1,2,...,N\\ & \xi_i\geq 0, i=1,2,...,N\end{aligned}$$

求其对偶形式

Solution 7.3.

定义原问题的拉格朗日函数为

$$L(w,b,\xi,\alpha,\mu)=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}\xi_i-\sum_{i=1}^{N}\alpha_i\left(y_i\left(w\cdot x_i+b\right)-1+\xi_i\right)-\sum_{i=1}^{N}\mu_i\xi_i \tag{7.1}$$

其中$\alpha_i\geq 0, \mu_i\geq 0$

分别求L对w, b, ξ_i的偏导，令其等于零，得到

$$\begin{aligned}\frac{\partial L(w,b,\xi,\alpha,\mu)}{\partial w} &= w-\sum_{i=1}^{N}\alpha_i y_i x_i=0\\ \frac{\partial L(w,b,\xi,\alpha,\mu)}{\partial b} &= -\sum_{i=1}^{N}\alpha_i x_i=0\\ \frac{\partial L(w,b,\xi,\alpha,\mu)}{\partial \xi_i} &= C-\alpha_i-\mu_i=0\end{aligned}$$

进而解得

$$
\begin{aligned}
&w = \sum_{i=1}^{N} \alpha_i y_i x_i \\
&\sum_{i=1}^{N} \alpha_i y_i = 0 \\
&C = \alpha_i + \mu_i, i = 1, 2, ...N
\end{aligned}
$$

代入公式(7.1)中，得到拉格朗日函数的极小值

$$
\begin{aligned}
\min_{w,b,\xi} L(w, b, \xi, \alpha, \mu) &= \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\left(y_i\left(w \cdot x_i + b\right) - 1 + \xi_i\right) - \sum_{i=1}^{N}\mu_i\xi_i \\
&= \frac{1}{2}\left(\sum_{i=1}^{N}\alpha_i y_i x_i\right) \cdot \left(\sum_{i=1}^{N}\alpha_i y_i x_i\right) + \sum_{i=1}^{N}\left(\alpha_i + \mu_i\right)\xi_i \\
&\quad - \sum_{i=1}^{N}\alpha_i y_i (\sum_{j=1}^{N}\alpha_j y_j x_j) \cdot x_i - b\sum_{i=1}^{N}\alpha_i y_i + \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\alpha_i\xi_i - \sum_{i=1}^{N}\mu_i\xi_i \\
&= -\frac{1}{2}\left(\sum_{i=1}^{N}\alpha_i y_i x_i\right) \cdot \left(\sum_{i=1}^{N}\alpha_i y_i x_i\right) + \sum_{i=1}^{N}\alpha_i \\
&= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^{N}\alpha_i
\end{aligned}
$$

再对上述结果求极大值，得到对偶问题

$$
\begin{aligned}
\max_{\alpha} \quad & -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^{N}\alpha_i \\
s.t. \quad & \sum_{i=1}^{N}\alpha_i y_i = 0 \\
& 0 \le \alpha_i \le C
\end{aligned}
$$

其中约束$0 \le \alpha_i \le C$可由约束$\alpha_i \ge 0, \mu_i \ge 0, C = \alpha_i + \mu_i$推导出。

7.4 正定核函数

[支持向量机（三）核函数，http://www.cnblogs.com/jerrylead/archive/2011/03/18/1988406.html] [核函数的定义与作用，https://www.zhihu.com/question/24627666]

Exercise 7.4. 求证内积的正整数幂函数$K(x, z) = (x \cdot z)^p$是正定核函数，其中p为正整数，$x, z \in \mathbb{R}^n$

Proof. 设$x = (x_1, x_2, ...x_n)^T, z = (z_1, z_2, ...z_n)^T$，于是可以得到

$$
\begin{aligned}
K(x,z) &= (x\cdot z)^p \\
&= (\sum_{i=1}^{n} x_i z_i)^p \\
&= \sum_{k_1+k_2+\ldots+k_n=p} \frac{p!}{k_1!k_2!\ldots k_n!}\left(x_1^{k_1}x_2^{k_2}\ldots x_n^{k_n}\right)\left(z_1^{k_1}z_2^{k_2}\ldots z_n^{k_n}\right) \quad \text{(根据多项式定理)} \\
&= \left(\sqrt{\frac{p!}{k_1!k_2!\ldots k_n!}}x_1^{k_1}x_2^{k_2}\ldots x_n^{k_n}\right)\cdot\left(\sqrt{\frac{p!}{k_1!k_2!\ldots k_n!}}z_1^{k_1}z_2^{k_2}\ldots z_n^{k_n}\right), \sum_{i=1}^{n} k_i = p \\
&= \phi(x)\cdot\phi(z)
\end{aligned}
$$

其中$\phi(x)$是一个列向量，行元素的表达式为$\sqrt{\frac{p!}{k_1!k_2!\ldots k_n!}}x_1^{k_1}x_2^{k_2}\ldots x_n^{k_n}$，行维度等于服从$k_1+k_2+\ldots+k_n=p$约束的非负向量$(k_1,k_2\ldots k_n)$的数目。下面讨论此非负向量$(k_1,k_2\ldots k_n)$有多少种情况

1. 如果$n>p$，问题等价于p个相同的球放入n个盒子中的组合数，每个球有n种选择，总的组合数等于n^p

2. 如果$n\le p$，问题等价于p个相同的球排成一列，首尾以及中间插入$n-1$个隔板分成n份(考虑到k_i可以为零)，这样问题进一步等价于p 个相同的球和$n-1$ 个相同的隔板排成一列，有多少种组合的方式。解答为$\frac{(p+n-1)!}{(p)!(n-1)!}=C_{p+n-1}^{p}$

综上所述，可以找到特征空间$\mathcal{H}=\mathbb{R}^d, d=\mathbb{1}_{n>p}n^p+\mathbb{1}_{n\le p}C_{p+n-1}^{p}$以及映射$\phi(x):\mathbb{R}^n\to\mathcal{H}$，使得$K(x,z)=(x\cdot z)^p=\phi(x)\cdot\phi(z)$，满足正定核的充要条件，故$K(x,z)=(x\cdot z)^p$为正定核。

□

8 提升方法

8.1 Adaboost学习

Exercise 8.1. 假设弱分类器为决策树桩，用*Adaboost*学习一个强分类器

表 10: 应聘人员情况数据表

	1	2	3	4	5	6	7	8	9	10
身体	0	0	1	1	1	0	1	1	1	0
业务	1	3	2	1	2	1	1	1	3	2
潜力	3	1	2	3	3	2	2	1	1	1
分类	-1	-1	-1	-1	-1	-1	1	1	-1	-1

Solution 8.1.

