CS5691 Pattern Recognition and Machine Learning

Assignment 2

Submitted by

CS22M025 Ayana Satheesh B

Indian Institute of Technology Madras

Contents

List of Figures

List of Tables

Chapter 1

Question 1

1.1 Binomial Mixture Model

Since the given data belongs to 0,1 it should have been generated from a coin toss model. The generative story is that let there be 4 coins with probability p1, p2, p3, p4 respectively. The probability of choosing the kth coin is given by πk. Now this coin is tossed d times (where d is the dimension of data point) to generate the data. These parameters p1,p2,p3,p4 and πk (for randomly choosing any of the 4 coins) which are unknown is to be estimated. The derivation of the EM algorithm is depicted in figure1.1, figure1.2, figure 1.3. The plot of log likelihood over the iterations is depicted in figure 1.4

1.2 Gaussian mixture model

The plot of log likelihood for gaussian mixture model as a function of iterations is shown in figure 1.5. The plot is different from that of the binomial mixture model since it is fluctuating between the values and the maximum likelihood obtained when the algorithm converges is found to be low. Also the plot of the gaussian model is not smooth unlike binomial mixture model.

1.3 Kmeans

The plot for kmeans is shown in figure 1.6. The error function is high even though it reaches steady state.

1.4 Best Algorithm

The best model for the given dataset is binomial mixture model since the data points are either 0 or 1 and from the plot it is also visible that the it reaches a steady state when the parameters converges. Also higher the log likelihood better the model is. For binomial mixture model , the log likelihood function when the parameters converges is found to be high compared to gaussian mixture model. Also gaussian mixture model have more parameters to estimate than binomial mixture model proposed above. And the plot becomes more stable in binomial mixture model.

Likelihood function ⇒ $L(\theta)$ ⇒

$$L(\theta) = \prod_{i=1}^{n} P(x_i;\theta)$$

$$= \prod_{i=1}^{n} \sum_{k=1}^{K} P(x_i, z_i = k;\theta)$$

∵ There is some mixture model (need not be Gaussian) from which we sample the data once the data is fixed.

∵ the data given is $\{1,0\}$, it must be generated by some ~~two~~ coin toss model.

π_k ⇒ probability of x_i coming from kth cluster/mixture

P_k ⇒ probability of head (i.e. 1) for coin in kth mixture

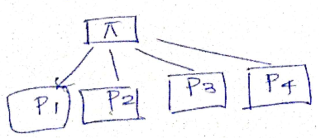
θ ⇒ parameters $= \pi, P$.

x_{i0} ⇒ no. of 0s in x_i

x_{i1} ⇒ no. of 1s in x_i

$$\therefore L(\theta) = \prod_{i=1}^{n}\left[\sum_{k=1}^{K} \pi_k (1-P_k)^{x_{i0}} (P_k)^{x_{i1}}\right]$$

i.e. It is a Binomial Mixture Model with K-biased coins with diff. probabilities (P_1, P_2, P_3, P_4) & each of the K coins generate the data by bernoulli trial i.e.



i.e. $x_i|_{z=k} \sim \text{Bernoulli}(P_k)$

& this process is repeated n times

Figure 1.1: Derivation of EM algorithm for binomial mixture model

∴ It is a Binomial Mixture Model.

Taking Log on both sides,

$$\log L(\theta) = \log \prod_{i=1}^{m} \left[\sum_{k=1}^{m} \pi_k (1-P_k)^{x_{i0}} P_k^{x_{i1}} \right]$$

$$= \sum_{i=1}^{m} \log \sum_{k=1}^{k} \pi_k (1-P_k)^{(50-x_{i1})} P_k^{(x_{i1})}$$

$$(x_{i1} + x_{i0} = 50)$$

$$= \sum_{i=1}^{m} \log \sum_{k=1}^{k} \lambda_k^i \left[\frac{\pi_k (1-P_k)^{(50-x_{i1})} P_k^{x_{i1}}}{\lambda_k^i} \right]$$

⇒ According to Jensen's inequality

$\log L(\theta) \geq$ modified $\log L(\theta, \lambda)$

$$\geq \sum_{i=1}^{m} \sum_{k=1}^{k} \lambda_k^i \log\left(\frac{\pi_k (1-P_k)^{50-x_{i1}} P_k^{x_{i1}}}{\lambda_k^i} \right)$$

For we fix λ & maximise over θ:

$$\max_{\theta} \sum_{i=1}^{m} \sum_{k=1}^{k} \lambda_k^i \left[\log(\pi_k) + (50-x_{i1}) \log(1-P_k) + x_{i1} \log P_k - \underbrace{\log \lambda_k^i}_{\text{constant}} \right]$$

$$\cong \max_{\theta} \sum_{i=1}^{m} \sum_{k=1}^{k} \lambda_k^i \left[\log \pi_k + (50-x_{i1}) \log(1-P_k) + x_{i1} \log P_k \right]$$

