Cpts_540 hw 10

Lu Xiao

11624097

1.a Compute the prior probabilities P(HaveFun=yes) and P(HaveFun=no).

P(HaveFun=yes) = 6/11

P(HaveFun=no) = 5/11

1.b Compute P(Weather | HaveFun) for all values of Weather Î {clear, cloudy, rain} and HaveFun Î {yes, no}.

P(clear | yes)= 1/2

P(cloudy | yes)= 1/6

P(rain | yes)= 1/3

P(clear | no)= 1/5

P(cloudy | no)= 2/5

P(rain | no)= 2/5

1.c Compute P(AIDone | HaveFun) for all values of AIDone Î {yes, no} and HaveFun Î {yes, no}

P(AIDone yes | yes)= 5/6

P(AIDone no | yes)= 1/ 6

P(AIDone yes | no)= 0

P(AIDone no | no)= 1

1d. Compute P(Costume | HaveFun) for all values of Costume Î {yes, no} and HaveFun Î {yes, no}.

P(Costume yes | yes)= 2/3

P(Costume no | yes)= 1/3

P(Costume yes | no)= 2/5

P(Costume no | no)= 3/5

1e. Compute P(HaveFun=yes | Weather=cloudy, AIDone=yes, Costume=no) and P(HaveFun=no | Weather=cloudy, AIDone=yes, Costume=no).

We have compute the $P(AIDone = yes|Havefun = no) = 0$,

$according\ to\ the$ “add 1 / |values|” technique we can

get $P(AIDone = yes|Havefun = no) = \frac{1}{7}$

Using Bayes Rule we can get

$\mathrm{P(HaveFun = yes \mid Weather = cloudy, AIDone = yes, Costume = no)}$

$$= \frac{\mathrm{P\big(Weather = cloudy, AIDone = yes, Costume = no|Havefun = yes) \times P(Havefun = yes)\big)}}{\mathrm{P(Weather = cloudy, AIDone = yes, Costume = no)}}$$

$$= \frac{\left(\frac{1}{6} \times \frac{5}{6} \times \frac{1}{3}\right) \times \frac{6}{11}}{\frac{3}{11} \times \frac{5}{11} \times \frac{5}{11}}$$

$= \alpha < 0.4481 >$

$\mathrm{P(HaveFun = no \mid Weather = cloudy, AIDone = yes, Costume = no)}$

$$= \frac{\mathrm{P\big(Weather = cloudy, AIDone = yes, Costume = no|Havefun = no) \times P(Havefun = no)\big)}}{\mathrm{P(Weather = cloudy, AIDone = yes, Costume = no)}}$$

$$= \frac{\left(\frac{2}{5} \times \frac{1}{7} \times \frac{3}{5}\right) \times \frac{5}{11}}{\frac{3}{11} \times \frac{5}{11} \times \frac{5}{11}}$$

$= \alpha < 0.2765 >$

$\mathrm{P(HaveFun = yes \mid Weather = cloudy, AIDone = yes, Costume = no) = 0.6188}$

$\mathrm{P(HaveFun = no \mid Weather = cloudy, AIDone = yes, Costume = no) = 0.3818}$

f. Which class would Naïve Bayes choose for the new instance?

Havefun=yes

2.a First, translate the examples (including the HaveFun class value) according to the mapping: clear $\rightarrow$ 1, cloudy $\rightarrow$ 2, rain $\rightarrow$ 3, no $\rightarrow$ 0, yes $\rightarrow$ 1. Show a new table of examples using this mapping.

The table as follows:

	Weather	AIDone	Costume	HaveFun
1	1	1	1	1
2	1	1	0	1
3	1	0	1	1
4	1	0	0	0
5	2	1	1	1
6	2	0	1	0
7	2	0	0	0
8	3	1	1	1
9	3	1	0	1
10	3	0	1	0
11	3	0	0	0

2.b

11 training data as follows:

X	Y
1 1 1	1
1 1 0	1
1 0 1	1
1 0 0	0
2 1 1	1
2 0 1	0
2 0 0	0
3 1 1	1
3 1 0	1
3 0 1	0
3 0 0	0

According to the rule: $\Delta w_i = \eta(y_i - o_i)x_{ij} \quad w_o = 1, \eta = 0.5$

For example1-3:

$\sum w_i x_i => 0, y_1 = 1, \Delta w = 0, w = 1$

For example4:

$$\sum w_i x_i = 2 > 0, y_4 = 1 \;\; y_{true} = 0$$

Since we got a wrong predict we need update weight as followed:

