**Homework 1**

Instructions :

\* Training teams:

- GIF-4101: the assignment is made in teams of two to three students

- GIF-7005: the assignment is done individually

- Teams must be trained in myPortal before September 26th

\* Programming:

- Use Python and scikit-learn as much as possible

- Produce your solutions in the files provided, respecting the instructions

- The expected performance and the approximate computation time required are verified in the code,

any significant deviation from these expected values ​​will result in a score of zero (0) for

the corresponding sub-question

\* Discount:

- The delivery of the report and the source code is done in myPortal

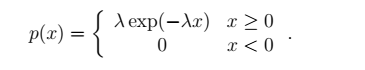
- Discount must be made no later than Wednesday, October 3rd at 9:30 am

\* Weighting:

- This assignment counts for 5% of the final grade

Statistical Estimators (5pt)

Let the exponential law, for which the probability density is defined by the following equation:



The mathematical expectation of this law is E [x] = λ1, while its variance is Var (x) = λ12.

(a) Using the procedure presented in class, calculate the expression of the estimator of λ according to a maximum of likelihood.

(b) Determine if this estimator of λ is biased in the general case.

Hint: You can consider that E [1 /x] not equal 1/E [x] in the general case.

2. Experiments with scikit-learn

Fisher's Irises form a classic data set in machine learning, which has been

first used in 1936 by R. A. Fisher to illustrate his method of **linear discriminant analysis**. Each piece of the game has four measures, the length and width of

sepals, and the length and width of petals, and that of three varieties of iris: Iris Setosa, Iris Versicolore (the floral emblem of Quebec) and Iris Virginia. The data come from iris harvested

Gaspé

Fisher's Iris game available through the scikit-learn datasets.load\_iris function.

Do the following with scikit-learn using the file d1q2.py 1, join them

your report results and provide the modified source file for your solution. Press

as much as possible your discussions by quantitative arguments and avoid verbiage.

(a) Produce graphs for your report representing the 2D dataset with class indicators, for a few pairs of measurements (sepal length vs. length of petals, length vs width of sepals, etc.), in order to **visualize** the data. Discuss briefly of the distribution of the data according to the classes.

(b) Experiment with the following parametric **classifiers**.

i**. Bayesian classifier** of normal multivariate distribution with complete covariance matrices Σi

and distinct for each class.

ii. Bayesian classifier of multivariate normal distribution with complete cov covariance matrix and shared between each class.

iii. Bayesian classifier of multivariate normal distribution with covariance matrices Σi diagonals

(σi; j = 0; 8i 6 = j) and distinct between classes.

iv. Bayesian classifier of multivariate normal distribution with isotropic covariance matrix,

let Σ = σ2I with equal values ​​on the diagonal (σj2 = σ2) and zero off the diagonal

(σj; k = 0; 8j 6 = k), with also sharing of the covariance matrix between each

class and probabilities a priori equal for each class (P (Ci) = P (Cj); 8i; j).

For each classifier tested, give the empirical error corresponding to the error rate of

ranking on the entire dataset (training game error) with each

pairs of possible measures. Also represent the results visually, by plotting

data (with class indicators) and decision regions in 2D figures, for

some of these pairs of measures. In the light of the results obtained, determine the

tested classifier that seems to have the most appropriate level of complexity for this game of

data.

(c) Compare and discuss the results obtained using three experimental methodologies:

i. Empirical error reported on the entire game (error on the training game).

ii. Random score in a training game (50%), used to evaluate the parameters of

distributions, and validation (50%), used to calculate the error in generalization. Re-

experience 10 times, each time with separate random drive / validation partitions. Report the error in general average.

iii. Performance evaluation according to a k-fold validation methodology, using

k = 3 folds. Do 10 repetitions of the experiment, with a different partitioning to

each time, and report the average error rate.

For these experiments, limit yourself to a multivariate normal law Bayesian classifier

with covariance matrices Σi complete and distinct for each class.

(d) Now create a new dataset using the function make\_circles with

the argument factor = 0.3. Test the four classifiers mentioned in (b) on this dataset, using a random partition (but identical for all classifiers) of 50% in

training and 50% validation. For each classifier tested, give the empirical error

corresponding to the classification error rate on the validation set. Represent also

the results visually, by plotting the data (with class indicators) and the regions of  
decision in 2D figures. How do you explain the difference in performance between different approaches? Which classifier seems to have the most appropriate level of complexity for this dataset?

3. Classification with rejection option . Either a Bayesian classifier of multivariate normal distribution with distinct covariance matrices for each class and isotropic, ie with equal values ​​across the diagonal and otherwise, Σi = σi2I; for i.

(a) Calculate the equation for the estimation of the parameter σi by the maximum likelihood method, providing the complete mathematical developments in your report.

(b) Suppose now that we add a rejection option to the ranking. The cost of errors is equal for all types of errors (cost of 1), except for rejection (cost of λ 2 [0; 1]). Give the complete equation to calculate the posterior probability P (Cijx) and the function for risk minimizing decision making (minimizing cost) with the rejection option.

(c) Make an implementation of the model presented in the previous point using the interface scikit-learn, allowing to use it similarly to other available algorithms in the bookstore. Implement the methods fit, predict, predict\_proba, and score in your model. For the score function, use the total cost of the application of your classifier on the data (sum of the cost of the rejections and the cost of the bad classifications). Use the file d1q3.py 2 to make your implementation and attach the source code re- sulting at your shed.

(d) Use Fisher's Iris game to test the algorithm you've implemented previous. To do this, run the algorithm by varying the cost of rejection. Test with the the following rejection costs: λ = f0; 1; 0; 3; 0; 5; 1: 0g. For each configuration, report the empirical error for the misclassification rate on the entire dataset (error on the training game). Also represent the results visually, in data (with class indicators), decision regions in 2D figures, including rejection regions, for some pairs of variables. Compare the risk of classifiers with rejection to the one that does not reject data (λ = 1). Provide your solution in the d1q3.py file.