Stevens Institute of Technology

Department of Computer Science

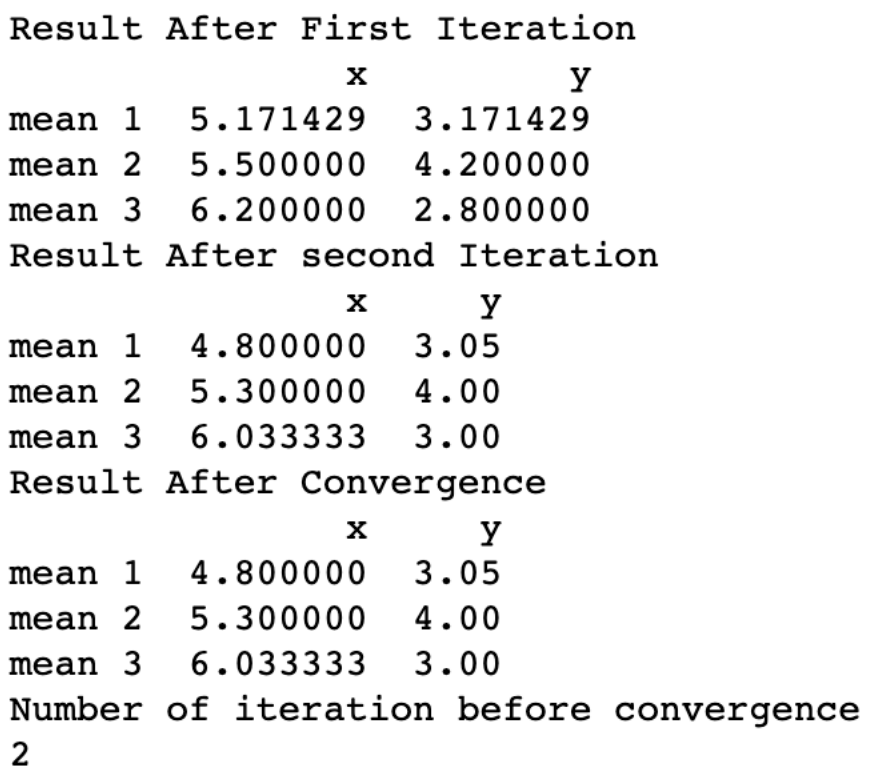
CS559: Machine Learning Fundamentals and Applications

Prof. Tian Han

Homework Assignment 3

Submitted by: Malav Shastri, CWID:10456244

Solution 1: (Jupyter Notebook Submitted)



```
Result After First Iteration
              x         y
mean 1  5.171429  3.171429
mean 2  5.500000  4.200000
mean 3  6.200000  2.800000
Result After second Iteration
              x     y
mean 1  4.800000  3.05
mean 2  5.300000  4.00
mean 3  6.033333  3.00
Result After Convergence
              x     y
mean 1  4.800000  3.05
mean 2  5.300000  4.00
mean 3  6.033333  3.00
Number of iteration before convergence
2
```

Solution 2:

1 . Compact form of p(z) and p(x|z).

Where 0 <= πk <= 1 and $\sum_{k=1}^{K} \boldsymbol{\pi_k} = \mathbf{1}$

Therefore ,

Compact Form of p(z):

$$p(z) = \prod_{k=1}^{K} \pi_k^{z_k} \qquad \text{(i)}$$

Compact Form of p(x|z)

$$p(x|z) = \prod_{k=1}^{K} \mathrm{N}(x|\mu_k, \Sigma_k)^{z_k} \qquad \text{(ii)}$$

2. Show the marginal distribution p(x) has the form:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

The marginal distribution of x is then obtained by summing the joint distribution over all possible values of z,
Compact Form of p(z):
$p(z) = \prod_{k=1}^{K} \pi_k^{z_k}$

Compact Form of p(x|z)
$p(x|z) = \prod_{k=1}^{K} \mathrm{N}(x|\mu_k, \Sigma_k)^{z_k}$

Therefore,

p(x) = $\boldsymbol{\sum_{k=1}^{K} \pi_k N}\ (x|\mu_k, \Sigma_k)$

3. MLE solution for parameters πk,μk,Σk and difference to K-Means.

MLE Solutions using discrete latent variable z : ,

ln p(X|π, μ, Σ) = $\sum_{n=1}^{N} \ln \{\boldsymbol{\sum_{k=1}^{K} \pi_k N}\ (x|\mu_k, \Sigma_k)\}$