用x, y, z分别表示代表身体，业务和潜力的数值向量。基本分类器定义为针对其中某一维度的决策树桩，故迭代过程中，需要对所有维度进行循环并挑选出误差最低的分类器，进行下一步循环。

另外决策树桩还有方向，定义阈值为v的正树桩为

$$G(x) = \begin{cases} 1, & x > v \\ -1, & x \leq v \end{cases}$$

阈值为v的负树桩为

$$G(x) = \begin{cases} 1, & x \leq v \\ -1, & x > v \end{cases}$$

针对本例的Adaboost算法可以描述为

1. 迭代第一步：
   - 初始化权值分布$D_1 = (w_{11}, w_{12}, ...w_{110}) = (0.1, 0.1, ..., 0.1)$
   - 在权值分布为D_1的训练数据上，对身体维度取阈值$v = -0.1$时的负树桩分类误差最低，故基本分类器为
   $$G_1(x) = \begin{cases} 1, & x \leq -0.1 \\ -1, & x > -0.1 \end{cases}$$
   - $G_1(x)$误差率为0.2, $\alpha_1 = \frac{1}{2}\log\frac{1-e_1}{e_1} = 0.6931$
   - 更新的权值分布为$D_2 = (0.0625, 0.0625, 0.0625, 0.0625, 0.0625, 0.0625, 0.25, 0.25, 0.0625, 0.0625)$
   - 分类器$f = \text{sign}[0.6931G_1(x)]$，在训练集上有2个误分类点。
2. 迭代第二步：

- 在权值分布为D_2的训练数据上，对业务维度取阈值$v=1.0$时的负树桩分类误差最低，故基本分类器为

$$G_2(y)=\begin{cases}1, & y\le 1\\ -1, & y>1\end{cases}$$

- $G_2(x)$误差率为$e_2=0.1875$, $\alpha_2=\frac{1}{2}\log\frac{1-e_2}{e_2}=0.7331$
- 更新的权值分布为$D_3=(0.1666,0.0384,0.0384,0.1666,0.0384,0.1666,0.1538,0.1538,0.0384,0.0384)$
- 分类器$f=\text{sign}[0.6931G_1(x)+0.7331G_2(y)]$，在训练集上有3个误分类点。

3. 迭代第三步：

- 在权值分布为D_3的训练数据上，对潜力维度取阈值$v=1.0$时的负树桩分类误差最低，故基本分类器为

$$G_3(z)=\begin{cases}1, & y\le 0\\ -1, & y>1\end{cases}$$

- $G_3(x)$误差率为$e_3=0.2692$, $\alpha_3=\frac{1}{2}\log\frac{1-e_3}{e_3}=0.4992$
- 更新的权值分布为$D_4=(0.1140,0.0714,0.0263,0.1140,0.0263,0.1140,0.2857,0.1052,0.0714,0.0714)$
- 分类器$f=\text{sign}[0.6931G_1(x)+0.7331G_2(y)+0.4992G_3(z)]$，在训练集上有1个误分类点。

4. 迭代第四步：

- 在权值分布为D_4的训练数据上，对身体维度取阈值$v=0.0$时的正树桩分类误差最低，故基本分类器为

$$G_4(x)=\begin{cases}1, & x>0\\ -1, & x\le 0\end{cases}$$

- $G_4(x)$误差率为$e_4=0.2380$, $\alpha_4=\frac{1}{2}\log\frac{1-e_4}{e_4}=0.5815$
- 更新的权值分布为$D_5=(0.0748,0.0468,0.0552,0.2394,0.0552,0.0748,0.1875,0.0690,0.15,0.0468)$
- 分类器$f=\text{sign}[0.6931G_1(x)+0.7331G_2(y)+0.4992G_3(z)+0.5815G_4(x)]$，在训练集上有1个误分类点。

5. 迭代第五步：

- 在权值分布为D_5的训练数据上，对身体维度取阈值$v=-0.1$时的负树桩分类误差最低，故基本分类器为

$$G_5(x)=\begin{cases}1, & x\le -0.1\\ -1, & x>-0.1\end{cases}$$

- $G_5(x)$误差率为$e_5=0.2565$, $\alpha_5=\frac{1}{2}\log\frac{1-e_5}{e_5}=0.5319$
- 更新的权值分布为$D_6=(0.0503,0.0315,0.0371,0.1610,0.0371,0.0503,0.3653,0.1346,0.1008,0.0315)$
- 分类器$f=\text{sign}[0.6931G_1(x)+0.7331G_2(y)+0.4992G_3(z)+0.5815G_4(x)+0.5319G_5(x)]$，在训练集上有1个误分类点。

6. 迭代第六步：

   - 在权值分布为D_6的训练数据上，对潜力维度取阈值$v = 2.0$时的负树桩分类误差最低，故基本分类器为

$$G_6(z) = \begin{cases} 1, & x \leq 2.0 \\ -1, & x > 2.0 \end{cases}$$

   - $G_6(x)$误差率为0.2514, $\alpha_6 = \frac{1}{2}\log\frac{1-e_6}{e_6} = 0.5454$
   - 更新的权值分布为$D_7 = (0.0748, 0.0468, 0.0552, 0.2394, 0.0552, 0.0748, 0.1875, 0.0690, 0.15, 0.0468)$
   - 分类器$f = \text{sign}[0.6931G_1(x) + 0.7331G_2(y) + 0.4992G_3(z) + 0.5815G_4(x) + 0.5319G_5(x) + 0.5454G_6(z)]$，在训练集上已经没有误分类点。

代码参见机器学习实战第七章源代码。

8.2 学习策略与算法比较

Exercise 8.2. 比较支持向量机、*Adaboost*、逻辑斯蒂回归模型的学习策略与方法

Solution 8.2.