$\Delta w_0 = 0.5(0-1)1 = -0.5\ w_0 = 0.5$

$\Delta w_1 = 0.5(0-1)1 = -0.5\ w_1 = 0.5$

$\Delta w_2 = 0.5(0-1)0 = 0\ w_2 = 1$

$\Delta w_3 = 0.5(0-1)0 = 0\ w_3 = 1$

For example5:

$$\sum w_i x_i = 3 > 0, y_5 = 1, \Delta w = 0, w_0 = 0.5, w_1 = 0.5, w_2 = 1, w_3 = 1$$

For example6: $\sum w_i x_i = 2.5 > 0, y_6 = 1\, , y_{true} = 0$

update weight as followed:

$\Delta w_0 = 0.5(0-1)1 = -0.5, w_0 = 0$

$\Delta w_1 = 0.5(0-1)2 = -1, w_1 = -0.5$

$\Delta w_2 = 0.5(0-1)0 = 0, w_2 = 1$

$\Delta w_3 = 0.5(0-1)1 = -0.5, w_3 = 0.5$

For example7:

$$\sum w_i x_i = -1 < 0, y_7 = 0, \Delta w = 0, w_0 = 0, w_1 = -0.5, w_2 = 1, w_3 = 0.5$$

For example8:

$\sum w_i x_i = 0 = 0, y_8 = 1, \Delta w = 0, w_0 = 0, w_1 = -0.5, w_2 = 1, w_3 = 0.5$

For example9:

$$\sum w_i x_i = -0.5 < 0, y_9 = 0, y_{true} = 1$$

update weight as followed:

$\Delta w_0 = 0.5(1-0)1 = 0.5, w_0 = 0.5$

$\Delta w_1 = 0.5(1-0)3 = 1.5, w_1 = 1$

$\Delta w_2 = 0.5(1-0)1 = 0.5, w_2 = 1.5$

$\Delta w_3 = 0.5(1-0)0 = 0, w_3 = 0.5$

For example10:

$$\sum w_i x_i = 4 > 0, y_1 = 1, y_{true} = 0$$

update weight as followed:

$\Delta w_0 = 0.5(0-1)1 = -0.5, w_0 = 0$

$\Delta w_1 = 0.5(0-1)3 = -1.5, w_1 = -0.5$

$\Delta w_2 = 0.5(0-1)0 = 0, w_2 = 1.5$

$\Delta w_3 = 0.5(0-1)1 = -0.5, w_3 = 0$

For example11:

$\sum w_i x_i = -1.5 < 0, y_1 = 0, \Delta w = 0, w_0 = 0, w_1 = -0.5, w_2 = 1.5, w_3 = 0$

Begin second iteration:

For example 1:

$$\sum w_i x_i = 1 > 0, y_1 = 1, \Delta w = 0, w_0 = 0, w_1 = -0.5, w_2 = 1.5, w_3 = 0$$

For example 2:

$$\sum w_i x_i = 1 > 0, y_1 = 1, \Delta w = 0, w_0 = 0, w_1 = -0.5, w_2 = 1.5, w_3 = 0$$

For example 3:

$$\sum w_i x_i = 0.5 < 0, y_1 = 0, \mathrm{y}_{\mathrm{true}} = 1$$

update weight as followed:

$\Delta w_0 = 0.5(1-0)1 = 0.5, w_0 = 0.5$

$\Delta w_1 = 0.5(1-0)1 = 0.5, w_1 = 0$

$\Delta w_2 = 0.5(1-0)0 = 0, w_2 = 1.5$

$\Delta w_3 = 0.5(1-0)1 = 0.5, w_3 = 0.5$

For example 4:

$$\sum w_i x_i = 0 = 0, y_1 = 1, \mathrm{y}_{\mathrm{true}} = 0$$

update weight as followed:

$\Delta w_0 = 0.5(0-1)1 = -0.5, w_0 = 0$

$\Delta w_1 = 0.5(0-1)1 = -0.5, w_1 = -0.5$

$\Delta w_2 = 0.5(0-1)0 = 0, w_2 = 1.5$

$\Delta w_3 = 0.5(0-1)0 = 0, w_3 = 0.5$

For example 5:

$$\sum w_i x_i = 1.5 > 0, y_1 = 1, \Delta w = 0, w_0 = 0, w_1 = -0.5, w_2 = 1.5, w_3 = 0.5$$

For example 6:

$$\sum w_i x_i = -0.5 < 0, y_1 = 0, \Delta w = 0, w_0 = 0, w_1 = -0.5, w_2 = 1.5, w_3 = 0.5$$

