Week 6 Topics

1. Chapter 8 – Naïve Bayes General View

# Introduction

Naïve Bayes is a simple but surprisingly powerful algorithm for predictive modeling.

In this paper you will discover the Naïve Bayes algorithm for classification with the help of R. After reading this paper and doing examples, you will know:

* How, a *learned model,* can be used to classify and make predictions. Which also, applies to KNN and classification (decision) tree algorithms.
* How can learn a Naïve Bayes model from training data.
* How to best prepare your data for the Naïve Bayes algorithm.

# Bayes Algorithm: Bayesian Classifier (Exact)

The basic principle is simple. For each record(observation) to be classified:

1. Find all the other records with the same predictor profile (i.e. where the predictor values are the same).
2. Determine what classes the records belong to and which class is most prevalent.
3. Assign that class to the new record.

Alternatively, it might be desirable to twist the method so that it answers the questions: “What is the propensity of belonging to the class of interest?” Obtaining the class probabilities allows using a sliding cutoff to classify a record as belonging to class i, even i is not the most probable class for that record. When we use the cutoff approach, the above steps are modified as follow:

1. Establish a cutoff probability for the class of interest above which we consider that a record belong to that class.
2. Find all the training records with the same predictor profile as the new record (i.e. where the predictor values are the same).
3. Determine the probability that those records belong to the class of interest
4. If the probability is above the cutoff probability, assign the new record to the class of interest.

# Conditional probability

Conditional probability is probability of event A given that the event B is true (or occurred), P(A|B). Outcome Y is a function of one or more predictors. That is Y = f(P1, P2, ..., Pi, …, Pp) and it is presented as a record (observation).

Regarding conditional probability, in our dataset, we will be looking at the probability of the record belonging to class k given that its predictors’ value are x1, x2, …, xp where p is number of predictors. In general, for a response with m classes, C1, C2, …, Cm and predictors’ value x1, x2, …, xp we want to compute P(Ck| x1, x2, …, xp)

To classify a record, we compute its probability of belonging to each of classes in this way, then classify the record to the class that has the highest probability. or use the cutoff probability to decide whether it should be assigned to the class of interest.

We see that the Bayesian classifier works only with categorical predictors. If we use a set of numerical predictors, then it is highly unlikely that multiple records will have identical values on these numerical predictors. Therefore, numerical predictors must be binned and converted to categorical predictors.

In conclusion:

*The Bayesian classifier is the only classification or prediction method that is especially suited for (and limited to) categorical predictor values.*

Textbook Example: Predicting Fraudulent Financial Reports, two predictors.

Each customer submit an annual financial report to the accounting firm, which is then audited by the accounting firm. For simplicity, the outcome of audit as classified *Truthful* or *Fraudulent*, referring to the accounting firm’s assessment of the customer financial report. Accounting firm notes that, in addition to all the financial records, it has information on whether or not the customer has had *prior legal trouble*. This information has not been used in previous audits. Now, accounting firm wants to know whether having had prior legal trouble is predictive of fraudulent reporting! The data sample is shown below (figure 1).

|  |  |  |  |
| --- | --- | --- | --- |
| Company | Prior Legal Trouble | Company Size | Status |
| 1 | Yes | Small | Truthful |
| 2 | No | Small | Truthful |
| 3 | No | Large | Truthful |
| 4 | No | Large | Truthful |
| 5 | No | Small | Truthful |
| 6 | No | Small | Truthful |
| 7 | Yes | Small | Fraudulent |
| 8 | Yes | Large | Fraudulent |
| 9 | No | Large | Fraudulent |
| 10 | Yes | Large | Fraudulent |

Figure 1: Accounting firm sample data.

We need to answer the following conditional probabilities:

P(Truthful | Prior Legal Trouble, Small) =?

P(Prior Legal Trouble |Truthful) = ?

The response of interest is Y = {fraudulent, truthful}), has two classes C1 = fraudulent and C2 = truthful. The predictor *prior legal trouble* has two possible values: 0 (no *prior legal trouble)* and 1 (*prior legal trouble*).

The accounting firm has data on 1500 companies (customers) that it has already investigated. The data were partitioned into a training set (1000 firms) and a validation set of 500 firms. Figure 2 illustrates the pivot table of these counts.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Prior Legal (X = 1) | No Prior Legal (X = 0) | Total |
| Fraudulent C1 | 50 | 50 | 100 |
| Truthful C2 | 180 | 720 | 900 |
| Total | 230 | 770 | 1000 |

Figure 2: Training Data summary

Applying the full (Exact) Bayesian Classifier.