表 11: 学习策略与算法比较

方法	学习策略	算法
支持向量机	极小化正则合页损失(等价于软间隔最大化)	SMO
Adaboost	极小化加法模型的指数损失	前向分布加法算法
逻辑斯蒂回归	极大似然估计	拟牛顿法，梯度下降法

9 EM算法及其推广

9.1 三硬币模型的极大似然估计

Solution 9.1.

Exercise 9.1. 不同初值对三硬币模型参数估计的影响。

当初值为$\theta = (0.46, 0.55, 0.67)$时，$\theta = (\pi, p, q)$的极大似然估计为$(0.46, 0.55, 0.67)$，代码参见附录(12.1)

9.2 证明引理9.2

Exercise 9.2. 证明引理*9.2*。

Proof. 如果$\widetilde{\mathbb{P}}_\theta(Z) = \mathbb{P}(Z|Y,\theta)$，那么根据F函数的定义，

$$
\begin{aligned}
F(\widetilde{\mathbb{P}}, \theta) &= \mathbb{E}_{\widetilde{\mathbb{P}}}[\log \mathbb{P}(Y, Z|\theta)] - \mathbb{E}_{\widetilde{\mathbb{P}}} \log \widetilde{\mathbb{P}}_\theta(Z) \\
&= \sum_Z \widetilde{\mathbb{P}}_\theta(Z) \log \mathbb{P}(Y, Z|\theta) - \sum_Z \widetilde{\mathbb{P}}_\theta(Z) \log \widetilde{\mathbb{P}}_\theta(Z) \\
&= \sum_Z \mathbb{P}(Z|Y,\theta) \log \mathbb{P}(Y, Z|\theta) - \sum_Z \mathbb{P}(Z|Y,\theta) \log \mathbb{P}(Z|Y,\theta) \\
&= \sum_Z \mathbb{P}(Z|Y,\theta) \log[\mathbb{P}(Z|Y,\theta)\mathbb{P}(Y|\theta)] - \sum_Z \mathbb{P}(Z|Y,\theta) \log \mathbb{P}(Z|Y,\theta) \\
&= \sum_Z \mathbb{P}(Z|Y,\theta) \log \mathbb{P}(Y|\theta) + \sum_Z \mathbb{P}(Z|Y,\theta) \log \mathbb{P}(Z|Y,\theta) - \sum_Z \log \mathbb{P}(Z|Y,\theta)\mathbb{P}(Z|Y,\theta) \\
&= \sum_Z \mathbb{P}(Z|Y,\theta) \log \mathbb{P}(Y|\theta) \\
&= \log \mathbb{P}(Y|\theta) \sum_Z \mathbb{P}(Z|Y,\theta) \quad \big(\log \mathbb{P}(Y|\theta) \text{与} Z \text{无关}\big) \\
&= \log \mathbb{P}(Y|\theta) \quad \left(\text{累计概率和为1，即} \sum_Z \mathbb{P}(Z|Y,\theta) = 1\right)
\end{aligned}
$$

□

9.3 GMM参数估计

Exercise 9.3. 已知观测数据$-67, -48, 6, 8, 14, 16, 23, 24, 28, 29, 41, 49, 56, 60, 75$，试求两个分量的高斯混合模型的*5*个参数

9.4 朴素贝叶斯的非监督学习

[斯坦福ML公开课笔记13A，http://blog.csdn.net/stdcoutzyx/article/details/27368507]
[EM算法，http://blog.csdn.net/zouxy09/article/details/8537620]

Exercise 9.4. 将*EM*算法应用到朴素贝叶斯的非监督学习上，写出其算法。

Solution 9.2.

本题描述一个文本聚类成两个主题的混合贝叶斯模型。

假设观测数据集为$\{y_1, y_2, ...y_N\}$，每个数据均是n维的由0或者1组成的向量，表示一个文本的单词组成。样本i对应的观测数据y_i的第j个分量$y_i^{(j)}$，表示单词j 是否在文本i 中。

假设决定文本i聚类主题的隐变量z_i服从参数为ϕ的伯努利分布

$$\mathbb{P}(z_i = 1) = \phi, \phi \in [0, 1]$$

$z_i = 1$表示样本i属于主题1，$z_i = 0$表示样本i属于主题0。$\{z_1, z_2, ...z_N\}$独立同分布。

观测数据的似然函数是

$$\begin{aligned} L(y_1, y_2, ...y_N; \phi) &= \sum_{i=1}^{N} \log \mathbb{P}(y_i; \phi) \\ &= \sum_{i=1}^{N} \log \sum_{z_i=0}^{1} \mathbb{P}(y_i, z_i; \phi) \\ &= \sum_{i=1}^{N} \log \sum_{z_i=0}^{1} Q_i(z_i) \frac{\mathbb{P}(y_i, z_i; \phi)}{Q_i(z_i)} \\ &\geq \sum_{i=1}^{N} \sum_{z_i=0}^{1} Q_i(z_i) \log \frac{\mathbb{P}(y_i, z_i; \phi)}{Q_i(z_i)} \end{aligned}$$

