

数据科学

Homework 3

第1题

(1) 在题目中, 唯一商品为 $\{milk, beer, diaper, bread, butter, cookie\}$, -共 $n=6$,
则总的可能项集数量为:

$$2^6 - 1 = 63$$

从每个项集可以产生的规则总数:

$$\sum_{k=1}^6 \binom{6}{k} (2^k - k - 1) = 602$$

(2). 若规定支持度 > 0 , 则能提取的频繁项集长度取决于事务最长长度,
则选 $\{\text{牛奶}, \text{尿布}, \text{面包}, \text{黄油}\}$,
频繁项集最大长度为 4.

(3). 可以提取的 3 项集最大数量为:

$\left[\begin{matrix} n \\ 3 \end{matrix} \right]$, 令 $n=6$ 为商品总数,

$$\text{则 } \left[\begin{matrix} 6 \\ 3 \end{matrix} \right] = \frac{6!}{3! \times 3!} = 20,$$

(4). 具有最大支持度的项集(长度 > 2): $\{\text{牛奶}, \text{尿布}\}, \{\text{尿布}, \text{啤酒}\}, \{\text{面包}, \text{黄油}\}$
并且其支持度均为 4

第2题:

1) APRIORI 算法: ①首先, 统计出频繁项集:

M	3	D	1	E	4
O	4	A	1		
N	2	U	1		
K	5	C	2		
Y	3	I	1		

由 $\min_{\text{sup}} = 60\%$ 得出, 项集出现次数不少于 3.

二、留下

M	3
O	4
K	5
Y	3
E	4

②: 其次统计频繁 2 项集:

{M, O}	1	{O, E}	3
{M, K}	3	{O, Y}	2
{M, E}	2	{K, E}	4
{M, Y}	2	{K, Y}	3
{O, K}	3	{E, Y}	2

因此频繁项集为 $\{ \{M, K\}, \{O, K\}, \{O, E\}, \{K, E\}, \{K, Y\} \}$

③: 从频繁 2 项集中生成候选 3 项集:

1	{M, K, O}	{M, O, Y}	{O, E, Y}
2	{M, K, E}	{M, E, Y}	
2	{M, K, Y}	{K, O, E}	
1	{M, O, E}	{K, O, Y}	

其中只有 $\text{Support}\{K, O, E\} \geq 3$

1. 频繁 3 项集为 {K, O, E}

FP-Growth:

