CS5691 Assignment 2

Pattern Recognition and Machine Learning

Arjun Vikas Ramesh

CS21B008

April 14, 2024

1 Question 1

1.1 EM algorithm using bernoulli mixture

- Our choice of mixture(**bernoulli distribution**) is derived from the observation that the given data points have features of values 0 or 1. Therefore an underlying bernoulli distribution for each feature would have generated the data. The given data of dimension 50 can be interpreted as a flattened version of a black and white, 5 by 10 image

- We will consider a bernoulli distribution with each cluster k having parameter p_k of dimension D

  - The likelihood is computed as :

$$L(x, p, \pi) = \prod_{i=1}^{n} \sum_{k=1}^{K} (\pi_k . (\prod_{d=1}^{D} p_{k,d}^{x_{i,d}} . (1 - p_{k,d})^{1 - x_{i,d}})) \tag{1}$$

  - Lambda at Expectation step is calculated as :

$$\lambda_i^k = \frac{(\pi_k . (\prod_{d=1}^{D} p_{k,d}^{x_{i,d}} . (1 - p_{k,d})^{1 - x_{i,d}}))}{\sum_{k=1}^{K} (\pi_k . (\prod_{d=1}^{D} p_{k,d}^{x_{i,d}} . (1 - p_{k,d})^{1 - x_{i,d}}))} \tag{2}$$

  - π_k is calculated at Maximisation step as :

$$\pi_k = \frac{\sum_{i=1}^{N} \lambda_i^k}{N} \tag{3}$$

  - p_k is calculated at Maximisation step as :

$$p_k = \frac{\sum_{i=1}^{N} \lambda_i^k . x_i}{\sum_{i=1}^{N} \lambda_i^k} \tag{4}$$

- We run the algorithm and average over 100 random initializations to obtain the following likelihood vs iteration plot

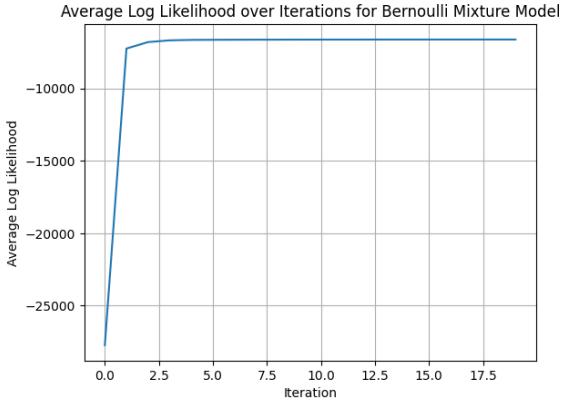

Figure 1: log-likelihood as a function of iterations for Bernoulli mixture

1.2 EM algorithm using GMM

- We perform the EM algorithm using a Gaussian mixture model with randomized means and covariance. Since the given data is multi-dimensional, the mulivariate Gaussian pdf for vectors was used.

$$\mathcal{N}(x|\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)} \tag{5}$$

- We run the algorithm and average over 100 random initializations to obtain the following likelihood vs iteration plot

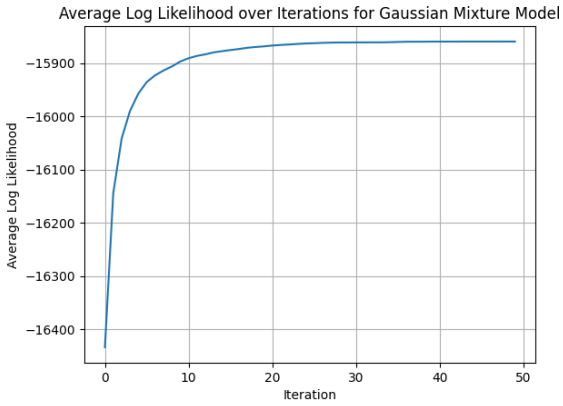

Figure 2: log-likelihood as a function of iterations for Gaussian mixture

- We observe from the plots that, computationally, the bernoulli mixture model converges faster to its higher likelihood value, than the Gaussian mixture model.
- Also, the number of parameters to estimate for the bernoulli model is $k * n$ for λ_k^i, k for π_k, and $k * d$ for p_k^d while for gaussian model is $k * n$ for λ_k^i, k for π_k, $k * d$ for μ_k, and $k * d * d$ for Σ_k
- Therefore, computationally the bernoulli mixture is faster and has less parameters to estimate/maximise, and also converges in less iterations.