根据Jensen不等式以及$Q_i(z_i)$是随机变量z_i的概率密度函数的假设，可得$Q_i(z_i) = \mathbb{P}(z_i|y_i; \phi)$（推导参见李航教材，此处暂略）

应用EM算法求解

- E步：计算Q函数

$$Q_i(z_i = 1) = \mathbb{P}(z_i = 1|y_i; \phi) = \frac{\mathbb{P}(y_i|z_i = 1)\mathbb{P}(z_i = 1; \phi)}{\sum_{j=0}^{1} \mathbb{P}(y_i|z_i = j)\mathbb{P}(z_i = j; \phi)}$$

令$\phi_{j|z=1} = \mathbb{P}(y^{(j)} = 1|z = 1), \phi_{j|z=0} = \mathbb{P}(y^{(j)} = 1|z = 0)$，因为$\{y_i\}, \{z_i\}$均是独立同分布序列，故此处忽略下标$i$。根据朴素贝叶斯的性质，可知

$$\begin{aligned} \mathbb{P}(y_i|z_i = 1) &= \prod_{j=1}^{n} \mathbb{P}(y_i^{(j)}|z_i = 1) \\ &= \prod_{j=1}^{n} (\phi_{j|z_i=1})^{\mathbb{1}_{y_i^{(j)}=1}} (1 - \phi_{j|z_i=1})^{\mathbb{1}_{y_i^{(j)}=0}} \end{aligned} \tag{9.1}$$

同理，可得

$$\mathbb{P}(y_i|z_i = 0) = \prod_{j=1}^{n} (\phi_{j|z_i=0})^{\mathbb{1}_{y_i^{(j)}=1}} (1 - \phi_{j|z_i=0})^{\mathbb{1}_{y_i^{(j)}=0}} \tag{9.2}$$

把公式(9.1)(9.2)，Q函数可以进一步表示为

$$Q_i(z_i = 1) = \frac{\phi \prod_{j=1}^{n} (\phi_{j|z_i=1})^{\mathbb{1}_{y_i^{(j)}=1}} (1 - \phi_{j|z_i=1})^{\mathbb{1}_{y_i^{(j)}=0}}}{\prod_{j=1}^{n} (\phi_{j|z_i=1})^{\mathbb{1}_{y_i^{(j)}=1}} (1 - \phi_{j|z_i=1})^{\mathbb{1}_{y_i^{(j)}=0}} + (1 - \phi) \prod_{j=1}^{n} (\phi_{j|z_i=0})^{\mathbb{1}_{y_i^{(j)}=1}} (1 - \phi_{j|z_i=0})^{\mathbb{1}_{y_i^{(j)}=0}}}$$

- M步：求解

$$\arg\max \sum_{i=1}^{N}\sum_{z_i=0}^{1} Q_i(z_i)\log\frac{\mathbb{P}(y_i,z_i;\phi)}{Q_i(z_i)}$$

目标函数可进一步转化为

$$\begin{aligned}
&\sum_{i=1}^{N}\sum_{z_i=0}^{1} Q_i(z_i)\log\frac{\mathbb{P}(y_i,z_i;\phi)}{Q_i(z_i)}\\
=&\sum_{i=1}^{N}\sum_{z_i=0}^{1} Q_i(z_i)\log\frac{\mathbb{P}(y_i|z_i)\mathbb{P}(z_i;\phi)}{Q_i(z_i)}\\
=&\sum_{i=1}^{N} Q_i(z_i=0)\log\frac{\mathbb{P}(y_i|z_i=0)(1-\phi)}{Q_i(z_i=0)}+Q_i(z_i=1)\log\frac{\mathbb{P}(y_i|z_i=1)\phi}{Q_i(z_i=1)}\\
=&\sum_{i=1}^{N} Q_i(z_i=0)\log\frac{\prod_{j=1}^{n}(\phi_{j|z_i=0})^{\mathbb{1}_{y_i^{(j)}=1}}(1-\phi_{j|z_i=0})^{\mathbb{1}_{y_i^{(j)}=0}}(1-\phi)}{Q_i(z_i=0)}+\\
&Q_i(z_i=1)\log\frac{\prod_{j=1}^{n}(\phi_{j|z_i=1})^{\mathbb{1}_{y_i^{(j)}=1}}(1-\phi_{j|z_i=1})^{\mathbb{1}_{y_i^{(j)}=0}}\phi}{Q_i(z_i=1)}
\end{aligned}\tag{9.3}$$

公式(9.3)对ϕ求偏导，令其等于零，得到

$$\begin{aligned}
\phi &= \frac{\sum_{i=1}^{N} Q_i(z_i=1)}{\sum_{i=1}^{N} Q_i(z_i=1)+Q_i(z_i=0)}\\
&= \frac{\sum_{i=1}^{N} Q_i(z_i=1)}{\sum_{i=1}^{N} 1} \qquad \left(Q_i\text{是概率密度函数，}\sum_{z_j}Q_i(z_j)=1\right)\\
&= \frac{\sum_{i=1}^{N} Q_i(z_i=1)}{N}
\end{aligned}$$

公式(9.3)分别对$\phi_{j|z_i=1},\phi_{j|z_i=0}$求偏导，令其等于零，得到(忽略$z_i$下标$i$)

$$\begin{aligned}
\phi_{j|z=1} &= \frac{\sum_{i=1}^{N} Q_i(z_i=1)\mathbb{1}_{y_i^{(j)}=1}}{\sum_{i=1}^{N} Q_i(z_i=1)}\\
\phi_{j|z=0} &= \frac{\sum_{i=1}^{N} Q_i(z_i=0)\mathbb{1}_{y_i^{(j)}=1}}{\sum_{i=1}^{N} Q_i(z_i=0)}
\end{aligned}$$

由此可以对$\phi,\phi_{j|z=1},\phi_{j|z=0}$进行迭代。

- 重复以上两步，直至收敛。

10 隐马尔科夫模型

10.1 后向算法

Exercise 10.1. 给定盒子和球组成的隐马尔科夫模型$\lambda = (A, B, \pi)$，其中

$$A = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}, B = \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \\ 0.7 & 0.3 \end{bmatrix}, \pi = (0.2, 0.4, 0.4)^T$$

设$T = 4$，$O = \big($红，白，红，白$\big)$，用后向算法计算$\mathbb{P}(O|\lambda)$

Solution 10.1.

