COMP9318 Assignment1

Fiona Lin z5131048

April 16, 2019

1. [40 marks]

Solution

1) (1) The tabular form with 4 attributes:

	Location	Time	Item	SUM(Quantity)
(location, time, item)	Sydney	2005	PS2	1400
	Sydney	2006	PS2	1500
	Sydney	2006	Wii	500
	Melbourne	2005	Xbox 360	1700
(location,item)	Sydney	All	PS2	2900
	Sydney	All	Wii	500
	Melbourne	All	Xbox 360	1700
(location,time)	Sydney	2005	All	1400
	Sydney	2006	All	2000
	Melbourne	2005	All	1700
(time, item)	All	2005	PS2	1400
	All	2006	PS2	1500
	All	2006	Wii	500
	All	2005	Xbox 360	1700
(location)	Melbourne	All	All	3400
	Sydney	All	All	1700
(time)	All	2005	All	3100
	All	2006	All	2000
(item)	All	All	PS2	2900
	All	All	Wii	500
	All	All	Xbox 360	1700
()	All	All	All	5100

2) equivalent sql:

```
SELECT Location, Time, Item, SUM(Quantity)
FROM sales
GROUP BY
        GROUPING SETS (
            (Location, Time, Item),
            (Location, Time),
            (Location, Item),
            (Time, Item),
            (Location),
            (Time),
            (Item),
            ()
        )
ORDER BY
Location, Time, Item;
```

3) The result of the query in tabular form:

Location	Time	Item	SUM(Quantity)
Sydney	2006	All	2000
Sydney	All	All	3400
All	All	All	5100
Sydney	All	PS2	2900
All	All	PS2	2900
All	2005	All	3100
All	2006	All	2000

4) Since the function need to map a multi-dimensional point to a one-dimensioinal point, the function need to be bijective to allow the reverse mapping.
$f(Location, Time, Item) = Location * 100 + Time * 10 + Item$

Location	Time	Item	SUM(Quantity)	$f(Location, Time, Item)$
1	1	1	1400	111
1	2	1	1500	121
1	2	3	500	123
2	1	2	1700	212
1	0	1	2900	101
1	0	3	500	103
2	0	2	1700	202
1	1	0	1400	110
1	2	0	2000	120
2	1	0	1700	210
0	1	1	1400	11
0	2	1	1500	21
0	2	3	500	23
0	1	2	1700	12
2	0	0	3400	200
1	0	0	1700	100
0	1	0	3100	10
0	2	0	2000	20
0	0	1	2900	1
0	0	3	500	3
0	0	2	1700	2
0	0	0	5100	0

The MOLAP cube (i.e., sparse multi-dimensional array) in a tabular form of (ArrayIndex, Value).

index = $f(Location, Time, Item)$	SUM(Quantity)
111	1400
121	1500
123	1500
212	1700
101	2900
103	1500
202	1700
110	1400
120	2000
210	1700
11	1400
21	1500
23	500
12	1700
200	3400
100	1700
10	3100
20	2000
1	2900
3	500
2	1700
0	5100

2. [30 marks]

Solution

- Proof: given d-dimensioin column vector $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_d \end{bmatrix}$ as the input feature vector, and each dimensioin of $\mathbf{x}$ takes only 2 values 0 or 1.
The Naïve Bayes classifier will product the label 0 iff $P(y = 0|\mathbf{x}) \geq P(y = 1|\mathbf{x})$, which equivalent:

$$\frac{P(\mathbf{x}|y = 0)P(y = 0)}{P(\mathbf{x}|y = 1)P(y = 1)} \geq 1 \tag{1}$$

Since $P(\mathbf{x}|y) = \prod_{j=0}^{d} P(x_j|y)$, then subsitute into (1), we have

$$\frac{P(y = 0)}{P(y = 1)} \cdot \prod_{j=0}^{d} \frac{P(x_j|y = 0)}{P(x_j|y = 1)} \geq 1 \tag{2}$$

In order to transform the inequality (1) into the linear regression representation. Let $P(y = 0) = p$ and $P(x_j|y = 0) = k_j$ also $P(x_j|y = 1) = m_k$, because f the features are only 2 values 0 or 1, then

$$P(x_j|y = 0) = k_j^{x_j}(1 - k_j)^{(1-x_j)} \qquad \text{and} \qquad P(x_j|y = 1) = m_j^{x_j}(1 - m_j)^{(1-x_j)}$$

Hence (2) can rearrage as below:

$$\frac{p}{1-p} \cdot \prod_{j=0}^{d} \frac{k_j^{x_j}(1 - k_j)^{(1-x_j)}}{m_j^{x_j}(1 - m_j)^{(1-x_j)}} \geq 1$$

$$\frac{p}{1-p} \cdot \prod_{j=0}^{d} \frac{k_j^{x_j}(1 - k_j)(1 - k_j)^{(-x_j)}}{m_j^{x_j}(1 - m_j)(1 - m_j)^{(-x_j)}} \geq 1$$

$$\frac{p}{1-p} \cdot \prod_{j=0}^{d} \frac{k_j^{x_j}(1 - k_j)(1 - m_j)^{(x_j)}}{m_j^{x_j}(1 - m_j)(1 - k_j)^{(x_j)}} \geq 1$$

$$\left(\frac{p}{1-p} \prod_{j=0}^{d} \frac{1 - k_j}{1 - m_j}\right) \cdot \prod_{j=0}^{d} \left(\frac{k_j(1 - m_j)}{m_j(1 - k_j)}\right)^{x_j} \geq 1$$

