EE527: Machine Learning Laboratory

Assignment 9

Due Date: 4 April 2022

1. ROC of Chebyshev inequality-based classifier.

(a) Write the following function in python to generate n number of points in a ring on $2D$ plane centered at $[c_x, c_y]$ of respective inner and outer radius r_{inner} and r_{outer}.

$$genRandPointsInRing(r_{inner}, r_{outer}, [c_x, c_y], n)$$

Please note that we have to check and maintain the condition that $r_{inner} > 0$ and $r_{outer} > r_{inner}$. Also, the same function can be used to generate points within a circle of radius r_{circle} by setting $r_{inner} = 0$ and $r_{outer} = r_{circle}$.

(b) Using the above function generate $n^{(+)} = 250$ points belonging to the positive class in a circle of radius $r_{circle} = 5$ centered at $[c_x, c_y]^T = [0,0]^T$. Using the same function generate $n^{(-)} = 350$ points belonging to the negative class in a ring of radii $r_{inner} = 4$, $r_{outer} = 8$ and centered at $\left[c_x, c_y\right]^T = [0,0]^T$. Display the scatter plot of the positive class points and negative class points in red and blue colors respectively.

(c) Use the Chebyshev inequality-based classifier on the positive class points and plot the ROC by varying the threshold λ in the interval $[0.01,10]$ in steps of 0.05.

(d) Experiment with non-overlapping classes ($r_{circle} < r_{inner}$) and overlapping classes ($r_{circle} > r_{inner}$) and plot the ROCs for the same. Please note that r_{outer} is always greater than r_{circle} and r_{inner}.

2. Incremental Clustering

(a) Write a function $generatePointFromRandomCluster(\,)$ that randomly generates a point in $\boldsymbol{R}^2$. The point must lie within any one of the following $N = 17$ circles $\{\boldsymbol{C}_i; i = 1, \dots N\}$ that are specified in the $(Center - X\,, Center - Y, Radius)$ format.

(0, 0, 10) ; (0, 50, 15) ; (50, 0, 15) ; (0,-50, 15) ; (-50, 0, 15) ; (35, 35, 15) ; (35,-35, 15) ; (-35, 35, 15) ; (-35,-35, 15) ; (0, 100, 20) ; (100, 0, 20) ; (0,-100, 20) ; (-100, 0, 20) ; (70, 70, 20) ; (70,-70, 20) ; (-70, 70, 20) ; (-70,-70, 20).

Any particular call to this function randomly chooses a circle and generates a point within it.

(b) The data point $x_t \in \boldsymbol{C}_i \ (i = 1 \dots N)$ at each instant is obtained using the function $generatePointFromRandomCluster(\,)$. Perform the incremental clustering with default variance $v_d = 10$ and Chebychev inequality threshold $\lambda = 3$.

(c) When the number of clusters increase beyond 50, drop the cluster with lowest weight value (π), so that at any given iteration, there are a maximum of $K = 50$ clusters.

(d) Plot the clusters as differently colored ellipses after every 100 iterations.

3. Consider the *MNIST* dataset used in *Assignment 7*. Use *50000* images (*5000* for each hand-written digit category) for train set and *10000* (*1000* for each category) for test set. Perform PCA on the train set and save the linear transformation parameters $\boldsymbol{\mu} \in \mathbb{R}^{768}$ and $Q \in \mathbb{R}^{768 \times d}$. Here, d is the dimension of the target lower dimensional space. Using $\{\boldsymbol{\mu}, Q\}$, project train set images $\Omega_i^{(train)}$ and test set images $\Omega_j^{(test)}$ to respective vector embeddings $\boldsymbol{x}_i^{(train)}$ and $\boldsymbol{x}_j^{(test)}$ ($i = 1, \dots 50000$, $j = 1, \dots 10000$). Learn *10* Gaussian Mixture Models GMM_k $(k = 1, \dots 10)$ from the training data set $\{\boldsymbol{x}^{(train)}\}$ of each digit category. Each GMM should have K modes. All modes should have diagonal covariance matrices. Classify each instance from the test set $\{\boldsymbol{x}^{(test)}\}$ by using the *Bayes Classifier* with *MAP Rule*. The likelihood for each category should be generated by using the corresponding GMM learned from the train set. Vary the value of the number of GMM modes ($K = 1,5,10, \dots 50$) and report the *Total Accuracy* and *F1 -Score* of each digit category for different values of K.