1.3 K-means algorithm

- Running the K-means algorithm, we obtain the following plot of objective function vs iterations:

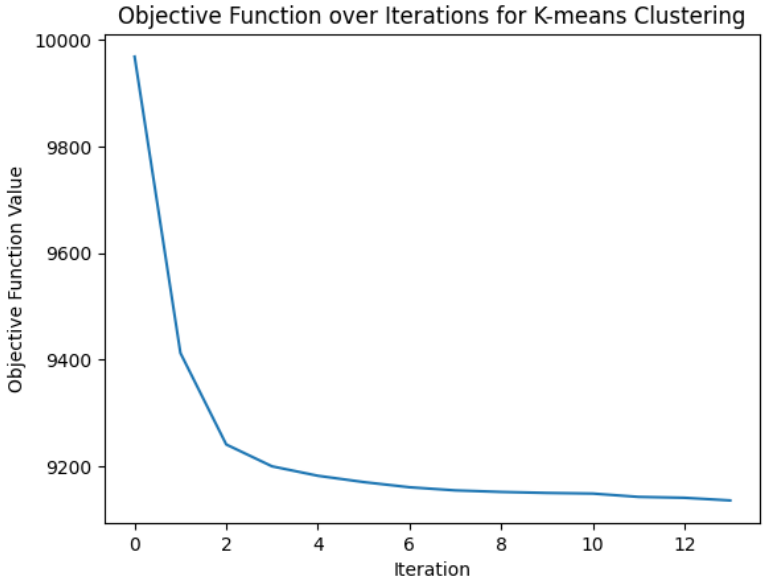

Figure 3: Error function over iterations for K-means

1.4 Optimal clustering algorithm

- We observe that the K-means also converges in a few iterations.

- w.r.t choosing Gaussian or Binomial mixture, we already inferred to chose binomial over gaussian. So the decision lies in choosing EM or K-means algorithm.
- When we want to differentiate an outlying data between two clusters, the larger cluster will have more probability of generating that data in EM clustering.
- The clustering in EM is soft, so we have the inference that the outlying data may have been generated by either clusters. However, such an inference cannot be obtained in K means algorithm.
- Therefore using EM algorithm is more suitable

2 Question 2

2.1 Analytical solution to Linear regression

- We compute w_{ML} as

$$w_{ML} = (X.X^T)^{-1}.X.y \tag{6}$$

to obtain:

```
[-7.84961009e-03 -1.36715320e-02
-3.61656438e-03  2.64909160e-03
1.88551446e-01  2.65314657e-03
9.46531786e-03  1.79809481e-01
 .......
-4.29446300e-03  5.69510898e-03
7.55483353e-03 -9.43540843e-03
1.82905446e-02 -1.16998887e-03
-2.61599136e-03 -8.58616114e-03]
```

2.2 Gradient Descent Algorithm

- We code the gradient descent algorithm with step size inversely proportional to iteration, and obtain the plot $||w_{GD}^t - w_{ML}||_2$ as a function of t.

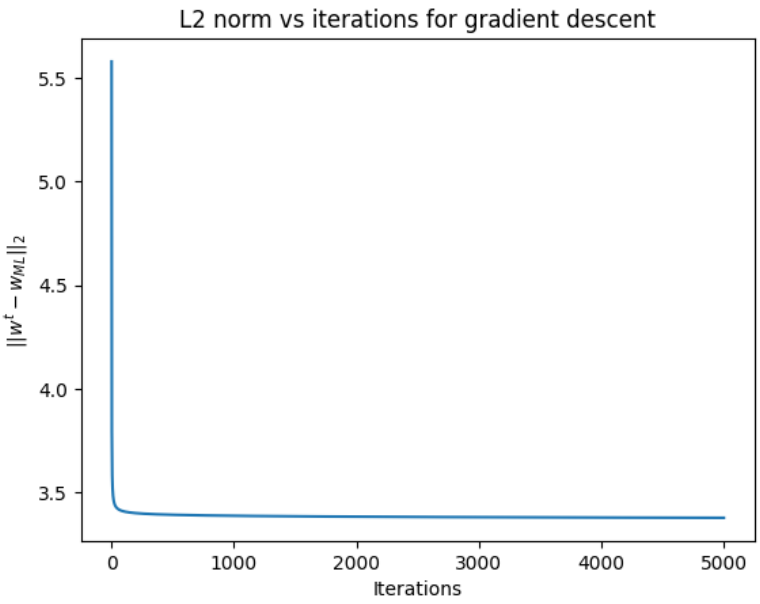

Figure 4: $||w_{GD}^t - w_{ML}||_2$ vs iteration

Step-size	Squared L2 norm
1e-07	105200
1e-06	420
11e-05	800
1e-03	8e267

Table 1: Step-size vs error for w

- We observe that the L2 norm of the difference of the analytical solution and gradient descent converges to a minimum in very few iterations.
- Step size was chosen appropriately by measuring Squared norm error on test data for various initial step size, which decrease inversely proportional to t.