① 扫描事务数据库，统计每个商品的支持度：

如 Apr(上述)所示结果

保留上述频繁 1-项集，并且按支持度从高到低排序为：

$\{K, O, E, M, Y\}$

② 对事务顺序进行重排：

T100: $\{K, O, E, M, Y\}$

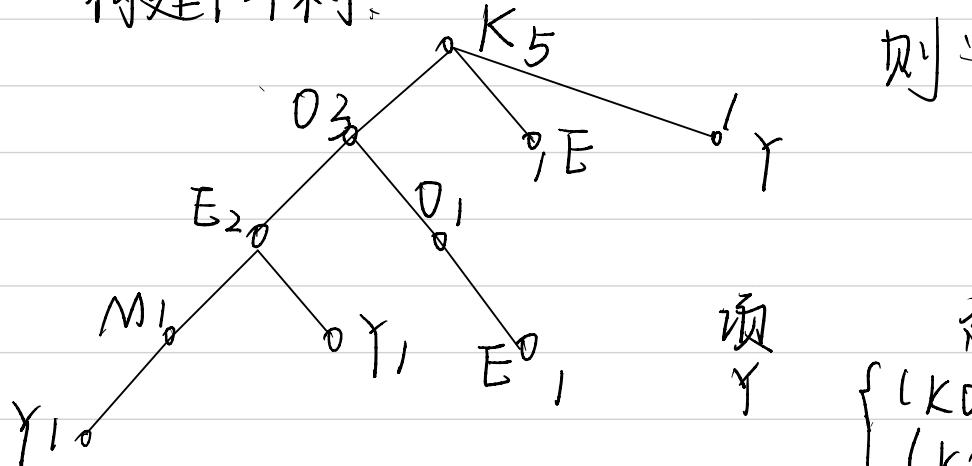
T200: $\{K, O, E, Y\}$

T300: $\{K, E\}$

T400: $\{K, Y\}$

T500: $\{K, O, O, E\}$

③ 构建 FP 树：



则：

项

条件模式基
 $\{(KOEM: 1),\}$
 $\{(KOE: 1)\}$
 $\{(K: 1),\}$

E

$\{(KO: 2),\}$
 $\{(KO: 1)\},$
 $\{(K: 1)\}$

条件 FP 树

$\langle K: 3 \rangle$

产生的频繁模式

$\{K, Y: 3\}$

$\{KE: 4\}$

$O \quad \{ (K: 3) \}$
 $\{ (K, O) : 1 \}$
 $\langle K: 1 \rangle$
 $\{ K_0 : 4 \}$

二者效率比较：Apriori 算法的计算过程必须对数据库作多次扫描，而 FP-增长算法在构造过程中只用扫描一次数据库，再加上初始时为确定支持度递减排序的一次扫描，共计只需要 2 次扫描；由于在 Apriori 算法中自身连接过程产生候选集，其计算代价非常高，而 FP 增长算法不需产生任何候选集。

(3): 由频繁子项集产生的规则有：

$(K, O) \rightarrow E$	Confidence = $\frac{4}{4} = 1$	Support = $\frac{4}{5} = 0.8$
$(K, E) \rightarrow O$	Confidence = $\frac{4}{4} = 1$	Support = $\frac{4}{5} = 0.8$
$(E, O) \rightarrow K$,	Confidence = $\frac{4}{4} = 1$	Support = $\frac{4}{5} = 0.8$

第3题：

(1) ① 特征随机性：在 Bagging 中，每棵树都使用整个特征集训练，在随机森林中，每棵树在节点分裂时随机选择部分特征（特征子集），减少了每次节点分裂计算复杂度。

② Bootstrap采样：通过自举采样法构建多个模型，并进行集成，可以有效减小模型的方差。
适用于高方差，低偏差的模型。

特征随机化：在每个分裂节点，使用随机的特征子集来构建不同的树，进一步增强模型的多样性。

适用于特征数目较多时，用随机特征来构建多个基础模型，减小相关性。

Boosting：
通过给不同样本不同权重的方式，使模型更关注之前分类错误的样本。
适用于如决策树等低偏差，高方差的模型

(3) 答：不能，或提升效果有限

① naive Bayesian 本身是“高偏差，低方差”的模型；而 Bagging 适用于低偏差，高方差，且 naive Bayesian 假设特征之间条件独立且简单；因此 naive Bayesian 的性能往往受限于数据特征的质量，以及假设的准确性。

(4) ① 相同：

① 两者都是 Boosting 方法，重点关注错误样本，结合多个弱分类器提升性能。

② 都会在迭代中对样本进行加权。

③ 不同：① 更新方式：Gradient Boosting 每一次建立模型都是在之前建立模型损失函数的梯度下降方向；而 AdaBoosting 通过调样本权重来聚焦于困难样本，而更新规则简单。

④ 对噪声的 Robustness：AdaBoosting 对噪声敏感，易受异常值影响。

Gradient Boosting 在噪声多时可以调整损失函数来控制 overfitting

第4题

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

(1) 在 x_1, x_3 未知时: $P(x_1, x_3, x_5, x_6) = P(x_1) \cdot P(x_3) \cdot P(x_5|x_1, x_3) \cdot P(x_6|x_4)$

$$\begin{aligned} x_1, x_3 \text{ 已知时: } P(x_5, x_6|x_1, x_3) &= \frac{P(x_5, x_6, x_1, x_3)}{P(x_1, x_3)} \\ &= \frac{P(x_1) P(x_3) P(x_5|x_1, x_3) P(x_6|x_4)}{P(x_1, x_3)} \end{aligned}$$

