CS5691: Pattern Recognition and Machine Learning
Assignment 2

Name: Avinash Singh Kushwah **Roll no: CS22M024**

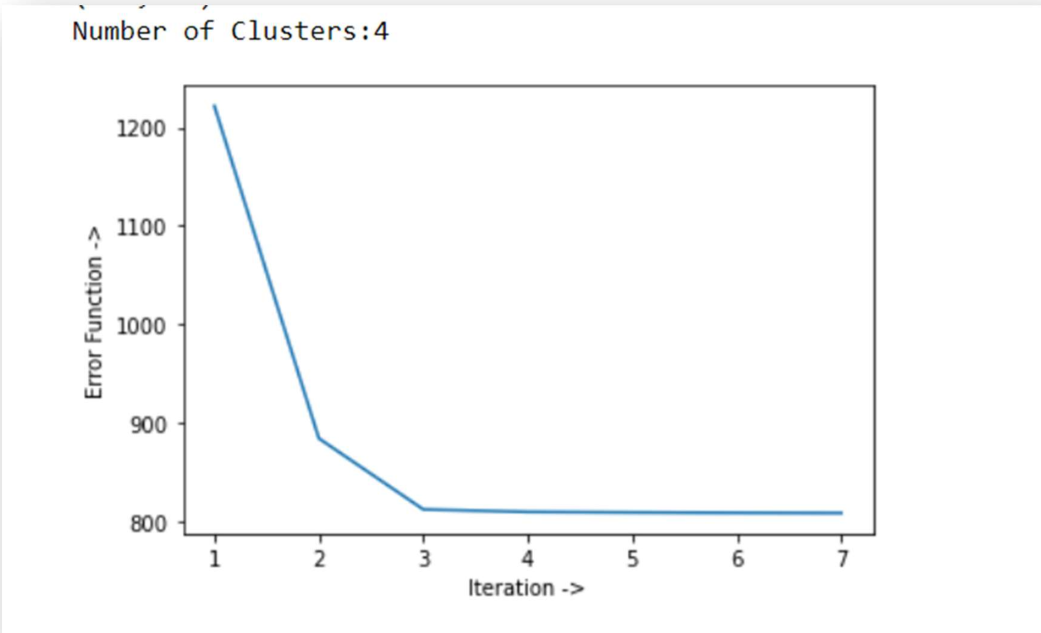
1.You are given a data set with 400 data points in {0,1}^50 generated from a mixture of some distribution in the file A2Q1.csv.

II. Assume that the same data was in fact generated from a mixture of Gaussians with 4 mixtures. Implement the EM algorithm and plot the log-likelihood (averaged over 100 random initializations of the parameters) as a function of iterations. How does the plot compare with the plot from part (i)? Provide insights that you draw from this experiment.
Soln. I was only able to implement the gaussian mixture model with 4 mixtures.

```python
def gaussianDistribution():

    #caclulating mean
    mean = np.zeros(col)
    for i in range(col):
        for j in range(rows):
            mean[i] += dataset[j][i]
        mean[i] = mean[i]/rows


    #calculating covariance
    sigma = np.dot(dataset.T,dataset)
    #print(mean.shape)
    #print(sigma.shape)
    temp =(dataset - mean).T

# print(temp)
#     det = numpy.linalg.det(sigma)
#     temp1 = 1 / ((pow((2*3.14),(col/2))) *(math.sqrt(det)))
#     temp2 =  math.exp(-(1/2)*((dataset - mean).T * (np.linalg.inv(sigma)) * (dataset- mean))
#     print(temp1*temp2)
#     #print(gauss)
    temp1 = ((2 * np.pi) ** (col / 2) * np.linalg.det(sigma) ** (1/2))
    temp2 = np.exp(-0.5 * np.dot(np.dot(temp.T, np.linalg.inv(sigma)), temp))
    return np.diagonal(1 / temp1* temp2)
```

III. Run the K-means algorithm with K = 4 on the same data. Plot the objective of K - means as a function of iterations.
Soln. Input = 4 (number of clusters).

Below is the plot of the objective of K-means as a function of iterations.



IV. Among the three different algorithms implemented above, which do you think you would choose for this dataset and why?

Soln. I would use the gaussian mixture model because K means is an unsupervised learning technique while mixture models are supervised learning techniques. K means uses hard clustering data points to a cluster, if we are uncertain about the data points where they belong or to which group, we use the gaussian mixture model.

