Introduction to Data Science

DSA1101

Semester 1, 2018/2019
Week 9

The Naïve Bayes Classifier

Naïve Bayes Classifier

- *Naïve Bayes* is a probabilistic classification method based on Bayes' theorem (or Bayes' law) with a few tweaks
- Bayes' theorem gives the relationship between the probabilities of two events and their conditional probabilities.

Naïve Bayes Classifier: Example

- The example is on using naïve Bayes classifier to predict whether employees would enroll in an onsite educational program based on feature variables such as `Age`, `Income`, `JobStatisfaction` and `Desire`.
- We will illlustrate with both manual calculation and using the `naiveBayes` function in the R package 'e1071'.

Naïve Bayes Classifier: Example in R

- The CSV file 'sample1.csv' has been posted to IVLE, and contains 15 records.
- The last record does not contain any outcome label for `Enrolls`
- ```
  > sample <- read.table("sample1.csv",header=TRUE,sep=",")
  > head(sample)
         Age Income JobSatisfaction
  1     <=30   High              No
  2     <=30   High              No
  3 31 to 40   High              No
  4      >40 Medium              No
  5      >40    Low             Yes
  6      >40    Low             Yes
       Desire Enrolls
  1      Fair      No
  2 Excellent      No
  3      Fair     Yes
  4      Fair     Yes
  5      Fair     Yes
  6 Excellent      No
  ```
  ```

Naïve Bayes Classifier: Example in R

- Two data frame objects called `traindata` and `testdata` are created for the naïve Bayes Classifier
- We will train the classifier using `traindata`, then make predictions for the single record in `testdata`
- ```
  > traindata <- as.data.frame(sample[1:14,])
  > testdata <- as.data.frame(sample[15,])
  > testdata
      Age Income JobSatisfaction Desire
  15 <=30 Medium             Yes   Fair
     Enrolls
  15
  ```
  ```

Naïve Bayes Classifier: Example in R

- We will first illustrate the naïve Bayes classifier via manual computation
- Recall that we need to compute the probabilities $P(Y = y_j)$ for $j = 1, 2, ..., k$.
-

```
> tprior <- table(traindata$Enrolls)
> tprior

     No  Yes
  0   5   9
> tprior <- tprior/sum(tprior)
> tprior

                 No       Yes
0.0000000 0.3571429 0.6428571
```

Naïve Bayes Classifier: Example in R

- We will first illustrate the naïve Bayes classifier via manual computation
- Recall that we need to compute the probabilities $P(Y = y_j)$ for $j = 1, 2, ..., k$.
- Here since Y is binary, we just compute $P(Y = 1)$ (for yes) and $P(Y = 0)$ (for no).
-

```
> tprior <- table(traindata$Enrolls)
> tprior

     No  Yes
  0   5   9
> tprior <- tprior/sum(tprior)
> tprior

                  No       Yes
0.0000000 0.3571429 0.6428571
```

Naïve Bayes Classifier: Example in R

- Next, we need to compute the conditional probabilities $P(X_i = x_i | Y = 1)$ and $P(X_i = x_i | Y = 0)$, where $i = 1, 2, 3, 4$ for the feature variables $X = \{Age, IncomeJobSatisfaction, Desire\}$.
- First, compute the conditional probabilities for `Age`:
-

```
> ageCounts <- table(traindata[,c("Enrolls", "Age"
  )])
> ageCounts
       Age
Enrolls <=30 >40 31 to 40
            0   0        0
     No     3   2        0
     Yes    2   3        4
> ageCounts <- ageCounts/rowSums(ageCounts)
> ageCounts
       Age
Enrolls       <=30       >40  31 to 40

     No  0.6000000 0.4000000 0.0000000
     Yes 0.2222222 0.3333333 0.4444444
```

Naïve Bayes Classifier: Example in R

- We perform similar operations for `Income`, `JobSatisfaction` and `Desire`:
- ```
  > incomeCounts <- table(traindata[,c("Enrolls", "Income")])
  > incomeCounts <- incomeCounts/rowSums(incomeCounts)
  > incomeCounts
         Income
  Enrolls       High       Low    Medium

      No   0.4000000 0.2000000 0.4000000
      Yes  0.2222222 0.3333333 0.4444444
  ```
  ```

Naïve Bayes Classifier: Example in R

- We perform similar operations for `Income`, `JobSatisfaction` and `Desire`:
- ```
  > jsCounts <- table(traindata[,c("Enrolls", "
      JobSatisfaction")])
  > jsCounts <- jsCounts/rowSums(jsCounts)
  > jsCounts
           JobSatisfaction
  Enrolls          No        Yes

      No   0.8000000 0.2000000
      Yes  0.3333333 0.6666667
  ```
  ```

Naïve Bayes Classifier: Example in R

- We perform similar operations for `Income`, `JobSatisfaction` and `Desire`:
-
```
> desireCounts <- table(traindata[,c("Enrolls", "Desire")])
> desireCounts <- desireCounts/rowSums(desireCounts)
> desireCounts
        Desire
Enrolls Excellent      Fair

     No  0.6000000 0.4000000
     Yes 0.3333333 0.6666667
```

