Machine Learning Assignment n.1

NAIVE BAYES CLASSIFIER

October 21, 2020

Chiara Terrile ID: 4337786
Cristina Naso Rappis ID: 4378585
University of Genoa

Contents

Chapter 1

Introduction

1.1 Defining the objective

The objective of this assignment is to implement a Naive Bayes Classifier using a Training Set and classify some patterns contained in a Test Set.
Given the data set which contains informations about the weather, the classifier should be able to estimate whether it is possible to play tennis or not for a specific test set.
After implementing this classifier, we have modified it using the Laplace Smoothing for computing probabilities.

1.2 Naive Bayes Classifier

A classifier is a machine learning model that is used to discriminate different objects based on certain features.
Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Bayes Theorem :

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

Using Bayes theorem, we can find the probability of A happening, given that B has occurred. Here, B is the evidence and A is the hypothesis. The assumption made here is that the features are independent,therefore the presence of one particular feature does not affect the other. Hence it is called naive.
In our case A was the probability of "playing tennis" that we can indicate with y. B is the independent feature vector (contained in the test set) composed by n features (4 in our case) for the weather conditions that verifies in a precise day. We indicate it as X.
According to this we can express the Bayes Theorem in the following way (that will be

the one used in the implementation) :

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

Where $X = (x_1, x_2, ..., x_n)$
Now, we can consider a naive assumption to the Bayes' theorem, which is, independence among the features. This simplify a lot the calculations, since the formula becomes the following:

$$P(y|x_1, x_2, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

1.3 Laplace (additive) Smoothing

Since we are considering a small data set, some combinations that appear in the test set can not be present in the training set. Therefore their probability is assumed to be zero. To deal with this case, we introduce a sort of prior information, which is called additive smoothing. This prior belief is that all values are equally probable. The only requirement for this assumption is that we must know in advance the number of possible levels(as in our case). According to this we compute the probabilities using the following formula:

$$P(x = i) = \frac{n_i + a}{N + av} \tag{1.1}$$

where:

- n_i is the number of times that the event i occurs
- N is the number of experiments
- a is the smoothing parameter
- v is the number of levels for attribute x

Chapter 2

Implementation

2.1 Task1

In this first part we have processed the data received. First of all we have encoded the string values into numerical ones in the following way:

- the input feature **outlook** : [*overcast,rainy, sunny*] has been substituted with [1 2 3]
- the input feature **temperature** : [*hot,cool, mild*] has been substituted with [1 2 3]
- the input feature **humidity** [*high medium*] has been substituted with [1 2]
- the input feature **windy** [*FALSE TRUE*] has been substituted with [1 2]
- the input feature **play** [*Yes No*] has been substituted with [1 2]

Once imported the pre-processed data set, we have created two submatrices in a random way, one for the Training Set (composed of 10 rows) and one for the Test Set (composed of 4 rows).

2.2 Task2

In this part we have implemented a function **NaiveBayesClassifier** which takes as input the *TrainingSet* and the *TestSet* matrices. First of all we have considered the TrainingSet and computed the probability $P(x_i|yes)$ for each x_i.
Where x_i is an observation belonging to an input feature, for example *rainy* conditioned to the target *yes* (so the probability that it will rain knowing that we play tennis).
We also computed the probability P(yes) for the whole Training Set (probability of playing tennis *P_play_yes*), that is obtained dividing the number of times we have *Yes* in the target column by the number of observations.
For the classification we have also computed the number of times that each value occurs (for example the number of times that *Rainy* appears in the Training Set). This number

will be divided by the number of observations in order to get the probability of that particular event.
Once computed all these probabilities, we are able to classify the Test Set according to the inferred rule and to apply the Bayes Theorem.
To do this, for each instance of the Test Set we have computed $P(Yes|X)$. For example, if the current instance is X = [*Sunny Hot Normal False*], the probability will be the following :

$$P(Yes|X) = \frac{P(SunnyOutlook|Yes)P(HotTemperature|Yes)P(NormalHumidity|Yes)P(NoWind|Yes)P(Yes)}{P(SunnyOutlook)P(HotTemperature)P(NormalHumidity)P(NoWind)}$$

If the computed probability is > 0.5, this means that the obtained classification for that particular instance of the Test Set is *we can play tennis*. Otherwise (if $P < 0.5$) the conclusion is the opposite. Since we have a target column in the Test Set, we can compare the result of each classification with the corresponding value for that instance contained in the target column. This allows us to compute the error rate such as (number of errors / m).

2.3 Task3

As we have already introduced in section 1.3, in order to use the Laplace smoothing we must know in advanced the number of levels for each feature.
In our case these values were respectively: 3 for outlook, 3 for temperature, 2 for humidity and 2 for windy. We added a line containing these 4 values both to the Training Set and to the Test Set matrices.
Later, we have just repeated the same steps as in Task2 using different formulas for calculating the probabilities, which take into account the number of levels v and the smoothing parameter a,that we considered equal to 1. (See formula 1.1)

Chapter 3

Results

Running the main code in MATLAB, we have obtained different results, depending on the matrices TrainingSet and TestSet which are ordered in a random way.
Below are reported the results of the three most significative cases.