For $\boldsymbol{\mu_k}$,
setting the derivative of ln p(X|π, μ, Σ) w.r.t μ_k to 0
0 = - $\sum_{n=1}^{N} \frac{\pi_k N(X_n|\mu_{k,}\Sigma_k)}{\sum_j \pi_j N\left(X\middle|\mu_{j,}\Sigma_j\right)} \Sigma_k (X_n - \mu_k)$
Where, $\sum_j \pi_j N\left(X\middle|\mu_{j,}\Sigma_j\right)$ = $\gamma(z_{nk})$
$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma\ (z_{nk}) x_n$
Where, $N_k = \sum_{n=1}^{N} \gamma\ (z_{nk})$

For $\boldsymbol{\sum_k}$,
setting the derivative of ln p(X|π, μ, Σ) w.r.t $\sum_k$ to 0,

$$\Sigma_k = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T$$

Where, $N_k = \sum_{n=1}^{N} \gamma\,(z_{nk})$

For $\boldsymbol{\pi_k}$,

We maximize ln p(X|π, μ, Σ) w.r.t. π_k,

By using langrage multiplier,

ln p(X|π, μ, Σ)+$\lambda(\sum_{k=1}^{K} \pi_k - 1)$

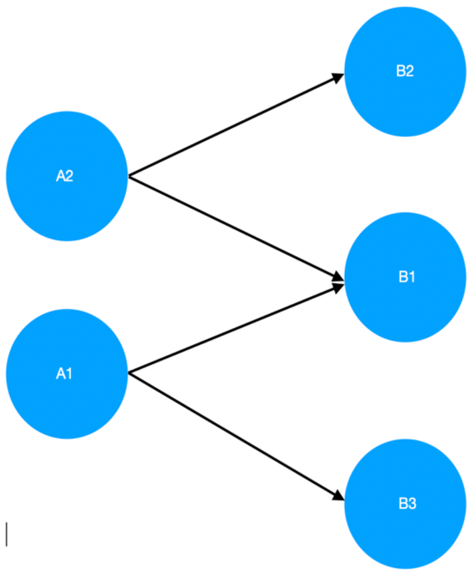
where, $\sum_k \pi_k = 1$

$$\pi_k = \frac{N_k}{N}$$

(P.T.O)

Solution 3:

Question 1: Draw the corresponding Bayesian network.



Question 2: Based on the Bayesian network in (1), write down the

joint distribution p(A1;A2;B1;B2;B3).

Solution:

Joint Distribution:

P(A1,A2,B1,B2,B3) = P(B1/A1,A2) * P(B3/A1) * P(B2/A2) * P(A1) * P(A2)

Question 3 : How many independent parameters are needed to fully specify the joint distribution in(2).

Solution:

Required independent parameters to fully specify the joint distribution of given bayesian network:

(P(B1/A1,A2)) + 2(P(B3/A1)) + 2(P(B2/A2)) + 1(P(A1)) + 1(P(A2)) = 10

Question 4 : Suppose we do not have any independence assumption, write down one possible factorization of p(A1;A2;B1;B2;B3), and state how many independent parameters are required to describe joint distribution in this case.

Solution:

1. Factorization :

(A1,A2,B1,B2,B3) =
P(A1)P(A2/A1)P(B1/A1,A2)P(B2/A1,A2,B1)P(B3/A1,A2,B1,B2)

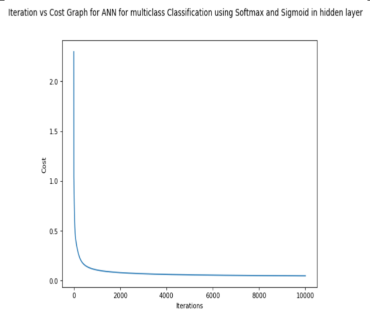
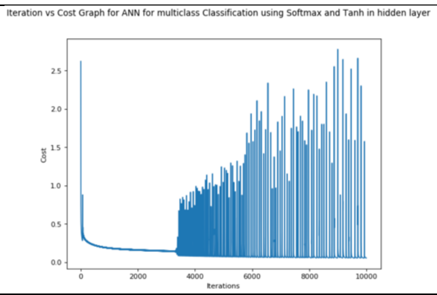
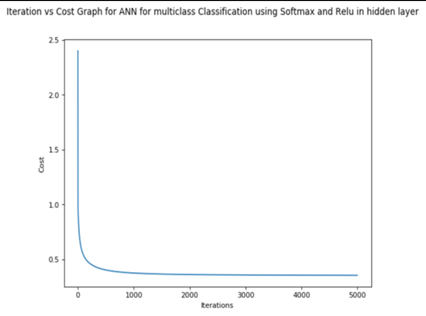
2. Required independent parameters to fully specify the joint Distribution in Bayesian Network:

