Multinomial Naive Bayes Classifier

Miguel Alcón Doganoc
Universitat Politècnica de Catalunya
Barcelona, Spain
miguel.alcon@est.fib.upc.edu

1 INTRODUCTION

The Multinomial Naive Bayes classifier (MNB) is part of the family of the Naive Bayes classifiers, which are simple probabilistic classifiers based on applying Bayes's theorem with strong (naive) independence assumptions between the features.

The main characteristic of MNB is that it works with transactional data. This means that each observation of the data is seen as a data structure itself, so it can contain different number of features (items) in each of them.

The goal of MNB is: given a set of observations of different classes and sizes, each of them formed by several items, create a model (conditional probabilities of the instances given the class, and the probabilities of the classes themselves) in order to predict the class of other observations.

2 ALGORITHM

Now is time to explain the algorithm behind the MNB. For giving a more practical view, we divided it in two processes, ***fit*** and ***predict***. In the ***fit*** stage, the classifier model is build from the input data, while in the ***predict*** stage, MNBC uses the model to predict an input observation.

2.1 Algorithm of *fit*

Given input X
- Let C be the number of classes in X
- Let c_k be a class of observation,
  - with $1 \geq k \geq C$
- Let fc_k be the frequency of class c_k within X
- Let T be the number of unique items in X
- Let t_i be an item of the observations in X,
  - with $1 \geq i \geq T$
- Let ft_i be the frequency of item t_i within X
- Let $f_{i,k}$ be the frequency of item t_i
  - within observations of class c_k
- For each c_k classes in X
  - $Pr(c_k) = \frac{fc_k}{\sum_j^C fc_j}$
  - For each t_i items in X
    - $Pr(t_i \mid c_k) = \frac{f_{i,k}+1}{\sum_j^T f_{j,k}+T}$

The algorithm, described above, makes the necessary counting of the classes c_k and items t_i of the input, in order to compute the probabilities $Pr(c_k)$ and $Pr(t_i \mid c_k)$ of each item and class. In the case of $Pr(t_i \mid c_k)$, we applied the Laplace Correction to make the algorithm behave correctly.

All these probabilities together with the information of the classes form what we call a *model* of the input, which is used in the ***predict*** stage.

2.2 Algorithm of *predict*

The algorithm of ***predict*** computes $L(c_k)$ for every possible class c_k in order to predict which of them is the class of the observation, with the highest probability (highest $L(c_k)$). This function, $L(c_k)$, comes from

$$Pr(O \mid c_k) = Pr(c_k) \cdot \prod_i^T Pr(t_i \mid c_k)$$

But instead of the multiplication of probabilities, we used the summation of the logarithms of the probabilities. This decision was made to avoid the high risk of underflow that the multiplication of small numbers has.

Given an observation O
- Let T be the number of items in O
- Let t_i be an item of O
  - with $1 \geq i \geq$ T
- For each c_k possible class
  - $L(c_k) = \log Pr(c_k) + \sum_i^T \log Pr(t_i \mid c_k)$
- Output the class $argmax_k L(c_k)$

3 IMPLEMENTATION

In this project, we developed a MNB classifier in *Python 3*, following the algorithm described in section 2. As we mentioned before, the algorithm is divided in two functions. The ***fit*** function needs the input data (an array of observations) and a set of classes. Each observation must contain one and only one class within it. If the set of classes is not passed as an input to the function, it will assume that the class of the observation is in its first position. In the case of ***predict***, it just needs an array of occurrences that it has to predict. It outputs an array of the predicted classes corresponding to each of the occurrences.

4 DATA

After we have implemented the algorithm, we have to test its correctness and behavior. To do the first, we applied it with the *toy* example of [1] as the input, and we compared the result of our classifier with the expected one. To test its behavior in a more realistic case, we also applied our MNB to the *markbask* data set. It is composed by a list of observations, each of them composed by supermarket products, the gender of the purchaser and the information of whether he/she owns a house or not (which are two possible classes of the observation).

5 EXPERIMENTATION

5.1 Correctness

We tested our algorithm with the *toy* example to check if it predicts the class of the 'a a a e f' observation as expected, and it does it. First, we tried it with the sum of logarithms of the probabilities, and it predicts the class 1, same as in the example. To ensure that it was not coincidence, we also tried it with the multiplication of the probabilities. Our MNB gave the exactly the same probabilities than in the example, as it is shown below.

$$Pr(a|1) = 0.42857 \qquad Pr(a|2) = 0.22222$$
$$Pr(b|1) = 0.14286 \qquad Pr(b|2) = 0.11111$$
$$Pr(c|1) = 0.14286 \qquad Pr(c|2) = 0.11111$$
$$Pr(d|1) = 0.14286 \qquad Pr(d|2) = 0.11111$$
$$Pr(e|1) = 0.07143 \qquad Pr(e|2) = 0.22222$$
$$Pr(f|1) = 0.07143 \qquad Pr(f|2) = 0.22222$$

$$Pr(a\ a\ f\ a\ e|1) = 0.00030$$
$$Pr(a\ a\ f\ a\ e|2) = 0.00014$$

5.2 Behavior in a realistic case

In this experiment, we used our classifier to predict the class of some instances of the *markbask* data. First we tried predicting the gender and then whether the purchaser owns a house or not (ownership). As usual, we started splitting (randomly) the input data into train and test sets, for ***fit*** and ***predict***, respectively. For the gender classification, we obtained an accuracy of 0.63, while in the ownership one we obtained 0.62.

For deeper information about the predictions of all the experiments, you can take a look to the 'experiments.out' file, or you can just execute the *Python* script.

6 CONCLUSIONS

We have achieved the objectives that we had for this project: understand MNB, implement it successfully, and test it. As a result of the experimentation, we can conclude that the accuracy of the classifier is not so much high, but at least is greater than $\frac{1}{2}$. This means that, even predicting the gender of a person or house ownership by a small set of products that he/she buys in the supermarket, which is hard, the classifier can recognize some patterns in the data, and predict most of the times the correct class.

As future work, we could make a further testing of our implementation of MNB, with bigger and more realistic (complex) data sets.

REFERENCES

[1] Department of Computer Science. Algorithmics for data mining (pg 155-158). http://www.cs.upc.edu/~balqui/slidesADM2017.pdf, 2017.