1. $T = 4$，$\beta_4(1) = \beta_4(2) = \beta_4(3) = 1$
2. $T = 3$
   - $\beta_3(1) = \sum_{j=1}^{3} a_{1j} b_j(o_4) \beta_4(j) = 0.5 * 0.5 + 0.2 * 0.6 + 0.3 * 0.3 = 0.46$
   - $\beta_3(2) = \sum_{j=1}^{3} a_{2j} b_j(o_4) \beta_4(j) = 0.3 * 0.5 + 0.5 * 0.6 + 0.2 * 0.3 = 0.51$
   - $\beta_3(3) = \sum_{j=1}^{3} a_{2j} b_j(o_4) \beta_4(j) = 0.2 * 0.5 + 0.3 * 0.6 + 0.5 * 0.3 = 0.43$
3. $T = 2$
   - $\beta_2(1) = \sum_{j=1}^{3} a_{1j} b_j(o_3) \beta_3(j) = 0.5 * 0.5 * 0.46 + 0.2 * 0.4 * 0.51 + 0.3 * 0.7 * 0.43 = 0.2461$
   - $\beta_2(2) = \sum_{j=1}^{3} a_{2j} b_j(o_3) \beta_3(j) = 0.3 * 0.5 * 0.46 + 0.5 * 0.4 * 0.51 + 0.2 * 0.7 * 0.43 = 0.2312$
   - $\beta_2(3) = \sum_{j=1}^{3} a_{2j} b_j(o_3) \beta_3(j) = 0.2 * 0.5 * 0.46 + 0.3 * 0.4 * 0.51 + 0.5 * 0.7 * 0.43 = 0.2577$
4. $T = 1$
   - $\beta_2(1) = \sum_{j=1}^{3} a_{1j} b_j(o_2) \beta_2(j) = 0.5 * 0.5 * 0.2461 + 0.2 * 0.6 * 0.2312 + 0.3 * 0.3 * 0.2577 = 0.112462$
   - $\beta_2(2) = \sum_{j=1}^{3} a_{2j} b_j(o_2) \beta_2(j) = 0.3 * 0.5 * 0.2461 + 0.5 * 0.6 * 0.2312 + 0.2 * 0.3 * 0.2577 = 0.121737$
   - $\beta_2(3) = \sum_{j=1}^{3} a_{2j} b_j(o_2) \beta_2(j) = 0.2 * 0.5 * 0.2461 + 0.3 * 0.6 * 0.2312 + 0.5 * 0.3 * 0.2577 = 0.104881$
5. $\mathbb{P}(O|\lambda) = \sum_{i=1}^{3} \pi_i b_i(o_1) \beta_1(i) = 0.2 * 0.5 * 0.112462 + 0.4 * 0.4 * 0.121737 + 0.4 * 0.7 * 0.104881 = 0.0601$

10.2 前向后向概率计算

Exercise 10.2. 给定盒子和球组成的隐马尔科夫模型$\lambda = (A, B, \pi)$，其中

$$A = \begin{bmatrix} 0.5 & 0.1 & 0.4 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}, B = \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \\ 0.7 & 0.3 \end{bmatrix}, \pi = (0.2, 0.3, 0.5)^T$$

设$T = 8$, $O =$ (红，白，红，红，白，红，白，白)，用后向算法计算$\mathbb{P}(i_4 = q_3|O, \lambda)$

Solution 10.2.

先计算前向概率

1. $T = 1$
   - $\alpha_1(1) = \pi_1 b_1(o_1) = 0.2 * 0.5 = 0.1$
   - $\alpha_1(2) = \pi_2 b_2(o_1) = 0.3 * 0.4 = 0.12$
   - $\alpha_1(3) = \pi_3 b_3(o_1) = 0.5 * 0.7 = 0.35$
2. $T = 2$
   - $\alpha_2(1) = \sum_{j=1}^{3} \alpha_1(j) a_{j1}(o_2) b_1(0_2) = (0.1 * 0.5 + 0.12 * 0.3 + 0.35 * 0.2) * 0.5 = 0.078$
   - $\alpha_2(2) = \sum_{j=1}^{3} \alpha_1(j) a_{j2}(o_2) b_2(0_2) = (0.1 * 0.1 + 0.12 * 0.5 + 0.35 * 0.2) * 0.6 = 0.086$
   - $\alpha_2(3) = \sum_{j=1}^{3} \alpha_1(j) a_{j3}(o_2) b_3(0_2) = (0.1 * 0.4 + 0.12 * 0.2 + 0.35 * 0.6) * 0.3 = 0.0822$
3. $T = 3$
   - $\alpha_3(1) = \sum_{j=1}^{3} \alpha_2(j) a_{j1}(o_2) b_1(0_3) = (0.078 * 0.5 + 0.086 * 0.3 + 0.0822 * 0.2) * 0.5 = 0.04062$
   - $\alpha_3(2) = \sum_{j=1}^{3} \alpha_2(j) a_{j2}(o_2) b_2(0_3) = (0.078 * 0.1 + 0.086 * 0.5 + 0.0822 * 0.2) * 0.4 = 0.02689$
   - $\alpha_3(3) = \sum_{j=1}^{3} \alpha_2(j) a_{j3}(o_2) b_3(0_3) = (0.078 * 0.4 + 0.086 * 0.2 + 0.0822 * 0.6) * 0.7 = 0.0684$
4. $T = 4$
   - $\alpha_4(1) = \sum_{j=1}^{3} \alpha_3(j) a_{j1}(o_2) b_1(0_4) = (0.04062 * 0.5 + 0.02689 * 0.3 + 0.0684 * 0.2) * 0.5 = 0.02101$
   - $\alpha_4(2) = \sum_{j=1}^{3} \alpha_3(j) a_{j2}(o_2) b_2(0_4) = (0.04062 * 0.1 + 0.02689 * 0.5 + 0.0684 * 0.2) * 0.4 = 0.01247$
   - $\alpha_4(3) = \sum_{j=1}^{3} \alpha_3(j) a_{j3}(o_2) b_3(0_4) = (0.04062 * 0.4 + 0.02689 * 0.2 + 0.0684 * 0.6) * 0.7 = 0.04386$