~~Derivative~~ Taking Derivative w.r.t. p ⇒

$$\sum_{i=1}^{m} \sum_{k=1}^{k} \lambda_k^i \cdot 0 + \frac{\lambda_k^i (50-x_{i1})}{(1-P_k)} - \frac{\lambda_k^i x_{i}}{P_k} = 0$$

$$\sum_{i=1}^{m} \sum_{k=1}^{k} \lambda_k^i \left[\frac{(50-x_{i1})}{P_k - 1} - \frac{x_{i1}}{P_k} \right] = 0$$

Figure 1.2: Derivation of EM algorithm for binomial mixture model

$$\sum_{i=1}^{m} \frac{\lambda_k^i (50 - x_{i1})}{(1 - P_k)} - \frac{\lambda_k^i x_{i1}}{P_k} = 0$$

$$\sum_{i=1}^{m} \lambda_k^i P_k 50 - \lambda_k^i P_k x_{i1} - \lambda_k^i x_{i1} + \lambda_k^i P_k x_{i1} = 0$$

$$\sum_{i=1}^{m} \lambda_k^i P_k 50 - \lambda_k^i x_{i1} = 0$$

$$\sum_{i=1}^{m} \lambda_k^u P_k 50 = \sum_{i=1}^{m} \lambda_k^u x_{i1}$$

$$\Rightarrow P_k = \frac{1}{50} \frac{\sum_{i=1}^{m} \lambda_k^i x_{i1}}{\sum_{i=1}^{m} \lambda_k^i}$$

Derivative w.r.t. π

$$\sum_{i=1}^{m} \sum_{k=1}^{k} \frac{\lambda_k^i}{\pi_k} = 0$$

For a particular k,

$$\sum_{i=1}^{m} \frac{\lambda_k^i}{\pi_k} = 0$$

$$m\pi_k = \sum_{i=1}^{m} \lambda_k^i$$

$$\pi_k = \frac{\sum_{i=1}^{m} \lambda_k^i}{m}$$

λ_k^i can is derived from Bayes Theorem

$$\lambda_k^i = \frac{P(x_i | z_i = k) \cdot P(z_i = k)}{P(x_i)}$$

$$= \frac{\pi_k P_k^{x_{i1}} \cdot (1 - P_k)^{(50 - x_{i1})}}{\sum_{u=1}^{k} \pi_u (P_u)^{x_{i1}} (1 - P_u)^{(50 - x_{i1})}}$$

where λ_k^u is the probability of data point x_i coming from kth cluster.

Figure 1.3: Derivation of EM algorithm for binomial mixture model

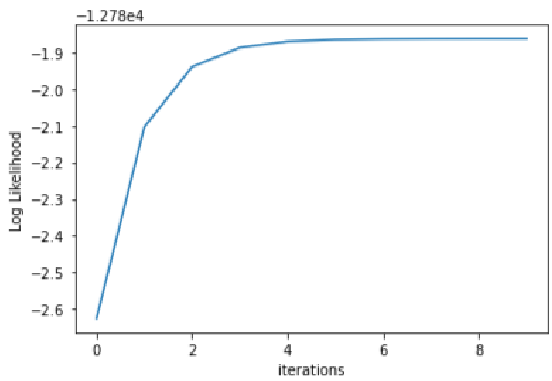


Figure 1.4: Plot of log likelihood over iterations for binomial mixture model

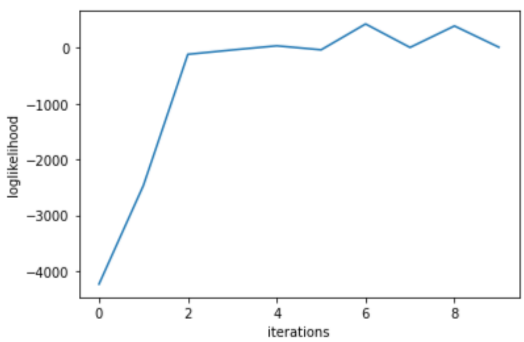


Figure 1.5: Plot of log likelihood over iterations for Gaussian mixture model

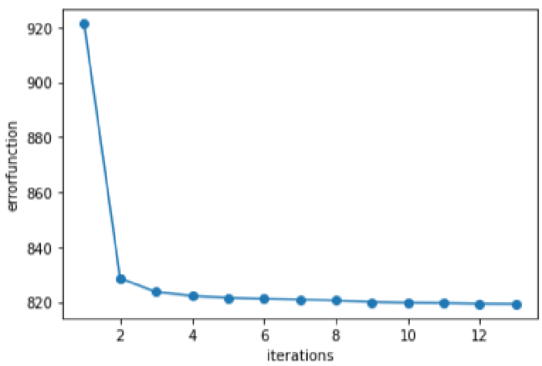


Figure 1.6: Plot of errorfunction of kmeans over iterations

Chapter 2

Question 2

2.1 Linear Regression using Analytical method

The least squared solution obtained using analytical method is shown in figure 2.1. The error on train data obtained using analytical w is shown in figure 2.2. The w performs well on training data but on test data the error is more.

```
The squared sum of error on test data for the calculated w is: 185.36365558489584
```

