Assignment 1

z5119536 Jinlong Zhang

Q1

1. List the tuples in the complete data cube of R in a tabular form with 4 attributes

Location	Time	Item	Quantity
Sydney	2005	PS2	1400
Sydney	2006	PS2	1500
Sydney	2006	Wii	500
Melbourne	2005	XBox360	1700
Sydney	2005	All	1400
Sydney	2006	All	2000
Melbourne	2005	All	1700
Sydney	All	PS2	2900
Sydney	All	Wii	500
Melbourne	All	XBox360	1700
All	2005	PS2	1400
All	2005	XBox360	1700

All	2006	PS2	1500
All	2006	Wii	500
All	All	PS2	2900
All	All	Wii	500
All	All	XBox360	1700
All	2005	All	3100
All	2006	All	2000
Sydney	All	All	3400
Melbourne	All	All	1700
All	All	All	5100

2. Write down an equivalent SQL statement that computes the same result (i.e., the cube). You can only use standard SQL constructs, i.e., no CUBE BY clause.

(Select Location,time,Item,sum(Quantity) from sales group by Location,time,Item)
union all
(Select Location,time,"All" ,sum(Quantity)from sales group by time,Location)
union all
(Select Location,"All",Item ,sum(Quantity)from sales group by Item,Location)
union all

(Select "All",time,Item ,sum(Quantity)from sales group by time,Item)
union all
(Select "All","All",Item ,sum(Quantity)from sales group by Item)
Union all
(Select "All",time,"All" ,sum(Quantity)from sales group by time)
union all
(Select Location,"All","All" ,sum(Quantity)from sales group by Location)
union all
(Select "All","ALL","All" ,sum(Quantity)from sales)

.

. **3.**

. **Consider the following ice-berg cube query:**

. **SELECT Location, Time, Item, SUM(Quantity)**

FROM Sales

CUBE BY Location, Time, Item

HAVING COUNT(*) > 1

Draw the result of the query in a tabular form.

Location	Time	Item	Quantity
Sydney	2006	All	2000

Sydney	All	PS2	2900
All	All	PS2	2900
All	2005	All	3100
All	2006	All	2000
Sydney	All	All	3400
All	All	All	5100

4.

The mapping function should satisfy the one-to-one function, The maping function is Offset = 12*Location+4*Time+Item, Hence,

Location	Time	Item	Quantity	Offset
1	1	1	1400	17
1	2	1	1500	21
1	2	3	500	23
2	1	2	1700	30
1	1	0	1400	16
1	2	0	2000	20
2	1	0	1700	28

1	0	1	2900	13
1	0	3	500	15
2	0	2	1700	26
0	1	1	1400	5
0	1	2	1700	6
0	2	1	1500	9
0	2	3	500	11
0	0	1	2900	1
0	0	3	500	3
0	0	2	1700	2
0	1	0	3100	4
0	2	0	2000	8
1	0	0	3400	12
2	0	0	1700	24
0	0	0	5100	0

Therefore, the The MOLAP cube (sparse multi-dimensional array) is :

Offset	Quantity
17	1400
21	1500

23	500
30	1700
16	1400
20	2000
28	1700
13	2900
15	500
26	1700
5	1400
6	1700
9	1500
11	500
1	2900
3	500
2	1700
4	3100
8	2000
12	3400
24	1700
0	5100

Q2

. 1. Prove that if the feature vectors are d-dimension, then a Na¨ive Bayes classifier is a linear classifier in a d + 1-dimension space. You need to explicitly write out the vector w that the Na¨ive Bayes classifier learns.

We have d dimensional vector $x = (x_1, x_2 \cdots x_d)^T$

The features x_j are all binary (i.e. 0 or 1).

The class label $y \in [0, 1]$,

The Naive Bayes will predict $y=1$ if $P(y=1|x) \geq P(y=0|x)$

or equivalently: $\dfrac{P(x|y=1)P(y=1)}{P(x|y=0)P(y=0)} \geq 1$ ①

According to the naive Byes assumption, all the x_i is independent,

We have $P(x|y) = \prod_{j=0}^{d} P(x_j|y)$

Then, rewrite the formula ①, we have ②.

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

Let us denote $P(y=1)$ by p, then $P(y=0)=1-p$;

Let us denote $P(x_j=1|y=1)$ by a_j, $P(x_j=1|y=0)$ by b_j

Because x_j is the binary feature, We have.

$P(x_j=0|y=1) = 1-a_j$; $P(x_j=0|y=0)=1-b_j$

Hense, using the notation above, we have following formula 3:

$$\frac{p}{1-p}\prod_{j=0}^{d}\frac{a_j^{x_j}(1-a_j)^{(1-x_j)}}{b_j^{x_j}(1-b_j)^{(1-x_j)}} \geq 1 \quad (3)$$

Changing the expression of the formula (3), We have.

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

Taking log,

$$\log\left(\frac{p}{1-p}\prod_{j=0}^{d}\frac{1-a_j}{1-b_j}\right)+\sum_{j=0}^{d}x_j\log\left(\frac{a_j}{b_j}\cdot\frac{1-b_j}{1-a_j}\right)\geq 0.$$

Let us denote $b=\log\left(\frac{p}{1-p}\prod_{j=0}^{d}\frac{1-a_j}{1-b_j}\right)$ which is a constant,

Let us denote $w_j=\log\left(\frac{a_j}{b_j}\cdot\frac{1-b_j}{1-a_j}\right)$

Then, we have $b+\sum_{j=0}^{d}x_j w_j \geq 0.$

Therefore, it is a linear classifer for vector $[1, X]$.

