Introduction to Machine Learning
Homework 10: Clustering: K-means and EM-GMM algorithm

Prof. Sundeep Rangan

1. You are given five data samples:

i	1	2	3	4	5
x_{i1}	0	1	0	2	2
x_{i2}	0	0	1	2	3

   (a) Draw the five points.

   (b) Starting with $K = 2$ cluster centers at $(0, 0)$ and $(1, 0)$, what are the cluster assignments and new cluster centers after one iteration of K-means?

2. *K-means for outlier detection.* Write a function for outlier detection:

```
def outlier_detect(Xtr,Xts,nc,t):
    ...
    return outlier
```

   The function should:

   - Perform K-means clustering on the training data `Xtr`;
   - Given the matrix of test data `Xts`, it sets an output `outlier[i]=1` if the sample `Xts[i,:]` is greater than some distance `t` from all cluster centers.

   Try to avoid for loops. You may assume you have the following functions:

```
km = KMeans(n_cluster=nc)   # Creates a K-Means object
km.fit(X)  # Fits the k-means clusters
km.cluster_centers_   # Returns the cluster centers
                      # (one cluster per row)
```

3. *Initialization.* Write a few lines of python code to initialize K-means by selecting K random samples of the training data as the cluster centers. Make sure you do not pick the same sample twice.

4. *Clustering as pre-processing.* Suppose we want to cluster data and then fit a linear model in each cluster. You are given training data `Xtr,ytr` and test data `Xts,yts` for a regression problem. Write code to do the following:

   - Perform K-means clustering on the training data `Xtr` with a given number `nc` clusters;

- In each cluster in the training data, fit a linear model.
- Compute the predicted outputs `yhat_ts` and mean squared error of the model on the test data.

You may assume you have the following functions:

```
km = KMeans(n_cluster=nc)   # Creates a K-Means object
km.fit(X)       # Fits the k-means clusters
km.predict(X)   # Finds the index of the closest cluster

reg = LinearRegression()   # Creates a linear regression object
reg.fit(X,y)  # Fits the linear model
yhat = reg.predict(X)  # Predicts the output
```

Note: You may need a list of regression objects.