Now consider the financial report from a new company, which we want to classify as either fraudulent or truthful by using the this dataset. To do this we calculate the probabilities of belonging to each of the two classes.

If the company had prior legal trouble, the probability of belonging to the fraudulent class should be P(fraudulent |prior legal) = 50/230 and the probability of belonging to other class, “truthful”, is 180/230

Using the “Assign to the Most Probable Class” Approach

If the company had prior legal trouble, then the new record is assigned to “truthful” class (higher probability). On the other hand if the if the company had no prior legal trouble, the probability of belonging to the fraudulent class should be P(fraudulent |no prior legal) = 50/770 and the probability of belonging to other class, “truthful”, is 720/770 and the new record is assigned to “truthful” class.

|  |  |  |
| --- | --- | --- |
|  | Prior-Legal Trouble | No Prior Legal Trouble |
| Fraudulent | 50/230 = 20% | 50/770 = 6.8% |
| Truthful | 180/230 = 78% | 720/770 = 98% |

Using the “Cutoff Probability” Approach

In this example we are more interested in identifying the fraudulent report. We know, in order to identify the fraudulent reports, some truthful report must will be misidentified (or misclassified) as fraudulent, and overall classification accuracy may decline. Our approach is therefore to establish a cutoff value for the probability of being fraudulent, and classify all records above that value as fraudulent. The Bayesian formula for the calculation of this probability that a record belong to class i (i.e. Ci) is as follow.

Using the formula for our example:

Considering both predictors *Prior Legal* and *Company size* and choosing class of interest Fraudulent given *Prior Legal* = y and *Company size* = s

LHS probabilities are extractable from our dataset.

To understand these conditional probabilities let me create two tables one is prior probabilities P(F) and P(T) where F = fraudulent and T = truthful. We copy the sample data in figure 2 to create the following table (Figure 2.b).

|  |  |  |  |
| --- | --- | --- | --- |
| Company | Prior Legal Trouble | Company Size | Status |
| 1 | Yes | Small | Truthful |
| 2 | No | Small | Truthful |
| 3 | No | Large | Truthful |
| 4 | No | Large | Truthful |
| 5 | No | Small | Truthful |
| 6 | No | Small | Truthful |
| 7 | Yes | Small | Fraudulent |
| 8 | Yes | Large | Fraudulent |
| 9 | No | Large | Fraudulent |
| 10 | Yes | Large | Fraudulent |

Figure 2.b

Practical Difficulty with complete (Exact) Bayes Algorithm

The complete Naïve Bayes Algorithm we discuss above amounts to finding all records in the sample data that are exactly like the new record to be classified in the sense that all the predictor values are all identical. This is easy in small example we had, where we had one and then two predictors.

When number of predictors get even slightly larger, then many new records to be classified may be without exact matches. To solve this problem we use the Naïve Bayes Algorithm.

Naïve Bayes Algorithm

In the Naïve Bayes algorithm, we are not restricting probability calculation to those records that match the record to be classified. Instead, we use the entire data set. We look again to the steps we had at the beginning of this paper:

1. Find all the other records with the same predictor profile (i.e. where the predictor values are the same).
2. Determine what classes the records belong to and which class is most prevalent.
3. Assign that class to the new record.

and rewrite them for Naïve Bayes as follow:

1. For the class C1, estimate the individual conditional probabilities for each predictor P(X*j*|C1) – these are probabilities that the predictor value in the record to be classified occurs in class C1. For example, for X1 this probability is estimated by proportion of x1 value among the C1 records in the training set.

In our example, C1 could be “truthful”, X1 could be “*Prior Legal Trouble”* and x1 could be “Yes”. Thus, P(Yes | truthful) = 1/6

1. Multiply these probabilities by each other, then by the proportion of record belonging to class C1 (e.g. P(T) = 6/10)
2. Repeat step 1 and 2 for all classes.
3. Estimate a probability C*i* by taking the value calculated in step 2 and divide it by sum of such value for all classes
4. Assign the new record to the class with highest probability for this set of predictor values.

The following is the general formula for the Naïve Bayes algorithm for *m* classes and p predictors. The formula is written only for class C1 but easily can be modifies to represent Naïve Bayesian probability (Pnb) for all other classes.