2. You are given a data set in the file A2Q2Data train.csv with 10000 points in (R^100, R)(Each row corresponds to a datapoint where the first 100 components are features and the last component is the associated y value).

I.Obtain the least squares solution wml to the regression problem using the analytical solution.

Soln. The least squares solution wml to the regression is :

wml = (xTx)^(-1) xTy

Where x is (10000 *100) & y is a label corresponding to each data point (10000*1)

```
[-7.84961009e-03 -1.36715320e-02 -3.61656438e-03  2.64909160e-03
  1.88551446e-01  2.65314657e-03  9.46531786e-03  1.79809481e-01
  3.73757317e-03  4.99608944e-01  8.35836265e-03  4.29108775e-03
  1.42141179e-02  3.94232414e-03  9.36795890e-03 -1.12038274e-03
  3.35727500e-03  1.16152212e-03 -9.40884707e-03 -2.45575476e-03
 -1.17409629e-02 -1.01960612e-02  7.95771321e-03 -1.00574854e-02
  6.04882939e-03 -4.67345192e-03 -3.09091547e-03  8.14909193e-03
  1.20264599e-02 -6.82458163e-03 -8.65405539e-03  9.86273479e-04
  4.92968011e-03  5.99772461e-03 -1.34667860e-02  1.07075729e-03
  1.32745992e-02 -1.14148742e-02 -2.01056697e-02  5.85096240e-01
  4.94483247e-04 -7.86666920e-04 -2.71926574e-03 -9.54021938e-03
 -5.44161058e-03  9.80679209e-03 -6.72540624e-03 -4.45414276e-04
  6.98516508e-03  3.16138907e-02  4.51763485e-01 -8.75221380e-03
  2.55167390e-03  4.24921150e-03  2.89847927e-01  7.03723255e-03
 -1.95796946e-03  1.41523883e-02 -1.06508170e-02  7.72743903e-01
 -5.67126044e-03 -6.30026188e-04  6.50943015e-03 -4.84019165e-03
  4.63832329e-03  4.54887177e-03 -2.99475114e-03  8.38781696e-03
 -2.47558716e-03  9.00947922e-04  1.14713514e-03 -1.87641345e-03
 -1.05175760e-02 -9.31304110e-03 -1.23550002e-03  5.97797559e-01
 -4.78625013e-03 -1.13727852e-02  2.88477060e-03  8.48999776e-01
 -1.08924235e-02  2.26346489e-03 -1.38099800e-03 -6.35934691e-03
  5.83784109e-03  5.69286755e-03  5.35566859e-03 -8.20616315e-03
  1.29884015e-02 -2.30575631e-03 -1.22263765e-04  8.66629171e-03
 -4.29446300e-03  5.69510898e-03  7.55483353e-03 -9.43540843e-03
  1.82905446e-02 -1.16998887e-03 -2.61599136e-03 -8.58616114e-03]
```

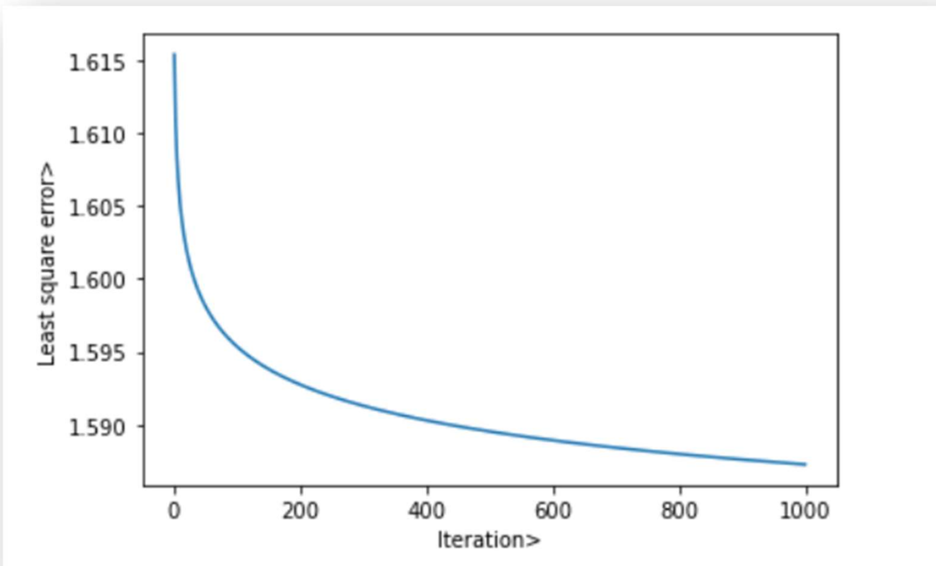
II. Code the gradient descent algorithm with a suitable step size to solve the least squares algorithms and plot || w-wml || as a function of t. What do you observe?

