Assignment 2

Computer Science Department, University of Crete

MACHINE LEARNING - CS 577, Fall 2024

Assignment 2

Deadline: Wednesday, 23/10/2024, 23:59 on e-learn (`https://elearn.uoc.gr`).

Deliverable files: Submit a zip file containing a report in PDF with the answers **AND all** Python files (.py) written by you in the scope of the assignment. The final grade will be the result of the quality of your submitted results in your report, together with the correctness of your submitted code.

Python Version: Ideally, use python **3.6**. If you have issues with this version of Python feel free to use a more recent version.

Exercise 1 - Probabilities (Theoretical) [20 points]

Let the random variable X follow the distribution:

$$f(x;\theta) = \theta^2(x+1)(1-\theta)^x \quad , x = 0, 1, 2.., \theta \in [0, 1]$$

a. **[5 points]** Find the expression describing the MLE estimators for θ for N independent identically distributed (i.i.d.) samples. How can you be sure that this value you found is indeed the maximum?

b. **[5 points]** Calculate θ for f(x;θ) using the formula calculated in the first step, and applying it to the following 15 samples:

$$[3.2, 1.4, 2.2, 7, 0.5, 3.3, 9, 0.15, 2, 3.21, 6.13, 5.5, 1.8, 1.2, 11]$$

Exercise 2 - Naïve Bayes (Theoretical) [15 points]

Consider Table 1, presenting a dataset with 7 samples, each comprised of three Boolean variables x, y and z, and a Boolean target variable U. You will use this data to train a Naïve Bayes classifier and predict U. Specifically:

Table 1:

x	y	z	U
1	1	1	0
0	1	1	0
0	0	1	0
1	0	0	1
0	0	1	1
0	1	0	1
1	1	0	1

a. [**5 points**] After learning is complete, what would be the predicted probability $P(U = 0|x = 0, y = 0, z = 1)$?

b. [**5 points**] Why — in this case — did we not need to exploit the Laplace trick to solve for question *a*?

***Hint**: Try to solve for P(U = 0|x = 0, y = 0, z = 0) and check which element in the Naïve Bayes classifier formula causes the method to fail.*

c. [**5 points**] Using the probabilities obtained during the Bayes Classifier training, what would be the predicted probability $P(U = 1|x = 0)$?

Exercise 3 - Naïve Bayes Classifier (Programming) [65 points]

***NOTE**: You are* **NOT** *allowed to use any existing implementations of the Naive Bayes Classifier.*

You will have to implement the Naïve Bayes Classifier (NBC) by coding the formulas you saw in class, recitations and the reading material. The classifier should be able to handle either categorical (i.e. discrete values) *or* continuous variables. The implementation of the classifier will be split in 2 functions, one for training and one for predicting.

a. [**25 points**] Implement the NBC training function:

def train_NBC(X, X_dtype, Y, L, D_categorical)

Inputs:

- X : IxM matrix of variables. Rows correspond to the I samples and columns to the M variables.
- X_dtype : String describing the data type of X, which could be either "categorical" or "continuous".
- Y : Ix1 vector. Y is the class variable you want to predict.
- L : Scalar. L is the parameter referred to in the MAP estimates equation. For L = 0 you get the MLE estimates. $L \geq 0$.

- D_categorical: 1xM vector. Each element D(m) contains the number of possible different values that the categorical variable m can have. This vector is ignored if X_dtype = "continuous".

Output:

- Model: This model should contain all the parameters required by the NBC to classify new samples. It is up to you to decide its structure. The only requirement is that it is compatible with your next function.

Notes − Categorical values:

- All categorical variables take values starting from 0. If a variable can take K possible values, its values are in [0, K-1]. This holds for both the class values in Y and the values in X.
- If some combinations of values do not occur in the data they take probability 0, unless L is greater than 0.
- D_categorical − It is important to pass this information to the function because the samples used for training do not necessarily contain all possible values that the variables can take.

b. [**20 points**] Implement the NBC prediction function:

def predict_NBC(model, X, X_dtype)

Inputs:

- model: The model previously trained using **train_NBC**.
- X: JxM matrix of variables. Rows correspond to the J samples and columns to the M variables.
- X_dtype : String describing the data type of X, which could be either "categorical" or "continuous".

Output:

- predictions: Jx1 vector. Contains the predicted class for each input samples.

c. [**20 points**] Assess the classifier using the datasets uploaded along with the assignment.

The analysis shall consist of the following steps:

– **Randomly** split each dataset into two parts, one containing 75% of the samples (training set) and one containing the remaining 25% (test set).

– Train the classifier on the training set. Train the algorithm for the categorical/continuous data on the corresponding datasets.

- Perform predictions on the test datasets and assess the model's accuracy, i.e. **the percentage of correctly classified samples.**
- Repeat the procedure 100 times, each time randomly re-splitting the dataset into 75% train and 25% test sets. Compute the average accuracy of the algorithm.

Note: The content of the attached csv files is the following:

a) Dataset*_X_categorical: categorical variable data

b) Dataset*_X_continuous: continuous variable data

c) Dataset*_Y: class labels for the corresponding dataset

d) Dataset*_D_categorical: number of possible values that a feature might have

d. *Bonus question:* [**10 points**] How does the choice of the hyperparameter L affect the results, in the case of the categorical classification? Experiment with small and large values of L.