Back to our example with two classes C1 = fraudulent (F) and C2 = truthful (T) with two predictors

Prior Legal Trouble X1 = {x1 = yes and x2 = no}

and

Company Size X2 = { x1 = small and x2 = large}

Calculating for fraudulent class with prior legal trouble yes and company size small. That is *Pnb = (fraudulent| Prior Legal = yes, Company Size = small)*

These probabilities and conditional probabilities are summarized in the following figures



Figure 3 Prior class priorities

Figure 4: Conditional probabilities

# Naïve Bayes Algorithm Revisited

Naïve Bayes is a simple but surprisingly powerful algorithm for predictive modeling.

In this paper you already discovered the Naïve Bayes algorithm for classification later we will learn how it is implemented with the help of R. However, let’s look at it again.

In machine learning we are often interested in selecting the best hypothesis (h) given data (d). In your terminology, selecting the best model given the data we have!

One of the easiest ways of selecting the most probable hypothesis given the data that we have is the one that we can use as our prior knowledge about the problem. Naïve Bayes’ Theorem provides a way that we can “*calculate the probability of a hypothesis given our prior knowledge”.*

I try to minimal use of probability in my explanation ☺

# The Naïve Bayes Theorem

In machine learning we are often interested in selecting the best hypothesis (h) given data (d).

One of the easiest ways of selecting the most probable hypothesis given the data that we have that we can use as our prior knowledge about the problem. Naïve Bayes’ theorem provides a way that we can calculate the probability of a hypothesis given our prior knowledge.

Bayes’ Theorem is stated as:

Where:

* **P(h|d)** is the probability of hypothesis *h* given the data *d.* This is called the posterior probability.
* **P(d|h)** is the probability of data *d* given that the hypothesis *h* was true.
* **P(h)** is the probability of hypothesis *h* being true (regardless of the data). This is called the prior probability of hypothesis *h*.
* **P(d)** is the probability of the data (regardless of the hypothesis)

When we use the Naïve Bayes algorithm, we are interested in calculating the posterior probability of P(h|d) from the prior probability p(h) with P(d) and P(d|h).

In data mining, after calculating the posterior probability for a number of different hypotheses, you can select the hypothesis with the highest probability. This is the maximum probable hypothesis and may formally be called the Maximum a Posteriori (MAP) hypothesis

We can formulate the above statistical statement into the following

MAP(h) = max(P(h |d)) or MAP(h) =

or

MAP(h) = max(P(d |h) x P(h))

The P(d) is a normalizing term which allows us to calculate the probability. We can drop it when we are interested in the most probable hypothesis as it is constant and only used to normalize.

If we have an even number of instances in each class in our training data, then the probability of each class (e.g. P(h)) will be equal. Again, this would be a constant term in our equation and we could drop it so that we end up with:

MAP(h) = max(P(d|h))

This is a useful exercise, because when reading up further on Naïve Bayes you may see all of these forms of the theorem.

What is the Naïve Bayes Classifier?

Naïve Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values.

It is called Naïve Bayes or idiot Bayes because the calculation of the probabilities for each hypothesis are simplified to make their calculation tractable. Rather than attempting to calculate the values of each attribute value P(d1, d2, d3|h), they are assumed to be conditionally independent given the target value and calculated as P(d1|h) x P(d2|h) and so on. Remember ***we******already talk about it under Exact or Complete Bayesian classifier and Naïve Bayes classifier***

This is a very strong assumption that is most unlikely in real data, i.e. that the attributes do not interact. Nevertheless, the approach performs surprisingly well on data where this assumption does not hold.

Representation Used By Naïve Bayes Models

The representation for Naïve Bayes is probabilities.

A list of probabilities are stored to file for a learned Naïve Bayes model. This includes:

* **Class Probabilities**: The probabilities of each class in the training dataset.
* **Conditional Probabilities**: The conditional probabilities of each input value or predictor, given each class value.

# Learn a Naïve Bayes Model From Data

Learning a Naïve Bayes model from the training data is fast.

Training is fast because only the probability of each class and the probability of each class given different input (x) values need to be calculated. No coefficients need to be fitted by optimization procedures.

* **Calculating Class Probabilities**

The class probabilities are simply the frequency of instances that belong to each class divided by the total number of instances.

For example in a binary classification the probability of an instance belonging to class 1 would be calculated as:

*P(class=1) = count(class=1) / (count(class=0) + count(class=1))*

In the simplest case each class would have the probability of 0.5 or 50% for a binary classification problem with the same number of instances in each class.