Soln.

Following are the steps for the gradient descent algorithm to solve the least square algorithm:

a. Initialize w = [0,0,0,0,0,0 … 0].
b. Find wml ((xTx)^(-1) xTy) { x is a dataset, y is a label }
c. Define number of iterations (T).
   for each iteration:
      stepsize = 0.000001/t;
      w= w - stepsize * gradient(w)
      calculate the L2 norm || w – wml ||

Plot of || w- wml || as a function of iteration :



III. Code the stochastic gradient descent algorithm using a batch size of 100 and plot || w- wml || as a function of t. What are your observations?

Soln.

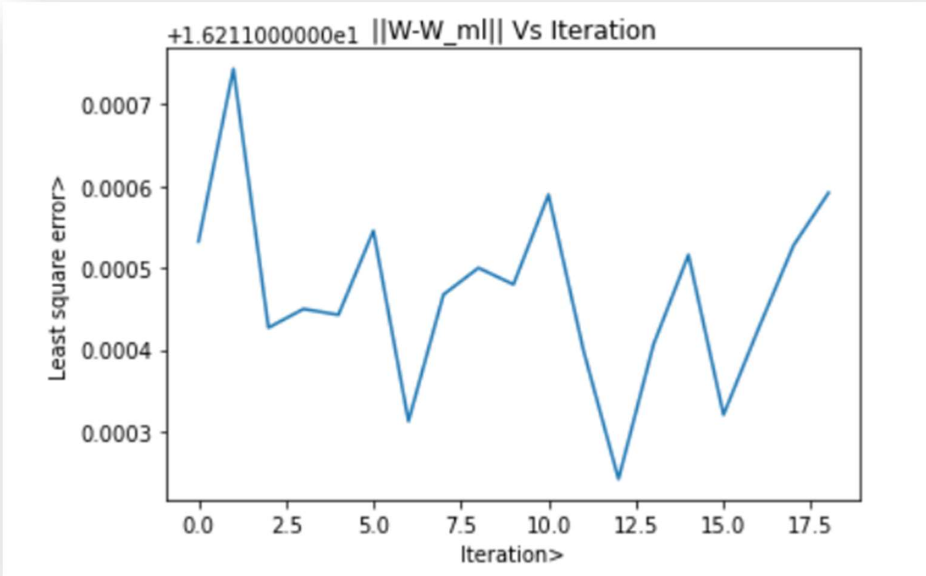
This is similar to mini-batch gradient descent where we are taking a batch of size =100. In this algorithm, at each iteration, we randomly pick 100 data points from our dataset instead of using the whole dataset.

Note: we can pick 100 data points randomly using two ways:

1. First reshuffle our dataset and select [0..99] in the first iteration,[100..199] in the second iteration, and so on.
2. Pick 100 data points uniformly at random at each iteration,

Stochastic gradient descent is computationally faster than the gradient descent algorithm because it works on a subset of the data point at each iteration instead of the whole dataset but SGD may take more iterations to converge.

Plot of || w -wml || as a function of iteration:



IV. Code the gradient descent algorithm for ridge regression. Cross-validate for various choices of lambda and plot the error in the validation set as a function of lambda. For the best-chosen lambda, obtain wR. Compare the test error (for the test data in the file A2Q2Data test.csv) of wR with wML. Which is better and why?

Soln.

```
Z=3500
z=2500
w2=[]
err=[]

while ( z< Z):
    w1=[]
    e=[]
    for i in range(5):
        w1.append(gradientDescent(train[i],z))
    w2=[sum(col)/5 for col in zip(*w1)]
    for i in range(5):
        e.append(test_d(w2,test[i],z))
    err.append(sum(e)/len(e))
    e.clear()
    w1.clear()
```

Wr is better than Wml, Wr avoid overfitting of data, if we change our data slightly than ridge regression changes its parameter only a bit but normal regression changes its parameter more.