Anomaly detection

The goal of this homework is to implement two anomaly detection methods and identify, which one is statistically better.

Instructions

1. Implement anomaly detector based on k-nearest neighbor. You can use library `FNN` for finding nearest neighbours. The implementation should be stored in a file `knn.r`.

2. Implement anomaly detector based on fitting Mixture of Gaussian distributions. You can use library `pdist` for calculating Euclidean distances. The implementation should be stored in a file `mog.r`.

   `https://people.csail.mit.edu/rameshvs/content/gmm-em.pdf`

3. Evaluate your anomaly detectors on a provided set of testing datasets using the area under ROC curve (AUC), for which you can use library `AUC`. The implementation should be stored in a file `evaluation.r` and the output should be a dataframe, where row corresponds to a problem, column corresponds to a anomaly method, and each cell contains AUC on the testing set. `https://en.wikipedia.org/wiki/Receiver_operating_characteristic`

   Since both methods (k-nearest neighbor and mixture of Gaussians) require hyper-parameters, select them using a leave-one-dataset out method (see hints for details).

4. Implement a 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.

Each step is graded by one point. In total you can earn four points.

Hints

Mixture of Gaussians are fitted using Expectation Maximization algorithm. It is sufficient to fit location, μ, of each component.

Testing problems Each directory with testing problems contain files `normal.txt` and `anomalous.txt`. The first one contains normal samples and the second contain anomalous samples.

Evaluating detector Randomly split available normal and anomalous samples to training and testing sets. Fit a detector on normal samples from the training set (do not be surprised that you are not using the anomalous samples). Measure the quality of the fitted detector on the data in testing set.

Leave-one-dataset out setting of hyper-parameters means that if you have n datasets, you select them based on a performance evaluated on a testing set of $n - 1$ of them and then test the method on n-th dataset. You repeat this n-times, every time skipping a different dataset.