2.3 Strochastic Gradient Descent

- We code the gradient descent algorithm with step size inversely proportional to iteration, and obtain the plot $||w_{SGD}^t - w_{ML}||_2$ as a function

of t.

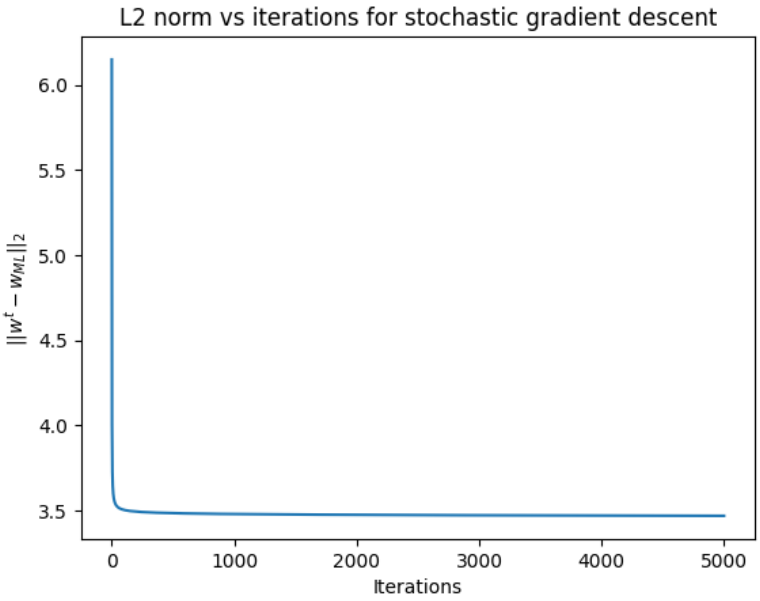

Figure 5: $||w^t_{SGD} - w_{ML}||_2$ vs iteration

- We observe that the L2 norm of the difference of the analytical solution and gradient descent converges to a minimum in very few iterations.
- The stochastic gradient descent is computationally less expensive since at each step a smaller sample size is used to compute the gradient.
- We also observed by performing several runs, that SGD is less sensitive to the random initialisation of w, in comparison to the GD algorithm.

2.4 Ridge Regression

- We code the gradient descent algorithm for ridge regression, and determine the optimal λ using k-fold cross validation on the training data set with $k = 4$. We obtain the following plot of mean error in validation set as a function of λ, which varies as l^2 where l in 0,100.

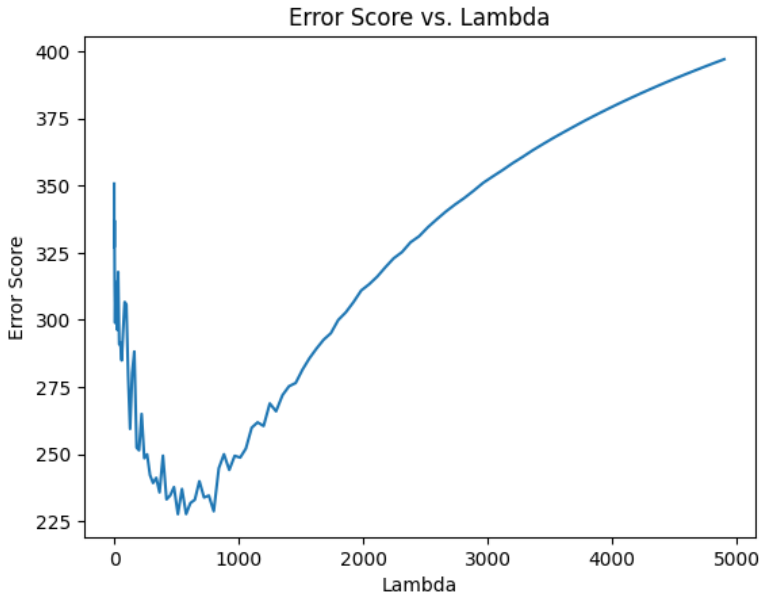

Figure 6: $||Y_{test} - X^T.W_r||_2^2$ vs λ

- We use the above obtained lambda to perform ridge regression. The following errors were obtained on the Test data provided.

w	Squared L2 norm error
w_{ML}	185
w_R	124

Table 2: errors for test data

- We observe that ridge regression performs better. This is because, ridge regression accounts for any overfitting of data that may arise in linear regression. There may be noise associated with the dataset that was used to train the model. Ridge regression accounts for this by introducing penalty on larger w. Therefore w_R is better.

END