For example 7:

$$\sum w_i x_i = -1 < 0, y_1 = 1, \Delta w = 0, w_0 = 0, w_1 = -0.5, w_2 = 1.5, w_3 = 0.5$$

For example 8:

$$\sum w_i x_i = 0.5 > 0, y_1 = 1, \Delta w = 0, w_0 = 0, w_1 = -0.5, w_2 = 1.5, w_3 = 0.5$$

For example 9:

$$\sum w_i x_i = 0 = 0, y_1 = 1, \Delta w = 0, w_0 = 0, w_1 = -0.5, w_2 = 1.5, w_3 = 0.5$$

For example 10:

$$\sum w_i x_i = -1, y_1 = 0, \Delta w = 0, w_0 = 0, w_1 = -0.5, w_2 = 1.5, w_3 = 0.5$$

For example 11:

$$\sum w_i x_i = -1.5 < 0, y_1 = 0, \Delta w = 0, w_0 = 0, w_1 = -0.5, w_2 = 1.5, w_3 = 0.5$$

Begin 3rd iteration:

For example 1:

$$\sum w_i x_i = 1.5 > 0, y_1 = 1, \Delta w = 0, w_0 = 0, w_1 = -0.5, w_2 = 1.5, w_3 = 0.5$$

For example 2:

$$\sum w_i x_i = 1 > 0, y_1 = 1, \Delta w = 0, w_0 = 0, w_1 = -0.5, w_2 = 1.5, w_3 = 0.5$$

For example 3:

$\sum w_i x_i = 0 = 0, y_1 = 1, \Delta w = 0, w_0 = 0, w_1 = -0.5, w_2 = 1.5, w_3 = 0.5$

For example 4:

$\sum w_i x_i = -0.5 < 0, y_1 = 1, \Delta w = 0, w_0 = 0, w_1 = -0.5, w_2 = 1.5, w_3 = 0.5$

The final weight as followed:

$w_0 = 0, w_1 = -0.5, w_2 = 1.5, w_3 = 0.5$

c. How would the learned perceptron classify the new instance <Weather=cloudy, AIDone=yes, Costume=no> Show your work.

According to the weight $w_0 = 0, w_1 = -0.5, w_2 = 1.5, w_3 = 0.5$ we can get $1 \times 0 + 2 \times (-0.5) + 1 \times (1.5) + 0 \times (0.5) = 0.5$

Therefore, it will classify the new instance <Weather=cloudy, AIDone=yes, Costume=no> to Havefun=true.

3

input arff file as followed:

```
@relation predict
@attribute Weather    {Clear, Cloudy, Rain}
@attribute AIDone       {yes, no}
@attribute Costume       {yes, no}
@attribute HaveFun.   {yes, no}

@data
Clear,yes,yes,yes
Clear,yes,no,yes
Clear,no,yes,yes
Clear,no,no,no
Cloudy,yes,yes,yes
Cloudy,no,yes,no
Cloudy,no,no,no
Rain,yes,yes,yes
Rain,yes,no,yes
Rain,no,yes,no
```

Rain,no,no,no

output:

=== Run information ===

Scheme: weka.classifiers.bayes.NaiveBayes

Relation: predict

Instances: 11

Attributes: 4

Weather

AIDone

Costume

HaveFun.

Test mode: evaluate on training data

=== Classifier model (full training set) ===

Naive Bayes Classifier

	Class	
Attribute	yes	no
	(0.54)	(0.46)
Weather		
Clear	4.0	2.0
Cloudy	2.0	3.0
Rain	3.0	3.0
[total]	9.0	8.0
AIDone		
yes	6.0	1.0
no	2.0	6.0
[total]	8.0	7.0

Costume

yes	5.0	3.0
no	3.0	4.0
[total]	8.0	7.0

Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances	10	90.9091 %
Incorrectly Classified Instances	1	9.0909 %
Kappa statistic	0.8197	
Mean absolute error	0.2093	
Root mean squared error	0.2438	
Relative absolute error	42.1626 %	
Root relative squared error	48.9498 %	
Total Number of Instances	11	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.833	0.000	1.000	0.833	0.909	0.833	1.000	1.000	yes
1.000	0.167	0.833	1.000	0.909	0.833	1.000	1.000	no

Weighted Avg. 0.909 0.076 0.924 0.909 0.909 0.833
1.000 1.000

=== Confusion Matrix ===

a b <-- classified as
5 1 | a = yes
0 5 | b = no