Probabilistic Learning – Classification using Naïve Bayes

Weather forecasts are usually provided in terms such as "70 percent chance of rain". These forecasts are known as probabilities of precipitation reports. But how are they calculated? It is an interesting question because in reality, it will either rain or it will not.

These estimates are based on probabilistic methods. Probabilistic methods are concerned about describing uncertainty. They use data on past events to extrapolate future events. In the case of weather, the chance of rain describes the proportion of prior (previous) days with similar atmospheric conditions in which precipitation occurred. So a 70% chance of rain means that out of 10 days with similar atmospheric patterns, it rained in 7 of them.

Naïve Bayes machine learning algorithm uses principles of probabilities for classification. Naïve Bayes uses data about prior events to estimate the probability of future events. For example, a common application of naïve Bayes uses frequency of words in junk email messages to identify new junk mail. We will learn:

- Basic principle of probabilities that are used for naïve Bayes.

- Specialized methods, visualizations and data structures used for analyzing text using R.

- How to employ an R implementation of naïve Bayes classifier to build an SMS message filter.

Understanding naïve Bayes:

A probability is a number between 0 and 1 that captures the chances that an event will occur given the available evidence. A probability of 0% means that the event will **not** occur, while a probability of 100% indicates that the event will certainly occur.

Classifiers based on Bayesian methods utilize training data to calculate an observed probability of each class based on feature values. When the classifier is used later on unlabeled data, it uses the observed probabilities to predict the most likely class for the new features.

Bayesian classifiers are best applied to problems in which there are numerous features and they all contribute simultaneously and in some way to the outcome. If a large number of features have relatively minor effects, taken together, their combined impact could be large.

Basic concepts of Bayesian Methods:

Bayesian methods are based on the concept that the estimated likelihood of an event should be based on the evidence at hand. **Events** are possible outcomes, such as sunny or rainy weather or spam or not spam emails. A **trial** is a single opportunity for the event to occur, such as today's weather or an email message.

Probability

The probability of an event can be estimated from observed data by dividing the number of trials in which an event occurred by the total number of trials. For example if it rained 3 out of 10 days, the probability of rain can be estimated to 30%. Similarly if 10 out of 50 emails are spam, then the probability of spam can be estimated as 20%. The notation *P(A)* is used to denote the probability of event *A*, as in *P(spam)=0.20*

The total probability of all possible outcomes of a trial must always be 100%. Thus, if the trial has only 2 outcomes that cannot occur simultaneously, such as rain or shine, spam or not spam, then knowing the probability of either outcome reveals the probability of the other.

When two events are **mutually exclusive** and **exhaustive (** they cannot occur at the same time and are the only two possible outcomes) and $P(A)= q$, then $P(\neg A)= 1-q$.

Joint Probability:

We may be interested in monitoring several non-mutually exclusive events for the same trial. If the events occur with the event of interest, we may be able to use them to make predictions. Consider, for instance, a second event based on the outcome that the email message contains the word Viagra. For most people, this word is only likely to show up in a spam message; its presence would be strong evidence that the email is spam. The probability that an email contains the word "Viagra" is 5%.

We know that 20% of all messages were spam , and 5% of all messages contained Viagra. We need to quantify the degree of overlap between the two proportions, that is we hope to estimate the probability of both Spam and Viagra occurring , which can be written as $P(spam \cap Viagra)$.

Calculating $P(spam \cap Viagra)$ depends on the **joint probability** of the two events. If the two events are totally unrelated, they are called **independent events**. On the other hand, **dependent events** are the basis of predictive modeling. For instance, the presence of clouds is likely to be predictive of a rainy day.

If we assume that $P(spam)$ and $P(Viagra)$ are independent, we could then calculate

$P(spam \cap Viagra)$ as the product of probabilities of each

$P(spam \cap Viagra)= P(spam)*P(Viagra)$ = .2*.05= 0.01, or 1% of all spam messages contain the word Viagra.

In reality, it is more likely that $P(spam)$ and $P(Viagra)$ are highly dependent, which means that the above calculation is **incorrect.**

Conditional probability with Bayes' theorem:

The relationship between dependent events can be described using **Bayes' theorem.** The notation *P(A|B)* is read as the probability of event A given that event B has occurred. This is known as **conditional probability,** since the probability of A is dependent (that is, conditional) on what happened with event B.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(A \cap B)}{P(B)}$$

To understand a little better how the Bayes' theorem works, suppose we are tasked with guessing the probability that an incoming email was spam. without any additional evidence, the most reasonable guess would be the probability that any prior message was spam (20%). This estimate is known as the **prior probability.**

Now suppose that we obtained an additional piece of evidence, that is that the incoming message the term Viagra was used. The probability that the world Viagra was used in previous spam messages is called the **likelihood** and the probability that Viagra appeared in any message at all is known as **marginal likelihood.**

By applying Bayes' theorem to this evidence, we can compute a **posterior probability** that measures how likely the message is to be spam. If the posterior probability is more than 50%, the message is more likely to be spam.

$$P(spam/Viagra) = \frac{P(Viagra/spam) * P(spam)}{P(Viagra)}$$

To calculate the components of Bayes's theorem, we must construct a **frequency table** that records the number of times Viagra appeared in spam and non-spam messages. The cells indicate the number of instances having a particular combination of class value and feature value.