由于 x_1 与 x_3 为 head-to-head, 因此 x_1 与 x_5 独立

$$\begin{aligned} \therefore P(x_5, x_6|x_1, x_3) &= \frac{P(x_1) P(x_3) P(x_5|x_1, x_3) \cdot P(x_6|x_4)}{P(x_1) P(x_3)} \\ &= P(x_5|x_1, x_3) P(x_6|x_4) \end{aligned}$$

\therefore 在给定 x_1, x_3 的情况下, x_5, x_6 不是条件独立

(2) 在 x_2, x_3 已知时, $P(x_2, x_3, x_5, x_6) = P(x_2) P(x_3) P(x_5|x_3) P(x_6|x_4)$

$$\begin{aligned} x_2, x_3 \text{ 未知时, } P(x_5, x_6|x_2, x_3) &= \frac{P(x_2, x_3, x_5, x_6)}{P(x_2, x_3)} \\ &= P(x_5|x_3) \cdot P(x_6|x_4) \end{aligned}$$

由于 x_4 未知, $\therefore x_5, x_6$ 不是条件独立

(3) $P(x_1, x_2, \dots, x_7) = P(x_1) P(x_2) P(x_3) P(x_4|x_1, x_2, x_3) P(x_3|x_1, x_2) P(x_6|x_4) P(x_7|x_4, x_5)$

联合分布如上。

第5題

$$P(X^{(1)}=1 | Y=1) = \frac{1/10}{1/2} = \frac{1}{5}$$

$$P(X^{(1)}=1 | Y=-1) = \frac{3/10}{1/2} = \frac{3}{5} \times 2 = \frac{3}{5}$$

$$P(X^{(1)}=2 | Y=1) = \frac{1/5}{1/2} = \frac{2}{5}$$

$$P(X^{(1)}=2 | Y=-1) = \frac{2/10}{1/2} = \frac{2}{5}$$

$$P(X^{(1)}=3 | Y=1) = \frac{2/10}{1/2} = \frac{2}{5}$$

$$P(X^{(1)}=3 | Y=-1) = \frac{0}{1/2} = 0$$

$$P(X^{(2)}=S | Y=1) = \frac{1/2}{2/10} = \frac{2}{5}$$

$$P(X^{(2)}=S | Y=-1) = \frac{2}{2} = \frac{2}{5}$$

$$P(X^{(3)}=T | Y=1) = \frac{5}{5} = \frac{2}{5}$$

$$P(X^{(3)}=T | Y=-1) = \frac{3}{5} = \frac{3}{5}$$

$$\therefore P(Y=1) \times P(X^{(3)}=T | Y=1) \times P(X^{(2)}=S | Y=1) \times P(X^{(1)}=2 | Y=1)$$

$$= \frac{1}{2} \times \frac{2}{5} \times \frac{2}{5} \times \frac{2}{5}$$

$$= \frac{4}{125}$$

$$P(Y=-1) \times P(X^{(3)}=T | Y=-1) \times P(X^{(2)}=S | Y=-1) \times P(X^{(1)}=2 | Y=-1)$$

$$= \frac{1}{2} \times \frac{3}{5} \times \frac{2}{5} \times \frac{2}{5}$$

$$= \frac{6}{125}$$

\therefore 分类数 $Y=-1$

第6题

① 首先认为每个数据都是同等重要的，则($i=1, \dots, 10$)的权重一样

x	0	1	2	3	4	5	6	7	8	9
w	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
y	1	1	1	-1	-1	-1	1	1	1	-1

此时在训练数据上，阈值为2.5时，6、7、8号数据样本被错分为反例。

$$e_{\text{weight}1} = 0.1 + 0.1 + 0.1 = 0.3$$

$$\text{计算得 } \alpha_1 = \frac{1}{2} \log \frac{1-e_1}{e_1} \approx 0.4236$$

② 进行第2轮迭代：

x	0	1	2	3	4	5	6	7	8	9
w	0.07143	0.07143	0.07143	0.07143	0.07143	0.07143	0.16667	0.16667	0.16667	0.07143
y	1	1	1	-1	-1	-1	1	1	1	-1