Case 1: bigger error rate with Laplacian Smoothing

NaiveBayesClassifier

The classification for pattern 1 of the Test Set is: YOU CAN PLAY TENNIS
The classification for pattern 2 of the Test Set is: YOU CAN PLAY TENNIS
The classification for pattern 3 of the Test Set is: YOU CAN NOT PLAY TENNIS
The classification for pattern 4 of the Test Set is: YOU CAN PLAY TENNIS
The target for pattern 1 of the Test Set is: NO
The target for pattern 2 of the Test Set is: YES
The target for pattern 3 of the Test Set is: NO
The target for pattern 4 of the Test Set is: YES
The error rate is 0.250000

NaiveBayesClassifier using Laplace smoothing

The classification for pattern 1 of the Test Set is: YOU CAN PLAY TENNIS
The classification for pattern 2 of the Test Set is: YOU CAN PLAY TENNIS
The classification for pattern 3 of the Test Set is: YOU CAN PLAY TENNIS
The classification for pattern 4 of the Test Set is: YOU CAN PLAY TENNIS
The target for pattern 2 of the Test Set is: NO
The target for pattern 3 of the Test Set is: YES
The target for pattern 4 of the Test Set is: NO
The target for pattern 5 of the Test Set is: YES
The error rate is 0.500000

Case 2: same error rate

NaiveBayesClassifier

The classification for pattern 1 of the Test Set is: YOU CAN NOT PLAY TENNIS
The classification for pattern 2 of the Test Set is: YOU CAN NOT PLAY TENNIS
The classification for pattern 3 of the Test Set is: YOU CAN NOT PLAY TENNIS
The classification for pattern 4 of the Test Set is: YOU CAN NOT PLAY TENNIS
The target for pattern 1 of the Test Set is: YES
The target for pattern 2 of the Test Set is: NO
The target for pattern 3 of the Test Set is: YES
The target for pattern 4 of the Test Set is: YES
The error rate is 0.750000

NaiveBayesClassifier using Laplace smoothing

The classification for pattern 1 of the Test Set is: YOU CAN NOT PLAY TENNIS
The classification for pattern 2 of the Test Set is: YOU CAN NOT PLAY TENNIS
The classification for pattern 3 of the Test Set is: YOU CAN NOT PLAY TENNIS
The classification for pattern 4 of the Test Set is: YOU CAN NOT PLAY TENNIS
The target for pattern 2 of the Test Set is: YES
The target for pattern 3 of the Test Set is: NO
The target for pattern 4 of the Test Set is: YES
The target for pattern 5 of the Test Set is: YES
The error rate is 0.750000

Case 3: smaller error rate with Laplacian Smoothing

NaiveBayesClassifier

The classification for pattern 1 of the Test Set is: YOU CAN NOT PLAY TENNIS
The classification for pattern 2 of the Test Set is: YOU CAN NOT PLAY TENNIS
The classification for pattern 3 of the Test Set is: YOU CAN PLAY TENNIS
The classification for pattern 4 of the Test Set is: YOU CAN NOT PLAY TENNIS
The target for pattern 1 of the Test Set is: YES
The target for pattern 2 of the Test Set is: YES
The target for pattern 3 of the Test Set is: YES
The target for pattern 4 of the Test Set is: YES
The error rate is 0.750000

NaiveBayesClassifier using Laplace smoothing

The classification for pattern 1 of the Test Set is: YOU CAN PLAY TENNIS
The classification for pattern 2 of the Test Set is: YOU CAN PLAY TENNIS
The classification for pattern 3 of the Test Set is: YOU CAN PLAY TENNIS
The classification for pattern 4 of the Test Set is: YOU CAN PLAY TENNIS
The target for pattern 2 of the Test Set is: YES
The target for pattern 3 of the Test Set is: YES
The target for pattern 4 of the Test Set is: YES
The target for pattern 5 of the Test Set is: YES
The error rate is 0.000000

3.1 Conclusions

As we can see in the above experiments, we don't always achieve a better classification using the Laplace classifier. This is because we consider a small number of observations in the data set. Therefore, the classification we obtain only depends on how the observations are divided into Data Set and Test Set, that as we said happens in a random way.
As we have already said before, the Laplace smoothing introduce the prior belief that all value are equally probable.
If we consider the **Case 3** (which results have been shown above) and we report the corresponding content of the Training Set matrix (see Table 3.1), we can understand why the Laplace classifier gives us a better classification.
For example if we look at the first column of the matrix, which correspond to the outlook feature, we can see that the value *overcast* only appears in correspondence of *yes* (we can play tennis)in the target column.
This means that if we compute the probability (without the smoothing) that it's overcast knowing that we can not play tennis, it will result 0. However this is not true, since it's

only a consequence of the observations that have been randomly assigned to the Training Set.
We can see that the same situation also verifies for other values in the features. The Laplace classifier just modifies the criteria to compute probabilities and it ensures that, even if we happen to be in the case above, none of them will be 0.
On the other hand, there might also be cases in which the Laplace smoothing only introduce an error, which is the reason why sometimes its classification end up being less accurate than the one from the classical classifier(if you look at the error rate).
As a final conclusion, we can say that it's reasonable to use the Laplace smoothing just in case we have a limited number of observations, since its classification will be (NOT always) more accurate.

n.	Outlook	Temperature	Humidity	Windy	Play
1	rainy	cool	normal	TRUE	no
2	sunny	mild	normal	TRUE	yes
3	rainy	mild	high	FALSE	yes
4	sunny	hot	high	TRUE	no
5	**overcast**	hot	normal	FALSE	**yes**
6	rainy	mild	high	TRUE	no
7	**overcast**	mild	high	TRUE	**yes**
8	rainy	mild	normal	FALSE	yes
9	sunny	mild	high	FALSE	no
10	sunny	hot	high	FALSE	no

Table 3.1