Naive Bayes Classifier - II

Contents

Laplace Smoothing

- If a given class and feature value never occur together in the training data, then the frequency-based probability estimate will be zero.
- This is problematic because it will wipe out all information in the other probabilities when they are multiplied.
- Therefore, it is often desirable to incorporate a small-sample correction, called pseudo-count, in all probability estimates such that no probability is ever set to be exactly zero.
- This way of regularising naive Bayes is called Laplace Smoothing when the pseudo count is one, and Lidstone Smoothing in the general case.

Laplace Smoothing

$$P(x = x_i \mid y = y_j) = f_i / N_j$$

This prob will be 0 if numerator count (f_i) is 0

Laplace smoothing will replace this probability with a value obtained by the formula:

$$\boxed{\hat{\theta}_i = \frac{F_i + \alpha}{Nj + \alpha d}}$$

where

α : Smoothing Parameter

N_j : Number of observations for $Y = y_j$

d_i : Number of classes of x_i

Laplace Smoothing in R

```
# Importing Data
```

```
data1<-read.csv("Data for Laplace Smoothing.csv",header=T)

data1$X1<-as.factor(data1$X1)

data1
```

Y	X1	X2	X3
0	1	M	A
0	2	M	A
0	2	M	A
0	1	M	A
0	2	F	A
1	2	F	A
1	2	M	B
1	2	M	B
1	2	M	B
1	2	M	B
1	2	F	B
1	2	F	B
1	2	M	B
1	2	M	A
0	2	M	A
0	2	F	A
0	1	F	A
0	1	F	B
0	1	F	B
1	2	M	B
1	2	M	A
1	2	M	A
1	2	M	A
1	2	F	B
1	2	F	B
1	2	F	B
1	2	M	B
1	2	M	B

Variable X1 is a factor with two levels, 1 & 2. There is no observation in the data with X1 =1 when the dependent variable Y =1. Hence, P(X1 | Y=1) =0. We thus introduce smoothing, to avoid loss of information.

Laplace Smoothing in R

```
# Naive Bayes Model with Laplace Smoothing

model<-naiveBayes(Y~X1+X2+X3,data=data1)

laplacemodel<-naiveBayes(Y~X1+X2+X3,data=data1,laplace=2)
```

We first run the default Naive Bayes model.

laplace= tells R the value of pseudo-count to be used to smoothen the model.

$$\hat{\theta}_i = \frac{F_i + \alpha}{Nj + \alpha d} = \frac{2}{18 + 2 \times 2} = 0.09090$$

Since there are no observations for XI=I and Y=I,
f_i=0; α=2
Total no. of Y=I is 18
XI is a factor with two classes, hence d=2

* There is no rule for choosing the appropriate pseudo-count. The number should be low, but not so low that the resulting probability is almost 0.

Laplace Smoothing in R

```
model
```

```
# Output
```

```
> model

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
        0         1 
0.3571429 0.6428571 

Conditional probabilities:
   X1
Y     1   2
  0 0.5 0.5
  1 0.0 1.0

   X2
Y           F         M
  0 0.5000000 0.5000000
  1 0.3333333 0.6666667

   X3
Y           A         B
  0 0.8000000 0.2000000
  1 0.2777778 0.7222222
```

Interpretation : Conditional probability of X1=1|Y=1 is 0.

Laplace Smoothing in R

```
laplacemodel
```

```
# Output
```

```
> laplacemodel

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
        0         1 
0.3571429 0.6428571 

Conditional probabilities:
   X1
Y            1          2
  0 0.50000000 0.50000000
  1 0.09090909 0.90909091

   X2
Y           F         M
  0 0.5000000 0.5000000
  1 0.3636364 0.6363636

   X3
Y           A         B
  0 0.7142857 0.2857143
  1 0.3181818 0.6818182
```

Interpretation :
R has now replaced 0 with 0.0909 (Calculated using the Laplace smoothing formula).

Predictions After Smoothing

```
# Importing and Readying New Data
newdata1<-read.csv("New Data for Laplace Predictions.csv",header=T)

newdata1
# Output
 Y X1 X2 X3
1 1  1  M  A
2 0  2  M  A
3 0  2  M  A
4 1  1  M  A

newdata1$X1<-as.factor(newdata1$X1)
```

- New data to be used for predictions is saved as an object named **newdata1**.
- New data contains observations which were absent in training data, i.e. conditional probability in training data was zero.
- **as.factor()** converts X1 to factor variable.

```
# Predictions
prednew<-predict(laplacemodel,newdata1,type="raw")
prednew1<-predict(laplacemodel,newdata1,type="raw",
                  threshold=0.1,eps=0.1)
```

- **threshold=** and **eps=** are added to ensure predicted probabilities are not too low. Threshold is the value that replaces values within the eps range.
- Here, probabilities<=0.1 are replaced by 0.1. Defaults are **threshold=**0.001 and **eps=**0.

Predictions After Smoothing

```
# Predictions
prednew
```

```
# Output
> prednew
                0         1
[1,] 0.8434941 0.1565059
[2,] 0.3502079 0.6497921
[3,] 0.3502079 0.6497921
[4,] 0.8434941 0.1565059
```

```
prednew1
```

```
# Output
> prednew1
                0         1
[1,] 0.8304964 0.1695036
[2,] 0.3502079 0.6497921
[3,] 0.3502079 0.6497921
[4,] 0.8304964 0.1695036
```

Interpretation :
Predicted probabilities using just the smoothened model and by using additional constraints of epsilon and threshold are different.

Quick Recap

Laplace Smoothing

- If a given class and feature value never occur together in the training data, then the frequency-based probability estimate will be zero.
- A pseudo-count is incorporated, in all probability estimates such that no probability is ever set to be exactly zero.
- This way of regularizing naive Bayes is called Laplace Smoothing
- **naiveBayes**(Y~X_i,**data=,laplace=**)