再计算后向概率

- $T = 8$，$\beta_8(1) = \beta_8(2) = \beta_8(3) = 1$

- $T=7$
  1. $\beta_7(1)=\sum_{j=1}^{3}a_{1j}b_j(o_8)\beta_8(j)=0.5*0.5+0.1*0.6+0.4*0.3=0.43$
  2. $\beta_7(2)=\sum_{j=1}^{3}a_{2j}b_j(o_8)\beta_8(j)=0.3*0.5+0.5*0.6+0.2*0.3=0.51$
  3. $\beta_7(3)=\sum_{j=1}^{3}a_{3j}b_j(o_8)\beta_8(j)=0.2*0.5+0.2*0.6+0.6*0.3=0.4$
- $T=6$
  1. $\beta_6(1)=\sum_{j=1}^{3}a_{1j}b_j(o_7)\beta_7(j)=0.5*0.5*0.43+0.1*0.6*0.51+0.4*0.3*0.4=0.1861$
  2. $\beta_6(2)=\sum_{j=1}^{3}a_{2j}b_j(o_7)\beta_7(j)=0.3*0.5*0.43+0.5*0.6*0.51+0.2*0.3*0.4=0.2415$
  3. $\beta_6(3)=\sum_{j=1}^{3}a_{3j}b_j(o_7)\beta_7(j)=0.2*0.5*0.43+0.2*0.6*0.51+0.6*0.3*0.4=0.1762$
- $T=5$
  1. $\beta_5(1)=\sum_{j=1}^{3}a_{1j}b_j(o_6)\beta_6(j)=0.5*0.5*0.1861+0.1*0.4*0.2415+0.4*0.7*0.1762=0.1055$
  2. $\beta_5(2)=\sum_{j=1}^{3}a_{2j}b_j(o_6)\beta_6(j)=0.3*0.5*0.1861+0.5*0.4*0.2415+0.2*0.7*0.1762=0.1008$
  3. $\beta_5(3)=\sum_{j=1}^{3}a_{3j}b_j(o_6)\beta_6(j)=0.2*0.5*0.1861+0.2*0.4*0.2415+0.6*0.7*0.1762=0.1119$
- $T=4$
  1. $\beta_4(1)=\sum_{j=1}^{3}a_{1j}b_j(o_5)\beta_5(j)=0.5*0.5*0.1055+0.1*0.6*0.1008+0.4*0.3*0.1119=0.045851$
  2. $\beta_4(2)=\sum_{j=1}^{3}a_{2j}b_j(o_5)\beta_5(j)=0.3*0.5*0.1055+0.5*0.6*0.1008+0.2*0.3*0.1119=0.052778$
  3. $\beta_4(3)=\sum_{j=1}^{3}a_{3j}b_j(o_5)\beta_5(j)=0.2*0.5*0.1055+0.2*0.6*0.1008+0.6*0.3*0.1119=0.042788$
- $\mathbb{P}(i_4=q_3|O,\lambda)=\gamma_4(3)=\frac{\alpha_4(3)\beta_4(3)}{\sum_{i=1}^{3}\alpha_4(i)\beta_4(i)}=0.536478$

10.3 维特比算法

Exercise 10.3. 给定盒子和球组成的隐马尔科夫模型$\lambda=(A,B,\pi)$，其中

$$A=\begin{bmatrix}0.5&0.2&0.3\\0.3&0.5&0.2\\0.2&0.3&0.5\end{bmatrix},B=\begin{bmatrix}0.5&0.5\\0.4&0.6\\0.7&0.3\end{bmatrix},\pi=(0.2,0.4,0.4)^T$$

设$T=4$，$O=$(红，白，红，白)，用维特比算法计算最优路径$I^*=(i_1^*,i_2^*,i_3^*,i_4^*)$

Solution 10.3.

