Anomaly detection

The goal of this homework is to implement two anomaly detection methods, tune their hyper-parameters, compare their accuracy using area under receiver operator characteristic on a provided set of problems and test, if one of them is statistically better.

Instructions

1. Implement anomaly detectors
   - Implement anomaly detector based on parzen window estimator with width of kernel being a hyper-parameter. The implementation should be stored in a file `parzen.r`.
   - Implement anomaly detector based on Gaussian mixture model (GMM) of distributions. Fit mean and variance of each component using Expectation Maximization Algorithm (see `https://people.csail.mit.edu/rameshvs/content/gmm-em.pdf`). The anomaly detector would have a single hyper-parameter, which will be a number of components. The implementation should be stored in a file `mog.r`.

   The above function should be implemented by yourself. You can use library `pdist` for calculating Euclidean distances.

2. Since Parzen window estimator and GMM models both are valid probabilistic models, you can select hyper-parameters on validation data by maximizing likelihood (use only normal data).

3. To measure accuracy of your anomaly detector, you calculate the area under ROC curve (AUC), for which you can use library `AUC`. `https://en.wikipedia.org/wiki/Receiver_operating_characteristic`

4. Each problem contains normal data (in the file `normal.txt`) and anomalous data (in the file `anomalous.txt`. To create a valid conditions for evaluating anomaly detection, you randomly split the normal data into training set for fitting the model(s) (50% of samples normal samples), validation set for selecting hyper-parameters (25% of samples normal samples), and testing set for evaluation (25% of normal samples and 50% of anomalous samples). Training and validation set will contain only normal data, testing set has contain anomalous data (since ROC does not depend on class priors, you can add as many anomalous samples as you wish to have the precise estimate of AUC).

   You should repeat the above experiment for each detector on each dataset ten times and store the result in a dataframe, where row corresponds to a problem and repetition, column corresponds to an anomaly method, and each cell contains AUC on the testing set.

5. Choose and implement *robust* statistical test to assess, which anomaly detector is better. Justify your choice of the statistical test. The input to the test should be the dataframe obtained in previous step.

6. Document the experimental protocol, results, and draw conclusions.

Unless explicitly stated, you are not allowed to use libraries for anomaly detection. By implementing it is understood writing the algorithm by yourself and not calling the library. Calling the library immediately renders zero points.

Each step is graded by two point, the final protocol is awarded by three points, only in case that all steps were implemented. In total you can therefore earn $6 \times 2 + 3 = 15$ points. Note partial points for partial solutions are not awarded. That said, you either finish the task, which means you get two (or three) points, or you do not finish it, in which case you receive zero. As an engineers you should be able to design tests verifying your solutions.