Naïve Bayes Classifier: Example in R

- We finally compute the probability scores

$$P(Y = 1|X) \propto P(Y = 1) \times \prod_{i=1}^{4} P(X_i = x_i|Y = 1)$$

and

$$P(Y = 0|X) \propto P(Y = 0) \times \prod_{i=1}^{4} P(X_i = x_i|Y = 0)$$

Naïve Bayes Classifier: Example in R

- Computation of the probability scores:
-
```r
prob_yes <-
ageCounts["Yes",testdata[,c("Age")]]*
incomeCounts["Yes",testdata[,c("Income")]]*
jsCounts["Yes",testdata[,c("JobSatisfaction")]]*
desireCounts["Yes",testdata[,c("Desire")]]*
tprior["Yes"]


prob_no <-
ageCounts["No",testdata[,c("Age")]]*
incomeCounts["No",testdata[,c("Income")]]*
jsCounts["No",testdata[,c("JobSatisfaction")]]*
desireCounts["No",testdata[,c("Desire")]]*
tprior["No"]
```

Naïve Bayes Classifier: Example in R

- Since $P(Y = 1|X) > P(Y = 0|X)$, the predicted result for the test record is yes for `Enrolls`.
-
```
> prob_yes
      Yes
0.02821869
> prob_no
        No
0.006857143
```

Naïve Bayes Classifier: Example in R

- Alternatively, we can use the `naiveBayes` function in the R package 'e1071' to perform naïve Bayes classification:
-
```
> library(e1071)
>
> model <- naiveBayes(Enrolls ~
+ Age+Income+JobSatisfaction+Desire,
+ traindata, laplace=0)
>
> results <- predict(model,testdata)
> results
[1] Yes
Levels:  No Yes
```

Naïve Bayes Classifier: Diagnostics

- Recall that for diagnostics of classifiers, we have learnt about the *confusion matrix* as well as measures such as accuracy, true positive rate etc.
- We will familiarize ourselves with one additional diagnostics tool, the Receiver Operating Characteristic (ROC) curve.

Naïve Bayes Classifier: Diagnostics

- Recall that the False Positive Rate (FPR) and True Positive Rate (TPR) are calculated as

$$\text{FPR} = \frac{FP}{FP + TN}$$

and

$$\text{TPR} = \frac{TP}{TP + FN}$$

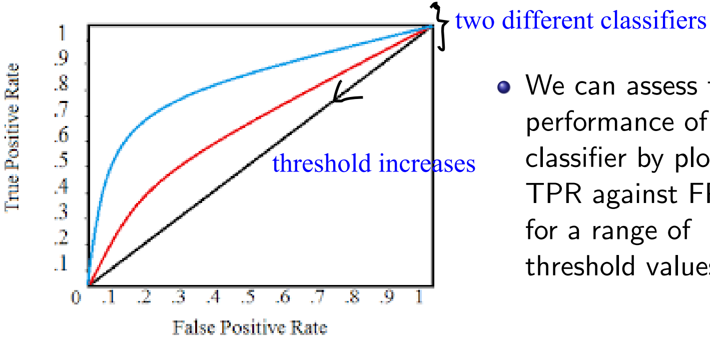
		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positives (TP)	False Negatives (FN)
	Negative	False Positives (FP)	True Negatives (TN)

Naïve Bayes Classifier: Diagnostics

- Recall that for classification using the majority rule, Y is predicted to be 1 if $\hat{Y} > 0.5$ and 0 otherwise.
- If the threshold is increased, then *less* test objects will be predicted to be 1, and so TP will be either constant or decreases. However, the sum $TP + FN$ is still constant because the number of objects with actual label $Y = 1$ is a constant in the test dataset, so TPR will either be constant or increases.
- Similarly, if the threshold is increased, FP will be either constant or increases, while the sum $FP + TN$ is a constant, so FPR will either be constant or increases.
- Therefore, in summary, if the threshold is increased, both TPR and FPR generally ~~increase~~ decrease

Find analogy in life?

Naïve Bayes Classifier: Diagnostics

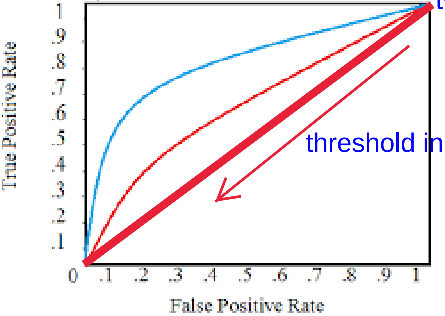


- We can assess the performance of a classifier by plotting TPR against FPR for a range of threshold values

Naïve Bayes Classifier: Diagnostics

the blue curve is better,
higher area under the ROC curve.

two different classifiers



threshold increases

- A useful metric is to compute the area under the ROC curve (AUC)
- We will talk more about the ROC curve next week

for a given FPR, if delta(TPR) is maximum
-> the best threshold.
A good model should have a high TPR and low FPR.