1(A1) + 2(A2/A1) + 4 P(B1/A1,A2) + 8 P(B2/A1,A2,B1) + 16 P(B3/A1,A2,B1,B2) = 31

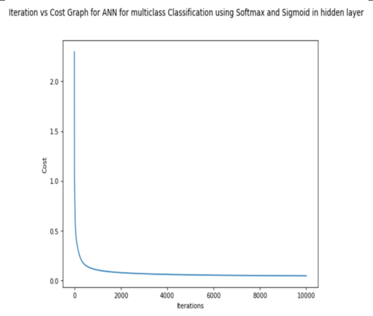
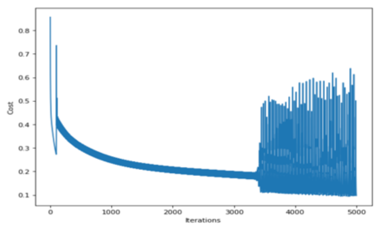
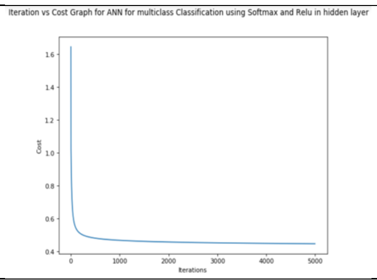
(P.T.O)

Solution 4: (Jupyter Notebook Submitted)

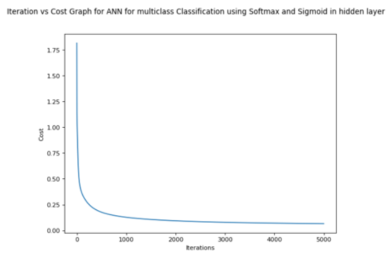
1) Iteration = 10000, learning rate = 0.01, hidden layer units = 6

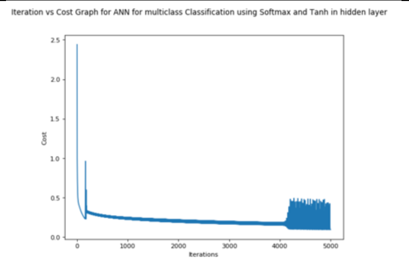
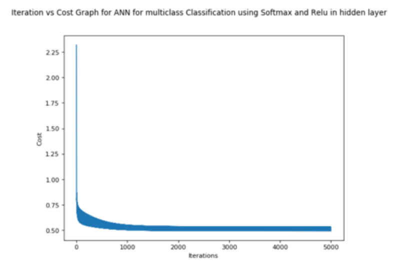
Activation Function in Hidden layer.	Accuracy	Cost vs Iteration
Sigmoid	92.10%	
Tanh	93.105%	
ReLu	79%	

2) Iteration = 5000, learning rate = 0.01, hidden layer units = 6

Activation Function in Hidden layer.	Accuracy	Cost vs Iteration
Sigmoid	97.36%	
Tanh	94.73%	
ReLu	52.63%	

3) Iteration = 10000, learning rate = 0.01, hidden layer units = 5

Activation Function in Hidden layer.	Accuracy	Cost vs Iteration
Sigmoid	97.36%	

Tanh	97.36%	
ReLu	60.52%	

General Observations:

1) Sigmoids have slow convergence.
2) In ReLu some gradients can be fragile during training and can die. It can cause a weight update which will makes it never activate on any data point again. Simply saying that ReLu could result in Dead Neurons it is also known as dying Relu problem.
3) ReLu Learns faster compares to Sigmoid and Tanh
4) Here one of the reasons why our ReLu model is not working properly is that Small values, even the non-positive ones, may be of value; they can help capture patterns underlying the dataset. With ReLU, this cannot be done, since all outputs smaller than zero are zero.
5) Here in the above tables for the given parameters as you can see that tanh function cost vs iteration graph is oscillating very much, the reason behind that is that derivative of tanh is much more steeper compared to sigmoid derivative and that's why in the training at last it is overshooting. Although at last the cost value is at the lowest so that means function is getting converged. This is one of the biggest differences between tanh and sigmoid. Process of convergence is faster in tanh compared to sigmoid as its gradience is very strong and so tanh is preferable compared to sigmoid. We can get rid of this fluctuation over some last iteration in tanh training by decreasing the learning rate, which is also displayed in one of the tables given here.
6) Here one more thing is also worth noting, as our dataset is very small, we are facing problem of overfitting on high iteration rates.
7) Here ideally ReLu should outperform other 2 but as the problems mentioned above and the size of our dataset it is under performing we can tune it with different parameters different test/ train data sizes and different learning rates to make it work better.

Conclusion: We can still try out different combination of hidden units, learning rate and activation functions and ca further tune these model respectively with different parameters as all of them behave differently under different circumstances also our dataset is very small that is also a big problem we are training on a size of 112 records and predicting for just 38 so this is also a worth noting factor.