此时取阈值为8.5时分类误差最低。

$$e_{\text{weight}2} = e_2 = 0.07143 \times 3 = 0.2143$$

$$\alpha_2 = \frac{1}{2} \log \frac{1-e_2}{e_2} \approx 0.6496$$

③ 进行第3轮迭代：

x	0	1	2	3	4	5	6	7	8	9
w	0.0455	0.0455	0.0455	0.16667	0.16667	0.16667	0.1060	0.1060	0.1060	0.0455
y	1	1	1	-1	-1	-1	1	1	1	-1

此时阈值取5.5时最佳。

$$e_{\text{weight}3} = e_3 = 0.0455 \times 4 = 0.1820$$

$$\alpha_3 = \frac{1}{2} \log \frac{1-e_3}{e_3} \approx 0.7514$$

④ 更新训练数据后权值分布，开始第4轮迭代：

x	0	1	2	3	4	5	6	7	8	9
w	0.125	0.125	0.125	0.102	0.102	0.102	0.065	0.065	0.065	0.125
y	1	1	1	-1	-1	-1	1	1	1	-1

最终分类器为：

$$G(x) = \begin{cases} 1, & 0 \leq x \leq 2.5 \vee 5.5 \leq x \leq 8.5 \\ -1, & 2.5 \leq x \leq 5.5 \vee x \geq 8.5 \end{cases}$$

第七题

证明：①当 basis function 为基本分类器时，该加法模型等价于 AdaBoosting 最终分类器。

为 $f(x) = \sum_{m=1}^M \alpha_m G_m(x)$ ，由基本分类器 $G_m(x)$ 及系数 α_m 组成， $m=1, 2, \dots, M$ 。前向分步算法与 AdaBoosting 逐一学习基本分类器的过程一致。

②现证明前向分步算法的损失函数是指数损失函数 $L(y, f(x)) = e^{-yf(x)}$ ，其学习的具体操作等价于 AdaBoosting。

设经过了 $(m-1)$ 轮迭代前向分步算法，得到 $f_{m-1}(x)$ ：

$$\begin{aligned} f_{m-1}(x) &= f_{m-2}(x) + \alpha_{m-1} G_{m-1}(x) \\ &= \sum_{i=1}^{m-1} \alpha_i G_i(x), \end{aligned}$$

在 m 轮得到了 $\alpha_m, G_m(x)$ 和 $f_m(x)$ ，

$$f_m(x) = f_{m-1}(x) + \alpha_m G_m(x).$$

目标为最小化 $f_m(x)$ 在训练数据集 T 的指数损失，即

$$\begin{aligned} (\alpha_m, G_m(x)) &= \underset{\alpha, G}{\operatorname{argmin}} \sum_{i=1}^N [\exp(-y_i(f_{m-1}(x_i) + \alpha G(x_i)))] \\ &= \underset{\alpha, G}{\operatorname{argmin}} \sum_{i=1}^N [\tilde{w}_{mi} \exp(-y_i \alpha G(x_i))], \end{aligned}$$

$\tilde{w}_{mi} = e^{-y_i f_{m-1}(x_i)}$ ，依赖于 f_{m-1} ，随迭代而发生改变。

现证明使上式达到最小的 $\alpha_m, G_m(x)$ 也能令 AdaBoosting 最小。

首先求 $G_m^*(x) = \underset{G}{\operatorname{argmin}} \sum_{i=1}^N \tilde{w}_{mi} I(y_i \neq G(x_i))$ ，($\tilde{w}_{mi} = e^{-y_i f_{m-1}(x)}$) 而

$$\begin{aligned} \sum_{i=1}^N \tilde{w}_{mi} \exp(-y_i \alpha G(x_i)) &= \sum_{y_i=G_m(x_i)} \tilde{w}_{mi} e^{-\alpha^2} + \sum_{y_i \neq G_m(x_i)} \tilde{w}_{mi} e^{\alpha^2} \\ &= (e^{\alpha^2} - e^{-\alpha^2}) \sum_{y_i \neq G_m(x_i)} \tilde{w}_{mi} I(y_i \neq G(x_i)) + e^{-\alpha^2} \sum_{i=1}^N \tilde{w}_{mi}, \end{aligned}$$