Figure 2.1: Least squared solution on test data using analytical method

```
print("The squared sum of error on train data for the calculated w is:", errorsum)
The squared sum of error on train data for the calculated w is: 18.845155496357222
```

Figure 2.2: Error on train data using analytical method

2.2 Linear Regression using gradient descent

The gradient descent algorithm for linear regression problem produces the same least square error (figure 2.3) on test data which means w obtained from gradient descent and wML is same. As it is also visible from the plot(figure 2.4) after certain number of iterations their difference approaches zero. This is a useful result especially when the dimension of the data point is large, it will be computationally expensive to calculate wml (analytical solution) due to inverse present in the formula. As the gradient descent converges to same wml ,this method could be useful.

```
Error on test data using gradient descent is: 185.362991279985
```

Figure 2.3: Error on test data using gradient descent

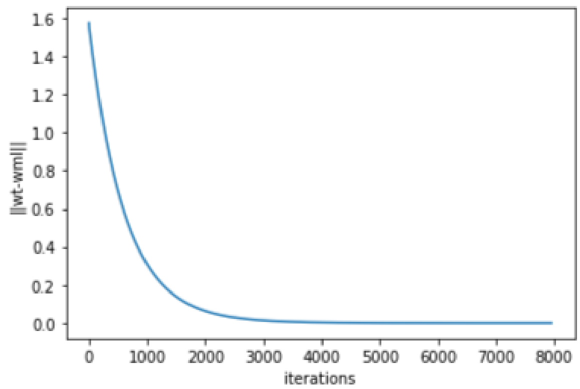


Figure 2.4: Plot for gradient descent

2.3 Stochastic Gradient Descent

The error on test data using stochastic gradient descent is less compared to gradient descent method (figure 2.5). Even though all the data points are not used together it still produces less error on this dataset. This is useful when the dataset itself is large as it helps to divide the dataset into different batches and then perform gradient descent on this batch and still it produces less error.The plot for stochastic gradient descent is shown in figure 2.6.

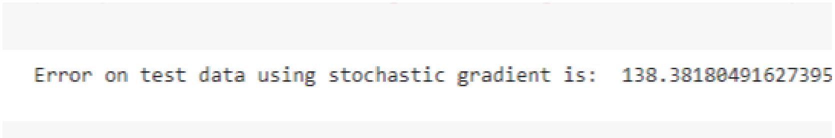

Error on test data using stochastic gradient is: 138.38180491627395

Figure 2.5: Error on test data using stochastic gradient descent

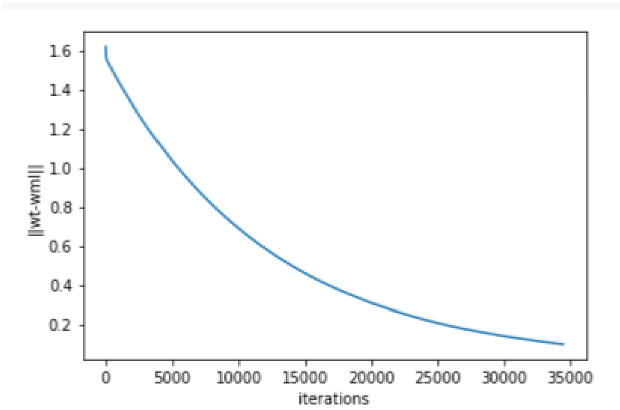


Figure 2.6: Plot using stochastic gradient descent

2.4 Gradient descent algorithm for ridge regression

The error is found to be less for the smallest eigen value of XXt after doing the cross validation by dividing the dataset into 80-20 percentage (figure 2.7.The plot for ridge regression for various choice of lambda is shown in figure 2.9. The error on test data using ridge regression (figure 2.8) is found to be less compared to other methods and also less compared to wml. This means that ridge regression performs best on the given dataset . This could possibly because of the fact that if there are lot of features and if they are correlated themselves then in case of ridge regression it pushes such features weights to be 0. This is visible in wr obtained from ridge regression. Most of the weights are closer to zero. Therefore ridge regression just takes a subset of features whose combination can clearly explain ‘y’ and pushes the rest to 0. i.e. it is an attempt to feature selection.

```
The error is minimum at index:-  99  for lambda= 685.5388004299396
```

Figure 2.7: Cross validation result

```
Error on test data is:  126.35205423940342
```

Figure 2.8: Error using ridge regression

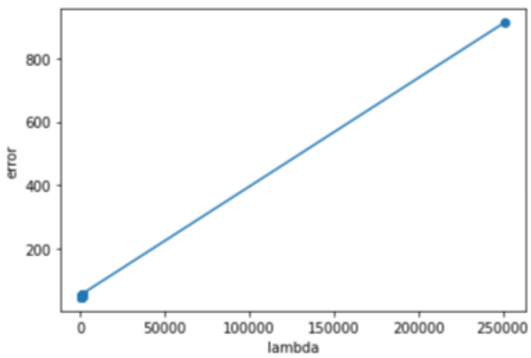


Figure 2.9: Plot for various choices of lambda