CZ4041/CE4041: Machine Learning Programming Assignment

Task: To implement the non-parametric density estimation method, Naïve Estimator (Pages 26-32 of "Lecture 10b: Density Estimation", to better understand this method, you may need to read through Pages 20-25 as well), in Python.

Submission: To submit a single Python source file named "naiveEst.py" via NTULearn by 11:59pm on 28 Apr. 2020.

Details: The functionality of the file "naiveEst.py" is to first read a dataset from an input file named "data.txt", and then apply the Naive Estimator method to estimate the probability density of each data instance, and finally write the results in an output file named "output.txt".

The format of "data.txt" is shown in Table 1. In "data.txt", the first line shows the meta information of the dataset, where n is the number of data instances, and m is the number of features or dimensions of each data instance (n and m are separated by a comma). Starting from the 2nd row to the $(n+1)$-th row, each row represents a data instance of m features. The m feature values of each instance are separated by space. For example, X_{11} denotes the value of the 1st feature of the 1st instance, and X_{22} denotes the value of the 2nd feature of the 2nd instance, etc.

Table 1: Format of "data.txt".

n,m			
X_{11}	X_{12}	...	X_{1m}
X_{21}	X_{22}	...	X_{2m}
...	...	...	...
X_{n1}	X_{n2}	...	X_{nm}

The format of the output file "output.txt" is shown in Table 2, where there are n rows, which is the same as the number of data instances in "data.txt". Each row contains a value which denotes the estimated probability density of the corresponding data instance in "data.txt". For example, P_2 denotes the estimated probability density of the 2nd data instance in "data.txt" (i.e., the 3rd row in "data.txt").

Table 2: Format of "output.txt".

P_1
P_2
...
P_n

An examples of "data.txt" and an example of "output.txt" are attached.

Regarding the implementation of the Naive Estimator method, as there is one parameter, i.e., the bin width Δ for the 1-dimensional case (each data instance is represented by only one feature or dimension) or the length of each edge of a hypercube h for the multivariate case (each data instance is represented by more than one features or dimensions), you need to **set it as 2** in your source file "naiveEst.py". Note that you are NOT allowed to use any **external** library for implementation.

For simplicity, you can assume the input data file "data.txt" and the output file "output.txt" to be generated are in the same folder as your Python source file "naiveEst.py". That means you do NOT need to consider the pathes of the input and the output files. Also, you can assume the format of the input data file "data.txt" is correct. That means you do NOT need to consider exceptions. You do NOT need to consider data preprocessing either. That means you just need to focus on how to implement the Naive Estimator method to estimate the probability density of each data instance.

Evaluation: After you submit your Python source file, we will generate a data file "data.txt" and put it in the same folder as your submitted "naiveEst.py". We will then run the your Python file in command line to see whether it could run properly to generate an output file "output.txt". Finally, we will check the correctness of the density estimation results in the generated "output.txt".