* **Calculating Conditional Probabilities**

The conditional probabilities are the frequency of each attribute value for a given class value divided by the frequency of instances with that class value.

For example, if a “weather” attribute had the values “sunny” and “rainy” and the class attribute had the class values “go-out” and “stay-home“, then the conditional probabilities of each weather value for each class value could be calculated as:

P(weather=sunny|class=go-out) = count(instances with weather=sunny and class=go-out) / count(instances with class=go-out)

P(weather=sunny|class=stay-home) = count(instances with weather=sunny and class=stay-home) / count(instances with class=stay-home)

P(weather=rainy|class=go-out) = count(instances with weather=rainy and class=go-out) / count(instances with class=go-out)

P(weather=rainy|class=stay-home) = count(instances with weather=rainy and class=stay-home) / count(instances with class=stay-home)

* **Make Prediction with a Naïve Bayes Model**

Given a Naïve Bayes model, we can make predictions for new data using Bayes theorem.

MAP(h) = max(P(d|h) x P(h))

Using our example above, if we had a new instance with the weather of sunny, we can calculate:

go-out = P(weather=sunny|class=go-out) x P(class=go-out)

stay-home = P(weather=sunny|class=stay-home) x P(class=stay-home)

We can choose the class that has the largest calculated value. We can turn these values into probabilities by normalizing them as follows:

P(go-out|weather=sunny) = go-out / (go-out + stay-home)

P(stay-home|weather=sunny) = stay-home / (go-out + stay-home)

If we had more input variables we could extend the above example. For example, pretend you have a “car” attribute with the values “working” and “broken“. We can multiply this probability into the equation.

For example below is the calculation for the “go-out” class label with the addition of the car input variable set to “working”:

go-out = P(weather=sunny|class=go-out) x P(car=working|class=go-out) x P(class=go-out)

1. Naïve Bayes Classifier – A simple Example

Bayesian classification is based on Bayes' Theorem. Bayesian classifiers are the statistical classifiers. Bayesian classifiers can predict class membership probabilities such as the probability that a given tuple belongs to a particular class.

## Bayes’ Theorem - Revisited

There are two types of probabilities

Posterior Probability [P(h/d)]

Prior Probability [P(h)]

where d is data tuple and h is some hypothesis.

According to Bayes' Theorem,

I will the Naïve Bayes Classification Algorithm through a very simple example. Consider the example given below:

As indicated in the figure 3, the objects can be classified as either GREEN or RED. Our task is to classify new cases as they arrive, i.e., decide to which class label they belong, based on the currently existing objects. There are 40 GREEN and 20 RED objects.

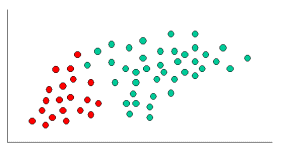


Figure 5: Naïve Baye Classification Example

Since there are twice as many GREEN objects as RED, it is reasonable to believe that a new case (which hasn't been observed yet) is twice as likely to have membership GREEN rather than RED. In the Bayesian analysis, this belief is known as the prior probability. Prior probabilities are based on previous experience, in this case the percentage of GREEN and RED objects, and often used to predict outcomes before they actually happen.

Therefore, we can write (our model):

P(h = Green) is Prior Probability of GREEN: number of GREEN objects / total number of objects

P(h = Red) is Prior Probability of RED: number of RED objects / total number of objects

Since there is a total of 60 objects, 40 of which are GREEN and 20 RED, our prior probabilities for class membership are:

Prior Probability for GREEN: 40 / 60 = 2/3 ≈ %67

Prior Probability for RED: 20 / 60 = 1/3 ≈ %33

Having formulated our prior probability (our model), we are now ready to classify a new object (WHITE dot (e.g. X) in the diagram below, Figure 6). Since the objects are well clustered, it is reasonable to assume that the more GREEN (or RED) objects in the vicinity of X, the more likely that the new cases belong to that particular color. To measure this likelihood, we draw a circle around X which encompasses a number (to be chosen a priori) of points irrespective of their class labels. Then we calculate the number of points in the circle belonging to each class label. From this we calculate the likelihood:

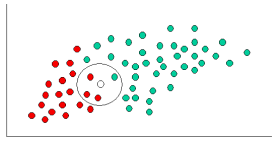


Figure 6. Classification example with new object

P(d|h) = P(number of Green in circle given class Green) =

And P(d|h) = P(number of Red in circle given class Red) =

From the illustration above, it is clear that Likelihood of X given GREEN is smaller than Likelihood of X given RED, since the circle encompasses 1 GREEN object and 3 RED ones. Thus:

Although the prior probabilities indicate that X may belong to GREEN (given that there are twice as many GREEN compared to RED) the likelihood indicates otherwise; that the class membership of X is RED (given that there are more RED objects in the vicinity of X than GREEN). In the Bayesian analysis, the final classification is produced by combining both sources of information, i.e., the prior and the likelihood, to form a posterior probability using the so-called Bayes' rule.