Taking logarithm on both sides, this become

$$\log\left(\frac{p}{1-p}\prod_{j=0}^{d}\frac{1-k_j}{1-m_j}\right)\cdot\prod_{j=0}^{d}\left(\frac{k_j(1-m_j)}{m_j(1-k_j)}\right)^{x_j}\geq 0$$

$$\log\left(\frac{p}{1-p}\prod_{j=0}^{d}\frac{1-k_j}{1-m_j}\right)+\sum_{j=0}^{d}\log\left(\frac{k_j(1-m_j)}{m_j(1-k_j)}\right)^{x_j}\geq 0$$

$$\log\left(\frac{p}{1-p}\prod_{j=0}^{d}\frac{1-k_j}{1-m_j}\right)+\sum_{j=0}^{d}x_j\log\left(\frac{k_j(1-m_j)}{m_j(1-k_j)}\right)\geq 0$$

$$\log\left(\frac{p}{1-p}\prod_{j=0}^{d}\frac{1-k_j}{1-m_j}\right)+x_0\log\left(\frac{k_0(1-m_0)}{m_0(1-k_0)}\right)+$$

$$x_1\log\left(\frac{k_1(1-m_1)}{m_1(1-k_1)}\right)+...+x_d\log\left(\frac{k_d(1-m_d)}{m_j(1-k_d)}\right)\geq 0$$

As the first term does not have any x_j, so it is a constant for any input $\mathbf{x}$.

Let's denote $b=\log\left(\frac{p}{1-p}\prod_{j=0}^{d}\frac{1-k_j}{1-m_j}\right)$ and $w_j=\log\left(\frac{k_j(1-m_j)}{m_j(1-k_j)}\right)$, the inequality (1) transforms into the below linear regression representation

$$\log\left(\frac{p}{1-p}\prod_{j=0}^{d}\frac{1-k_j}{1-m_j}\right)+x_0\log\left(\frac{k_0(1-m_0)}{m_0(1-k_0)}\right)+$$

$$x_1\log\left(\frac{k_1(1-m_1)}{m_1(1-k_1)}\right)+...+x_d\log\left(\frac{k_d(1-m_d)}{m_j(1-k_d)}\right)\geq 0$$

$$\Rightarrow$$

$$b+x_0w_0+x_1w_1+...+x_dw_d\geq 0$$

$$b+\sum_{j=0}^{d}x_jw_j\geq 0$$

Therefore, if the feature vectors are d-dimension, then a Naïve Bayes classifier is a linear classifier in a $d+1$-dimension space.

- Proof: Given the parameters w_{LR} and w_{NB}, we know from previous

$$w_{NB}=\log\left(\frac{k_j(1-m_j)}{m_j(1-k_j)}\right)$$

While we also know that w_{LR} is obtained by finding the minimum of the *sqared error function* J with the learning rate λ, so its formular of updating rule

$$w_{LR}=w_{LR}-\lambda\frac{\partial J(w)}{\partial w_j}$$

Obviously, it can easily compute w_{NB} in only 1 operation, while w_{LR} needs to compute all the possible w_{LR} and find the minimum one.

Hence, learning w_{NB} is much easier than learning w_{LR}.

3. [30 marks]

Solution

(1) Let y_m denote as our model for the sample of a mixture of S_1 and S_2 , and $q_1,\ q_2$ be the parameters θ. After measuring those 3 components we have obtained the percentages as $\{u_j\}_{j=1}^m$, and let y_o as the samples our observed; Then we have,

$$y_m = \sum_{j=1}^{m}(q_1 * p_{1,j} + q_2 * p_{2,j}) = \sum_{j=1}^{m}(q_1 * p_{1,j} + (1-q_1) * p_{2,j})$$
$$y_o = \sum_{j=1}^{m} u_j$$

Then let $Pr(y_o|\theta)$ be the probability of some y_o being measured from sample with the model y_m and the given θ, assuming all n measured instances are $y_1, y_2, ..., y_n$ independently.
Any θ is possible, but the best θ is more likely. In order to find the best θ, we need to maximise the likelihood $Pr(y_o|\theta)$ which means to be observed more measurement y_o close to y_m for some θ.
Hence,

$$L(\theta) = \prod_{o=1}^{n} P(y_1, y_2, ..., y_n|\theta)$$

Without any other knowledge or information, we assume $P(y_o|\theta)$ follows a normal distribution $\mathcal{N}(\mu,\, \sigma^2)$
The log-likelihood function:

$$l(y_o|\theta) = \sum_{o=1}^{n} \log P(y_o|\theta) = \sum_{o=1}^{n} \log\left(\frac{1}{\sigma\sqrt{2\pi}}\mathbf{e}^{-\frac{(y_o-y_m)^2}{2\sigma^2}}\right)$$
$$= \sum_{o=1}^{n} \log\left(\frac{1}{\sigma\sqrt{2\pi}}\mathbf{e}^{-\frac{(\sum_{j=1}^{m}(q_1*p_{1,j}+(1-q_1)*p_{2,j})-u_j)^2}{2\sigma^2}}\right)$$

(2)