将求得的 $G_m^*(x)$ 代入上式，并对 α 求偏导：

$$\alpha_m^* = \frac{1}{2} \log \frac{1 - e_m}{e_m}, \quad e_m = \frac{\sum_i \tilde{w}_{mi} I(y_i \neq G_m(x_i))}{\sum_i \tilde{w}_{mi}} = \frac{\sum_i w_{mi} I(y_i \neq G_m(x_i))}{\sum_i w_{mi}}$$

即和 AdaBoosting 更新 α_m 一致。

此时更新样本权值： $f_m(x) = f_{m-1}(x) + \alpha_m G_m(x)$ ，以及 $\tilde{w}_{mi} = \exp[-y_i f_{m-1}(x_i)]$ ，得：

$$\tilde{w}_{m+1,i} = \tilde{w}_{m,i} e^{-y_i \alpha_m G_m(x)}$$

与 AdaBoosting 样本权值更新一致。

第八题

对于 x_1 与 x_2 ， $L_p(x_1, x_2) = 4$ ，

对于 x_1 与 x_3 ， $L_1(x_1, x_3) = |3| + |3| = 6$ 。

$$L_2(x_1, x_3) = \sqrt{9 + 9} \approx 4.24$$

$$L_3(x_1, x_3) \approx 3.78$$

$$L_4(x_1, x_3) \approx 3.57$$

则 $p=1$ 或 2 时， x_2 是 x_1 的最近邻点；

$p \geq 3$ 时， x_3 是 x_1 最近邻。

第九題

解: $\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial b} \frac{\partial b}{\partial z} \frac{\partial z}{\partial w_i}$ (i=1,2)

而依鏈式法則可知: $\frac{\partial L}{\partial y} = \frac{-y}{y} - (1-y) \times \frac{-1}{1-y} = \frac{-y}{y} + \frac{1-y}{1-y}$

$$\frac{\partial y}{\partial b} = 1$$

$$\frac{\partial b}{\partial z} = b'(z) = \frac{-e^{-z}}{(1+e^{-z})^2} = b(z)[1-b(z)]$$

$$\frac{\partial z}{\partial w_i} = x_i$$

代入得: $\frac{\partial L}{\partial w_1} = \left[\frac{-y}{y} + \frac{1-y}{1-y} \right] \times b(z) \times [1-b(z)] \cdot x_1$

$$\frac{\partial L}{\partial w_2} = \left[\frac{-y}{y} + \frac{1-y}{1-y} \right] \times b(z) \times [1-b(z)] \cdot x_1$$

第十題:

1) 令 $kernel = k = \begin{bmatrix} -1/2 & 1 & -1/2 \\ -1/2 & 1 & -1/2 \\ -1/2 & 1 & -1/2 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \cdot k = 3 - 1 = 2$

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & -2 & 0 \end{bmatrix} \cdot k = -1 + 2 - 2 = -1, \quad \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \cdot k = -1 - 1 + 1 = -2$$

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ -2 & 0 & -2 \end{bmatrix} \cdot k = -1 - 1 + 1 + 1 + 1 = 1 \quad \therefore \text{得到输出矩阵} \begin{bmatrix} 2 & -2 \\ -1 & 1 \end{bmatrix}$$

2) 全 kernel $= k = \begin{bmatrix} -1/2 & -1/2 & -1/2 \\ 1 & 1 & 1 \\ -1/2 & -1/2 & -1/2 \end{bmatrix}$, $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \cdot k = -1/2 + 1 - 1 - 1/2 = -1$

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & -2 & 0 \end{bmatrix} \cdot k = 3 - 1/2 + 1 = 7/2 \quad \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \cdot k = -1 + 2 - \frac{3}{2} = -\frac{1}{2}$$

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ -2 & 0 & -2 \end{bmatrix} \cdot k = -1 + 3 + 1 \times 2 = 4 \quad \therefore \text{得到输出矩阵为} \begin{bmatrix} -1 & -\frac{1}{2} \\ \frac{7}{2} & 4 \end{bmatrix}$$