Posterior probability of X being GREEN ∝ Prior probability of GREEN x Likelihood of X given GREEN (probability of red in selected vicinity)

Thus:

Posterior probability of X being GREEN = 2/3 x 1/40 = 1/60

And:

Posterior probability of X being RED ∝ Prior probability of RED x Likelihood of X given RED

Thus:

Posterior probability of X being RED = 1/3 x 3/20 = 1/20

Finally, we classify X as RED since its class membership achieves the largest posterior

As you noticed we didn’t use the denominator of the Naïve Bayes general formula because it is the same in both calculating the Red and Green posterior probability.

1. Application of the Naïve Bayes Classifier in R

When we use Naïve Bays algorithm as a classifier all attributes should be categorical. If we have any numerical attributes in our dataset, first we have to change them into categorical.

The following codes are for textbook lab 12 on page 221

Naïve Bays function, “navieBayes”, is in “e1071” package

*library(“e1071”)*

Step 1 . Bring your table with categorical attributes into R

*cust<-read.csv("nb-example.csv")*

Step 2. Create a training and test datasets by partitioning and identify target attribute

*set.seed(2020))*

*sam<- createDataPartition(cust$Purchase, p=0.7, list=FALSE)*

*train <- cust[sam,]*

*test <- cust[-sam,]*

Step 3. Build your model using training dataset

*cust.nb <-naiveBayes(Purchase ~ . , data = train)*

Step 4. Run the model with the training data to create predictions

*pred.train <- predict(cust.nb, train)*

Step 5. View the quality of the model on the training data

*table (train$Purchase, pred.train, dnn = c("Actual", "Predicted"))*

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted | |
|  |  | No | Yes |
| Actual | No | 5 | 1 |
| Yes | 0 | 6 |

Step 6. Run the model with the test data to create predictions and view the result

*pred.test <- predict(cust.nb, test)*

*table (test$Purchase, pred.test, dnn = c("Actual", "Predicted"))*

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted | |
|  |  | No | Yes |
| Actual | No | 2 | 0 |
| Yes | 2 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| Company | Prior Legal Trouble | Company Size | Status |
| 1 | Yes | Small | Truthful |
| 2 | No | Small | Truthful |
| 3 | No | Large | Truthful |
| 4 | No | Large | Truthful |
| 5 | No | Small | Truthful |
| 6 | No | Small | Truthful |
| 7 | Yes | Small | Fraudulent |
| 8 | Yes | Large | Fraudulent |
| 9 | No | Large | Fraudulent |
| 10 | Yes | Large | Fraudulent |

Total number of records 10.

Predictors: Prior legal trouble (y = Yes, n = Not) and company size (s = Small, l = Large)

For example probability of having prior legal problem and being small company and being fraudulent true

Probability of being fraudulent true

**The following is the formula for calculating Bayes**

Where and

Therefore probability of the fraudulent class given predictor Prior Legal Trouble = y company size = s is calculated as follow:

We already calculated the numerator components.

The denominators are:

, , ,

There is a practical difficulty with the Exact Bayes’ Algorithm which it is remove in Naïve Bayes. Which is the formula we had it at the start of this week notes which is:

Writing the formula for *m* classes and *d* predictors in its detail form:

And the following is the Naïve Bayes for

|  |  |  |  |
| --- | --- | --- | --- |
| Company | Prior Legal Trouble | Company Size | Status |
| 1 | Yes | Small | Truthful |
| 2 | No | Small | Truthful |
| 3 | No | Large | Truthful |
| 4 | No | Large | Truthful |
| 5 | No | Small | Truthful |
| 6 | No | Small | Truthful |
| 7 | Yes | Small | Fraudulent |
| 8 | Yes | Large | Fraudulent |
| 9 | No | Large | Fraudulent |
| 10 | Yes | Large | Fraudulent |