Assignment 4
ROC curves and linear classifiers for binary classifications

1 Assignment for those who are achieving projects

1.1 ROC curve on a split predictor

- `[R,P,area]=PND_show_ROC1(name_of_file,feature,sign)` computes the ROC curve and its area concerning the splitting technique. It considers a large set of threshold and for each threshold value, it computes the recall and the precision when considering all samples from `name_of_file`. This function yields a figure when doing

  ```
  figure(1); plot(1-P,R);
  ```

1.2 Linear classifier

- `info=PND_train1(feature_l);`
  - This function selects randomly [1] the parameters needed for the linear classifier.
  - `feature_l` indicates the list of features that are considered by the linear classifier; it is a row-array containing indexes referring to columns of `X`.
  - `info` is a structure with the following fields.
    * `intercept`: value of b
    * `weights`: row vector of size `metadata.F` corresponding to $(a_f)_f$. Components corresponding to indexes not in `feature_l` are equal to 0.

  As an example of random distribution, you could consider $b = -1$ and and each non-null weight component a_f follows a uniform distribution between $\frac{1}{\max_{x_f}(x_f)}$ and $\frac{1}{\min_{x_f}(x_f)}$. This assumes that the values in the dataset are positive which I would expect to be true.

- `y_hat=PND_predict1(info,x);` It implements equation 5 of `context4.pdf`

$$\widehat{y} = \mathbf{1}\left(\sum_f a_f x_f + b > 0\right)$$

  where $(a_f)_f$ is to be found in `info.weights`, b in `info.intercept`, $\widehat{y}$ stands for `y_hat` and x_f are the components of `x`.

- `info=PND_train2();` [2] This functions uses `PND_predict1` and `PNC_score1` with `'training.mat'` as `name_of_file` to find the best value for `info` by testing randomly drawn values, a large number of times. Here *best* means getting the highest value of `overall accuracy`. `info` is a structure with three fields.
  - `feature_number` is an integer denoted here as f.
  - `threshold` is a number denoted here λ
  - `sign` is either 1 or -1 denoted here as s.
- `info=PND_train3();` derives the linear classifier by minimizing the mean square error as detailed in `content4.pdf`. `info` is a structure containing the necessary information to be used by `PND_predict1`. There are four steps, the first is to add a column to the X matrix in `training.mat`, [3], compute the pseudo-inverse of this extended matrix and multiply by the Y column-vector and read the relevant information.

1.3 ROC curve on a linear classifier using two or more features

- `[R,P,area]=PND_show_ROC2(name_of_file,info)` computes the ROC curve and its area concerning the linear classifier described in `info`. It considers a large set of intercept values and for each value, it computes the recall and the precision when considering all samples from `name_of_file`. This function yields a figure when doing

  ```
  figure(1); plot(1-P,R);
  ```

[1] The probability distribution used is of practical importance and its choice could be discussed based on experiments.
[2] `N` stands for the project number.
[3] This converts the intercept into a new weight.

1.4 Presenting the results

The `.pdf` document is named `project_ND.pdf` and contains any relevant information. The following issues are to be described.

1. Show the ROC figure using the splitting predictors for a specific feature. Pinpoint [4] the $(R, 1-P)$ values corresponding to both threshold obtained in the previous assignment (using histograms and using the second training script). Try proposing a new threshold value using this graph and test its performance.

2. Show the ROC figure using the linear predictors for a specific feature. Pinpoint the $(R, 1-P)$ values corresponding to `PND_train2` and `PND_train3`. Try proposing a new intercept value using this graph and test its performance.

3. Compare performances obtained by the splitting technique using one single feature and the linear classifying technique using two or more features.

4. Using more extensive simulations, it should be possible to show that when the training set is increased with samples having much larger feature values and still being correctly classified, `PND_train2` is quite robust to such a modification while `PND_train3` is quite sensitive. What do you think of this sensitive, is it good news?

2 Assignment for those who are reviewing projects

The goal is to build matlab functions that achieve some basic checks on the data provided along each project. Two files are to be delivered.

The first file is a `.pdf` document. Its name is `reviewer` followed by a number and an `D` indicating that it refers to the second assignment. The first part of this document explains what is tested by each test. The second part explains for each project what has passed and what has failed with precise values showing the problem. The third part is optional, it explains what supplementary information you would request from the projects and how this information could provide more valuable testing.

The second file is a `.m` script having the same name, it runs successively the different functions contained in this file that do the different testings.

Here are some ideas of tests that could be implemented.

1. Concerning `PND_show_ROC1`, If `sign` is positive, then recall is an increasing function of the threshold. If `sign` is negative, then recall is a decreasing function of the threshold. `area` should be between 0 and 1.

2. Concerning `PND_predict1`,

   - A predictor remains unchanged when a common positive number α is multiplied to both the intercept and to all weights.

$$\forall \alpha > 0,\ \forall \mathbf{x}, \quad \mathbf{1}(\sum_f a_f x_f + b > 0) = \mathbf{1}(\sum_f \alpha a_f x_f + \alpha b > 0) \tag{1}$$

   - A predictor predicts the opposite when a common negative number α is multiplied to both the intercept and to all weights.

$$\forall \alpha < 0,\ \forall \mathbf{x}, \quad \mathbf{1}(\sum_f a_f x_f + b > 0) = 1 - \mathbf{1}(\sum_f \alpha a_f x_f + \alpha b > 0) \tag{2}$$

3 Discussion

Your task is first of all to read all projects and check `Progress`. You should write a single `.pdf` document, named `discussionD.pdf` discussing how all projects have undergone this first step, the difficulties that have been overcome and those that remain challenging issues. You should then express your opinion as to whether I should come back on some specific issues. You may also add some specific comments to a specific project on *Discussions* [5] and some specific questions on *Questions*. You are also expected to write in *Questions* the answers to all other questions.

[4] pinpointing can be done with `gtext` in Matlab.

[5] Comments should be most respectful as any work needs attention, and regardless of it being possibly wrong, it is going to be useful to get a better understanding. So there can be no shame in being wrong.