- $T=1$时，$\delta_1(i)=\pi_i b_i(o_1)$
  1. $\delta_1(1)=0.2*0.5=0.1$
  2. $\delta_1(2)=0.4*0.4=0.16$
  3. $\delta_1(3)=0.4*0.7=0.28$
- $T=2$时，$\delta_2(i)=\max\limits_{1\le j\le 3}[\delta_1(j)a_{ji}]b_i(o_2)$
  1. $\delta_2(1)=\max\limits_{1\le j\le 3}\{0.1*0.5,0.16*0.3,0.28*0.2\}*0.5=0.028$，$\psi_2(1)=3$
  2. $\delta_2(2)=\max\limits_{1\le j\le 3}\{0.1*0.2,0.16*0.5,0.28*0.3\}*0.6=0.0504$，$\psi_2(2)=3$
  3. $\delta_2(3)=\max\limits_{1\le j\le 3}\{0.1*0.3,0.16*0.2,0.28*0.5\}*0.3=0.042$，$\psi_2(3)=3$
- $T=3$时，$\delta_3(i)=\max\limits_{1\le j\le 3}[\delta_2(j)a_{ji}]b_i(o_3)$
  1. $\delta_3(1)=\max\limits_{1\le j\le 3}\{0.028*0.5,0.0504*0.3,0.042*0.2\}*0.5=0.00756$，$\psi_3(1)=2$
  2. $\delta_3(2)=\max\limits_{1\le j\le 3}\{0.028*0.2,0.0504*0.5,0.042*0.3\}*0.4=0.01008$，$\psi_3(2)=2$
  3. $\delta_3(3)=\max\limits_{1\le j\le 3}\{0.028*0.3,0.0504*0.2,0.042*0.5\}*0.7=0.0147$，$\psi_3(3)=3$
- $T=4$时，$\delta_4(i)=\max\limits_{1\le j\le 3}[\delta_3(j)a_{ji}]b_i(o_4)$
  1. $\delta_4(1)=\max\limits_{1\le j\le 3}\{0.00756*0.5,0.01008*0.3,0.0147*0.2\}*0.5=0.00189$，$\psi_4(1)=1$
  2. $\delta_4(2)=\max\limits_{1\le j\le 3}\{0.00756*0.2,0.01008*0.5,0.0147*0.3\}*0.6=0.003$，$\psi_4(2)=2$
  3. $\delta_4(3)=\max\limits_{1\le j\le 3}\{0.00756*0.3,0.01008*0.2,0.0147*0.5\}*0.3=0.0022$，$\psi_4(3)=1$
- $P*=\max\limits_{1\le i\le 3}\delta_4(i)=0.003$，$i_4^*=2$，逆向找到之前的最优点
  1. 在$T=3$时，$i_3^*=\psi_4(i_4^*)=2$
  2. 在$T=2$时，$i_2^*=\psi_3(i_3^*)=2$
  3. 在$T=1$时，$i_1^*=\psi_2(i_2^*)=3$

  所以最优序列为$I^*=(i_1^*,i_2^*,i_3^*,i_4^*)=(3,2,2,2)$

10.4 观测序列的概率

Exercise 10.4. 证明$\mathbb{P}(O|\lambda)=\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)$

Solution 10.4.

Proof.

$$\begin{aligned}
\mathbb{P}(O|\lambda) &= \sum_{i=1}^{N}\sum_{j=1}^{N}\mathbb{P}(o_1,o_2,...o_T,i_t=q_i,i_{t+1}=q_j|\lambda) \quad (\text{根据全概率公式}) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N}\mathbb{P}(o_{t+1},o_{t+2},...o_T,i_{t+1}=q_j|o_1,o_2,...o_t,i_t=q_i,\lambda)\mathbb{P}(o_1,o_2,...o_t,i_t=q_i|\lambda) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N}\mathbb{P}(o_{t+1},o_{t+2},...o_T|o_1,o_2,...o_t,i_t=q_i,i_{t+1}=q_j,\lambda)\mathbb{P}(i_{t+1}=q_j|o_1,o_2,...o_t,i_t=q_i,\lambda) \\
&\quad \mathbb{P}(o_1,o_2,...o_t,i_t=q_i|\lambda)
\end{aligned}$$

其中

$$\begin{aligned}
&\mathbb{P}(o_{t+1},o_{t+2},...o_T|o_1,o_2,...o_t,i_t=q_i,i_{t+1}=q_j,\lambda) \\
=\ &\mathbb{P}(o_{t+1},o_{t+2},...o_T|i_{t+1}=q_j,\lambda) \quad (\text{根据观测独立性假设}) \\
=\ &\mathbb{P}(o_{t+2},o_{t+3},...o_T|o_{t+1},i_{t+1}=q_j,\lambda)\mathbb{P}(o_{t+1}|i_{t+1}=q_j,\lambda) \\
=\ &\mathbb{P}(o_{t+2},o_{t+3},...o_T|i_{t+1}=q_j,\lambda)\mathbb{P}(o_{t+1}|i_{t+1}=q_j,\lambda) \\
=\ &\beta_{t+1}(j)b_j(o_{t+1})
\end{aligned}$$

$$\begin{aligned}
&\mathbb{P}(i_{t+1}=q_j|o_1,o_2,...o_t,i_t=q_i,\lambda) \\
=\ &\mathbb{P}(i_{t+1}=q_j|i_t=q_i,\lambda) \quad (\text{根据齐次马尔科夫性质假设}) \\
=\ &a_{ij}
\end{aligned}$$

$$\mathbb{P}(o_1,o_2,...o_t,i_t=q_i|\lambda)=\alpha_t(i)$$

故综合以上结果，可以得到

$$\begin{aligned}
\mathbb{P}(O|\lambda) &= \sum_{i=1}^{N}\sum_{j=1}^{N}\beta_{t+1}(j)b_j(o_{t+1})*a_{ij}*\alpha_t(i) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)
\end{aligned}$$

□

10.5 维特比算法和前向算法的比较

Exercise 10.5. 比较维特比算法中的δ和前向算法中的α的计算的主要区别。

Solution 10.5.

1. 维特比算法中的$\delta_t(i)$依赖于t时刻之前的状态序列，而前向算法中的α仅依赖于t时刻的状态
2. 维特比算法使用动态规划方法求解$\delta_t(i)$，而前向算法中的α仅需要根据递推公式$a_{t+1}(i)=[\sum_{j=1}^{N}\alpha_t(j)a_{ji}]b_i(o_{t+1})$迭代即可

11 条件随机场

11.1 因子分解式

Exercise 11.1. 写出图11.3中无向图描述的概率图模型的因子分解式

Solution 11.1.

$$\mathbb{P}(Y) = \Psi(y_1, y_2, y_3)\Psi(y_2, y_3, y_4)$$

11.2 前向后向算法

Exercise 11.2. 求证$Z(x) = \alpha_n^T(x) \cdot 1 = 1^T \cdot \beta_1(x)$

Solution 11.2.

