TP 1 Data Mining: Naive Bayes (OBLIGATORY)

Tuesday 16th March, 2021
deadline: Monday 29th March, 2021, 23:59

This TP is obligatory. In this TP you are going to implement the Naive Bayes (NB) algorithm for categorical (titanic) and continuous (iris) data using **Python 3.**

You are going to fill a few missing functions in the python script nb.py to implement the exercises that we ask. So first of all read and understand the given python scripts To run your code you have to run the main TP1_NB.ipynb notebook. Here you have to write only a short code (it is mentioned where) to run the NB algorithm. Parts of the code are given and it works if the missing functions in nb.py are correctly implemented.

Theory Reminder on Naive Bayes

Remember that Naive Bayes works on computing the *Maximum A Posteriori Probability*, based on the assumption that the attributes of the training examples are independent.

$$
\begin{aligned}
C_{MAP} &= argmax_{c_i \in C} P(c_i|X) && (1)\\
&= argmax_{c_i \in C} \frac{P(X|c_i)P(c_i)}{P(X)} && (2)\\
&= argmax_{c_i \in C} \frac{P(X|c_i)P(c_i)}{\sum_{c_j \in C} P(X|c_j)P(c_j)} && (3)\\
&\propto argmax_{c_i \in C} P(X|c_i)P(c_i) && (4)\\
&= argmax_{c_i \in C} P(X_1, X_2, X_3, ..., X_d|c_i)P(c_i) && (5)
\end{aligned}
$$

Under the assumption of independent attributes we have :

$$
\begin{aligned}
C_{MAP} &= argmax_{c_i \in C} P(X_1, X_2, X_3, ..., X_d|c_i)P(c_i) && (6)\\
&= argmax_{c_i \in C} P(X_1|c_i)P(X_2|c_i)...P(X_d|c_i)P(c_i) && (7)\\
&= argmax_{c_i \in C} P(c_i)\Pi_{j=1}^{d} P(X_j|c_i) && (8)
\end{aligned}
$$

How to classify a new instance
In order to classify a new instance we will be using the formula 7. For example consider the following instance :

$$< a_1, a_2, a_3, ..., a_d, ? >$$

Formula 7 will be then written as

$$? = C_{MAP} = argmax_{c_i \in C} P(X_1 = a_1|c_i)P(X_2 = a_2|c_i)...P(X_d = a_d|c_i)P(c_i)$$

In practice, to find the class label that maximizes the above quantity we will have to compute:

$$P(X_1 = a_1|c_i)P(X_2 = a_2|c_i)...P(X_d = a_d|c_i)P(c_i)$$

for every class i, and select the one that gives the highest probability.

Computing the necessary probabilities to perform classification
Before we are able to classify a new instance we have to compute the following:

1. The probability $P(c_i)$ for every class

2. For each attribute $X_j, 1 \leq j \leq n$, with k distinct values : $a_{j1}, a_{j2}, ..., a_{jn}$, compute for every distinct value $a_{jz},\ 1 \leq z \leq k$ the conditional probability:
$$P(X_j = a_{jz}|c_i)$$

Exercises

1. Fill the missing parts of the train_nb(), normal_distribution() and predict() functions in nb.py to implement the the NB algorithm for continuous data.
   (a) Run your algorithm using the iris data set and compute the train and classification accuracy. **Comment your results.**
   (b) Draw the decision surfaces (code is given). For the visualization purposes we chose two attributes as predictive attributes and color the plane defined by these two attributes on the basis of class labels that Naive Bayes predicts. **Comment discuss the result.**
2. Implement Naive Bayes for categorical data,
   (a) Make the necessary modifications in the train_nb() and predict() functions in nb.py in order to be able to use these functions for both continuous and categorical data.
   (b) apply titanic dataset and report your misclassification rate. Discuss

General instruction

You have to submit your **code** and a **report**, both saved using the format: TP1_LASTNAME_Firstname.
If you prefer you can use the notebook for your report. In your report explain the problem and discuss the results, it should be a summary of the main findings, clear, and concise. Include in your report the visualization on iris and your comments on the decision surfaces (what form do they have? linear? quadratic? other? It should also contain the misclassification percentage/ accuracy and a discussion of performance.

The code should be well written with detailed comments to explain what you do at each step. Try to avoid the for loops using the nymPy library.