	Viagra		
Frequency	Yes	No	Total
spam	4	16	20
non spam	1	79	80
Total	5	95	100

The frequency table is used to construct the **likelihood table:**

	Viagra		
Frequency	Yes	No	Total
spam	4/20	16/20	20
non spam	1/80	79/80	80
Total	5/100	95/100	100

The likelihood table reveals that *P(Viagra/spam)=4/20=.20.* This indicates that probability is 20% that a spam email contains the term Viagra. Additionally, since the theorem says that

*P(B|A)*P(A)= P(A∩B),* we can calculate *P(spam∩ Viagra) as P(Viagra|spam)*P(spam)= (4/20) *(20/100)=0.04.*

This is 4 times the probability under independence.

To compute the posterior probability *P(spam|Viagra),* we take

*P(Viagra|spam)*P(spam)/P(Viagra)=(4/20)*(20/100)/(5/100)= .80.*

Therefore, the probability is 80% that a message is spam, given that it contains the word Viagra.

This is how commercial spam filters work in general, although they consider a much larger number of words simultaneously, when computing the frequency and likelihood tables.

The naïve Bayes algorithm

The **naïve Bayes(NB)** algorithm describes a simple application using Bayes' theorem for classification. It is the most common algorithm, particularly for text classification where it has become the standard. Strengths and weaknesses of this algorithm are as follows:

Strengths	Weaknesses
• Simple, fast and very effective. • Does well with missing or noisy data • Requires relatively few examples for training, but works well with very large numbers of examples. • Easy to obtain the estimated probability for a prediction	• Assumes that all features are equally important and independent • Not ideal for datasets with large numbers of numeric features • Estimated probabilities are less reliable than the predicted classes

The naïve Bayes algorithm is named as such because it makes a couple of "naïve" assumptions about the data. In particular, NB assumes that all of the features in the dataset are equally important and independent. These assumptions are often not true.

The naïve Bayes classification

Let's extend our spam filter by adding a few additional terms to be monitored: money, groceries and unsubscribe. The NB learner is

trained by constructing a likelihood table for the appearance of these four words (W1, W2, W3 and W4) as in the following:

	Viagra (W1)		Money (W2)		Groceries (W3)		Unsubscribe (W4)		
Likelihood	Yes	No	Yes	No	Yes	No	Yes	No	Total
spam	4/20	16/20	10/20	10/20	0/20	20/20	12/20	8/20	20
not spam	1/80	79/80	14/80	66/80	8/80	71/80	23/80	57/80	80
Total	5%	95%	24%	76%	8%	91%	35%	65%	100

As new messages arrive, the posterior probability must be calculated to determine whether they are more likely spam or not spam, given the likelihood of the words found in the message. For example, suppose that a message contains the terms Viagra and Unsubscribe, but does not contain either Money or Groceries.

Using Bayes theorem we can define the problem as shown in the following formula which captures that a message is spam, given that Viagra=yes, Money= No, Groceries= No and Unsubscribe=yes :

$$P(Spam|W1 \cap \neg W2 \cap \neg W3 \cap W4) = \frac{P(W1 \cap \neg W2 \cap \neg W3 \cap W4|Spam)P(Spam)}{P(W1 \cap \neg W2 \cap \neg W3 \cap W4)}$$

This formula is computationally difficult to solve. As additional features are added, large amounts of memory are needed to store the probabilities of all possible intersection events.

However, this becomes easier with the assumption that events are independent. Specifically, NB assumes that events are independent so long as they are related to the same class values. Assuming conditional independence allows us to simplify the formula using the probability rule for independent events $P(A \cap B) = P(A) * P(B)$.

So our formula becomes:

$$P(Spam|W1 \cap \neg W2 \cap \neg W3 \cap W4)$$
$$= \frac{P(W1|Spam)P(\neg W2|Spam)P(\neg W3|Spam)P(W4|Spam)P(Spam)}{P(W1)P(\neg W2)P(\neg W3)P(W4)}$$

The result of this formula is then compared to the probability that the message is not spam:

$$(NoSpam|W1 \cap \neg W2 \cap \neg W3 \cap W4)$$
$$= \frac{P(W1|NotSpam)P(\neg W2|NotSpam)P(\neg W3|NotSpam)P(W4|NotSpam)P(NotSp}{P(W1)P(\neg W2)P(\neg W3)P(W4)}$$

Using the values in the likelihood table, we can start filling the numbers in these equations. Because the denominator is the same, we will ignore it for now.