And the parameter

$$W^T=[b, w_1, w_2 \cdots w_j]$$

2.

Giving the training data $X = (x_1, x_2 \cdots x_n)$

if using the Naive Bayes model, we assume that $x_1 x_2 \cdots x_n$ is independent and Naive Bayes model is a restrictive generative model, this means that to build the model, we need to computer the $P(x|y)$ and $P(y)$,

The Logistic Regression, on the other hand, is less restrictive Although consistent with the Naive Bayes assumption that the x_i's are conditionally independent given y, but if the data is given that violates this assumption, then the Logistic Regression will adjust the weights to maximize fit to the data.

To conclude, if the dataSet is small, we use the Naive Baye model, the data type require for Naive Baye model is O(log n), if the dataSet is large, the Logistic Regression model is better, the data type require for this model is O(n)

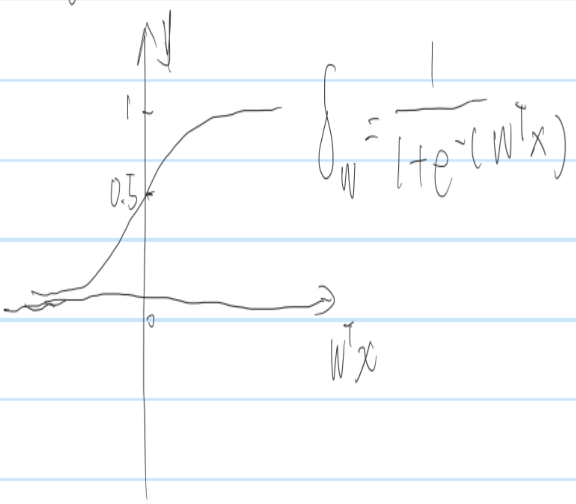
Q3

1)

For standard logistic regression model.

$$\sigma_w = \frac{1}{1+e^{-w^T x}} \quad ①$$

The graph of the function



Giving one training sample (x_i, y_i)

x_i is the training data, y_i is the class label, $y_i \in \{0, 1\}$.

$$P(y_i | x_i; w) = \sigma_w(x)^y (1 - \sigma_w(x))^{1-y}$$

when $y = 1$, $P(1 | x_i; w) = \sigma_w(x)$,

$y = 0$, $P(0 | x_i; w) = 1 - \sigma_w(x)$

(i.e. $\sigma_w(x_i) = 0.8$, then the x_i have 80% become 1)

In order to simplify the problem, we have Log-likelihood:

$$l(w) = \ln[\sigma_w(x)^y (1-\sigma_w(x))^{1-y}]$$
$$= y \ln \sigma_w(x) + (1-y) \ln(1-\sigma_w(x))$$

The fomula above is for one sample, In term of m

the training samples, the Log-likelihood:

$$\text{likelihood} = \sum_{i=1}^{m} y \ln \sigma_w(x) + (1-y) \ln(1-\sigma_w(x)) \quad (2)$$

According to the maximum likelihood method, we maximize $\ell(w)$.

(i.e. $P(1|x_1)=51\%$; $P(1|x_1)=90\%$, we want 90%).

from formula ① $\sigma_w(x)=\frac{1}{1+e^{-(w^Tx)}}$, we have:

$$\ln\frac{\sigma(x)}{1-\sigma(x)}=\ln\frac{P(y=1|x)}{P(y=0|x)}=w^Tx \quad ③$$

We also have:

$$P(y=1|x,w)=\sigma_w(x)=\frac{1}{1+e^{-(w^Tx)}}=\frac{e^{w^Tx}}{1+e^{w^Tx}} \quad ④$$

$$P(y=0|x,w)=1-\sigma_w(x)=\frac{1}{1+e^{w^Tx}} \quad ⑤$$

from ②, ④, ⑤, we have:

$$\text{log-likelihood} = \sum_{i=1}^{m} y \ln \sigma_w(x) + (1-y)\ln(1-\sigma_w(x))$$

$$= \sum_{i=1}^{m} y \ln \frac{e^{w^Tx}}{1+e^{w^Tx}} + (1-y)\ln \frac{1}{1+e^{w^Tx}}$$

$$= \sum_{i=1}^{m} y[\ln e^{w^Tx} - \ln(1+e^{w^Tx})] + (1-y)[\ln 1 - \ln(1+e^{w^Tx})]$$

$$= \sum_{i=1}^{m} y_i w^T x - \ln(1+e^{w^Tx})$$

the loss function = − log-likelihood, therefore:

$$l(w) = \sum_{i=1}^{m} -y_i w^T x_i + \ln(1+e^{w^T x_i})$$

2)

(2) Consider a variant of the logistic regression model:

$$P[y = 1 \mid \mathbf{x}] = f(\mathbf{w}^\top \mathbf{x})$$

where $f : \Re \to [0, 1]$ is a squashing function that maps a real value to a value between 0 and 1.

Write out its loss function.

$$P[y_i \mid x, w] = f(w^T x)^{y} \cdot (1 - f(w^T x))^{1-y}$$

$$l(w) = y \ln(f(w^T x)) + (1-y) \ln(1 - f(w^T x)).$$

For m training sample

$$l(w) = \sum_{i=1}^{m} y \ln(f(w^T x)) + (1-y) \ln(1 - f(w^T x)).$$

therefore, the cost function:

$$\text{loss function} = -\sum_{i=1}^{m} y \ln(f(w^T x)) + (1-y) \ln(1 - f(w^T x)).$$