Proof. 根据$Z(x)$定义，可知

$$\begin{aligned} Z(x) &= (M_1(x)M_2(x)...M_{n+1}(x))_{start,stop} \\ &= \alpha_0^T(x)M_1(x)M_2(x)...M_{n+1}(x) \cdot 1 \\ &= \alpha_1^T(x)M_2(x)...M_{n+1}(x) \cdot 1 \\ &= ... \\ &= \alpha_n^T M_{n+1}(x) \cdot 1 \end{aligned} \tag{11.1}$$

假设$\alpha = (\alpha_1, \alpha_2, ...\alpha_m)^T$，$M_{n+1}(x)$为某列为1其余均为0的$m$阶矩阵，不妨假设第$k$列为1。那么

$$\begin{aligned} \alpha_n^T M_{n+1}(x) \cdot 1 &= \left(0, 0, ..., \sum_{\substack{i=1 \\ k-th}}^{m} \alpha_i, ...0\right)_{1\times m} \cdot 1 \\ &= \sum_{i=1}^{m} \alpha_i \\ &= \alpha_n^T \cdot 1 \end{aligned} \tag{11.2}$$

故根据公式(11.1)和(11.2)可得到

$$\begin{aligned} Z(x) &= \alpha_n^T M_{n+1}(x) \cdot 1 \\ &= \alpha_n^T \cdot 1 \end{aligned}$$

同理可知

$$\begin{aligned} Z(x) &= (M_1(x)M_2(x)...M_{n+1}(x))_{start,stop} \\ &= 1 \cdot M_1(x)M_2(x)...M_{n+1}(x)\beta_{n+1}(x) \\ &= 1 \cdot M_1(x)...\beta_n(x) \\ &= ... \\ &= 1 \cdot \beta_1(x) \end{aligned}$$

综上可得$Z(x) = \alpha_n^T(x) \cdot 1 = 1^T \cdot \beta_1(x)$

□

11.3 条件随机场模型学习的梯度下降法

Exercise 11.3. 写出条件随机场模型学习的梯度下降算法

Solution 11.3.

输入:特征函数$f_1, f_2, ..., f_n$；经验分布$\widetilde{\mathbb{P}}(X, y)$，计算精度$\epsilon$
输出:最优参数值$\hat{w}$以及最优模型$\mathbb{P}_{\hat{w}}(y|x)$

1. 选定初值点w^0，令$k = 0$
2. 计算目标函数的梯度$g_k = g(w^k)$，当$\|g_k\| < \epsilon$，停止迭代，$\hat{w} = w^k$；否则令$p_k = -g_k$，求λ_k使得$f(w^k + \lambda_k p_k) = \min\limits_{\lambda \geq 0} f(w^k + \lambda p_k)$
3. 令$w^{k+1} = w^k + \lambda_k p_k$，计算$f(w^{k+1})$，当$\|w^{k+1} - w^k\| < \epsilon$或者$\|f(w^{k+1}) - f(w^k)\| < \epsilon$ 时，停止迭代，令$\hat{w} = w^{k+1}$
4. 转(2)

11.4 状态序列的概率

Exercise 11.4. 参考图*11.6*的状态路径图，假设随机矩阵分别是

$$M_1(x) = \begin{bmatrix} 0 & 0 \\ 0.5 & 0.5 \end{bmatrix}, M_2(x) = \begin{bmatrix} 0.3 & 0.7 \\ 0.7 & 0.3 \end{bmatrix}, M_3(x) = \begin{bmatrix} 0.5 & 0.5 \\ 0.6 & 0.4 \end{bmatrix}, M_4(x) = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

求以*start*= 2为起点*stop*= 2为终点的所有路径的状态序列y的概率以及概率最大的状态序列

Solution 11.4.

根据图11.6以start= 2为起点stop= 2为终点的路径为

$$(1,1,1),(1,1,2),(1,2,1),(1,2,2)$$
$$(2,2,2),(2,1,2),(2,2,1),(2,2,2)$$

对应的非规范化概率为

$$0.5*0.3*0.5, 0.5*0.3*0.5, 0.5*0.7*0.6, 0.5*0.7*0.4$$
$$0.5*0.3*0.4, 0.5*0.7*0.5, 0.5*0.3*0.6, 0.5*0.3*0.4$$

概率最大的状态序列为$(1,2,1)$

12 附录

12.1 习题(9.1)代码

```
import numpy as np

def mu_j(pi, p, q, y_j):
    nominator = pi * p ** y_j * (1 - p) ** (1 - y_j)
    denominator = pi * p ** y_j * (1 - p) ** (1 - y_j)
    + (1-pi) * q ** y_j * (1 - q) ** (1 - y_j)
    return nominator / denominator

def e_step(pi, p, q, y):
    ret = [mu_j(pi, p, q, y_j) for y_j in y]
    return np.array(ret)

def m_step(mu, y):
    pi = np.mean(mu)
    p = np.dot(mu, y) / (pi * len(y))
    q = (np.sum(y) - np.dot(mu, y)) / (len(y) - np.sum(mu))
    return pi, p, q

def update_params(params, y):
    pi, p, q = params
    mu = e_step(pi, p, q, y)
    pi, p, q = m_step(mu, y)
    return np.array([pi, p, q])

def em(init_params, y, tol=10 ** -10):
    prev_params = init_params
    current_params = update_params(prev_params, y)
    while np.max(np.abs(current_params - prev_params)) > tol:
        prev_params = current_params
        current_params = update_params(prev_params, y)
    return current_params

if __name__ == "__main__":
    y_test = np.array([1, 1, 0, 1, 0, 0, 1, 0, 1, 1])
    init_params_test = np.array([0.46, 0.55, 0.67])
    print em(init_params_test, y_test)
```