The overall likelihood of spam is then

(4/20)*(10/20)*(20/20)*(12/20)*(20/100)=0.012.

While the likelihood of non spam given this pattern of words is:

(1/80)*(66/80)*(71/80)*(23/80)*(80/100)=0.002

Since 0.012/0.002= 6, this says that an email with this pattern of words is 6 times more likely to be spam than non spam.

To convert these numbers to probabilities, we apply the formula= 0.012/(0.012+0.002)= 0.857=85.7%

The probability that the message is spam is equal to the likelihood that the message is spam divided by the sum of likelihoods that the message is either spam or non spam.

Similarly, the probability of non spam is : 0.002/(0.012+0.002)=0.143

Given the pattern of words in the message, we expect that the message is spam with 85.7% probability and non-spam with 14.3% probability.

The naïve Bayes classification algorithm used can be summarized by the following formula. The probability of level L for class C, given the evidence provided by features F1, F2,…, Fn, is equal to the product of probabilities of each piece of evidence conditioned on the class level, the prior probabilities of the class level and a scaling factor 1/Z which converts the result to a probability:

$$P(C_L|F_1, F_2, \dots, F_n) = \frac{1}{Z} p(C_L) \prod_{i=1}^{n} p(F_i|C_L)$$

The Laplace Estimator:

Let us look at one more example. Suppose we received another message, this time containing the terms: Viagra, Groceries, Money and Unsubscribe. Using the naïve Bayes algorithm, as before, we can compute the likelihood of spam as:

(4/20)*(10/20)*(0/20)*(12/20)*(20/100)=0

And the likelihood of non-spam as:

(1/80)*(14/80)*(8/80)*(23/80)*(80/100)=0.00005

Therefore the probability of spam = 0/(0+0.00005)=0

And the probability of non spam = 1

This result suggests that the message is spam with 0% probability and non spam with 100% probability. This prediction probably does not make sense, since it includes words that are very rarely used in legitimate messages. It is therefore likely that the classification is not correct.

This problem might arise if an event never occurs for one or more levels of the class in the training set. For example, the term Groceries had never previously appeared in a spam message. Consequently *P(spam|groceries)=0*

Because probabilities in NB are multiplies, this 0 value causes the posterior probability of spam to be 0, giving a word the ability to nullify and overrule all of the other evidence.

A solution to this problem involves using the **Laplace estimator.** The Laplace estimator adds a small number to each of the counts in the frequency table, which ensures that each feature has a non zero probability

of occurring with each class. Typically, the estimator is set to 1, which ensures that every feature has a non-zero probability.

Let us see how this affects our prediction for this message. Using a Laplace value of 1, we add 1 to each numerator in the likelihood function. The total number of 1's must also be added to each denominator. The likelihood of spam becomes:

(4/20)*(10/20)*(0/20)*(12/20)*(20/100)=0

(5/24)*(11/24)*(1/24)*(13/24)*(20/100)= 0.0004

And the likelihood of non spam is:

(2/84)*(15/84)*(9/84)*(24/84)*(80/100)=0.0001

Probability of spam = 0.0004/0.0005=.8= 80%

Probability of non spam = 20%

Using numeric features with naïve Bayes

Because naïve Bayes uses frequency tables for learning the data, each feature must be categorical in order to create the combinations of class and feature values. Since numeric features do not have categories of values, the NB algorithm would not work without modification.

One easy and effective solution is to discretize a numeric feature, which means that the numbers are put in categories known as **bins.** For this reason, discretization is often called **binning.**

There are several different ways of binning a numeric value. The most common is to explore the data for natural categories or **cut points** in the distribution. For example, suppose you added a feature to the spam dataset that recorded the time (on a 24 hours clock) the email was sent.

We might want to divide the day into 4 bins of 6 hours based on the fact that in the early hours of morning, messages frequency is low. Activity picks up during business hours, and tapers off in the evening. This seems to validate the choice of 4 natural bins of activity. Each email will have a feature stating which bin the email belongs to.

Note if there are no obvious cut points, one option is to discretize the feature using quantiles.

Practice exercises on Naïve Bayes

Exercise 1:

Using the data above find the probability that an email is spam or not if it has:

- “groceries”, “Money” and “unsubscribe” and not containing “Viagra”

-“Viagra” and “groceries”, but not “money” or “unsubscribe”

Exercise 2:

A retail store carries a product that is supplied by three manufactures, A, B, and C, and 30% from A, 20% from B and 50% from C.

It is known that 2% of the products from A are defective, 3% from B are defective, and 5% from C are defective.

A) If a product is randomly selected from this store, what is the probability that it is defective?

B) If a defective product is found what is the probability that it was from B?