1 problem 1

My classifier performs well on both fake dataset and MNIST dataset and achieved the desired accuracy.

2 problem 2

For fake dataset, I got 'bij': array([[0.05685779, 0.25532323], [0.97445933, 0.8597117]]), 'bjy': array([0.50747444, 0.49252556])

when k=5
Training Accuracy: 0.8659
Testing Accuracy: 0.86660000000000004

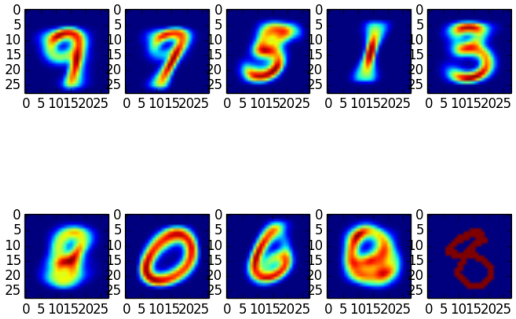

Figure 1: k=5

3 problem 3

when k=1
Training Accuracy: 0.8081333333333334
Testing Accuracy: 0.8054

when k=10
Training Accuracy: 0.89516666666666667
Testing Accuracy: 0.89639999999999997

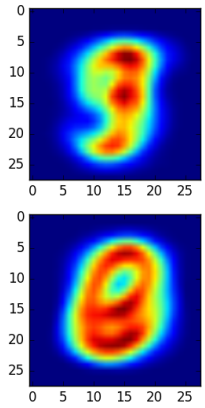

Figure 2: k=1

when k=20
Training Accuracy: 0.9231166666666667
Testing Accuracy: 0.9233

As K increases, the accuracy also increases. This is because the added latent variables helps the classifiers to classify the data. We can assume if we keep increase K, the performance would reach a plateau because at that point not all the latent variables would be useful and they would cause more overfitting.

4 Problem 4

We can approach this task using gradient descent. We would perform MLE through writing out the likelihood function, taking the derivative of the likelihood function and deriving the parameter value when the derivative is set to 0. The disadvantage for gradient descent is that we have to make sure the likelihood function is differentiable and taking derivative is computationally expensive.

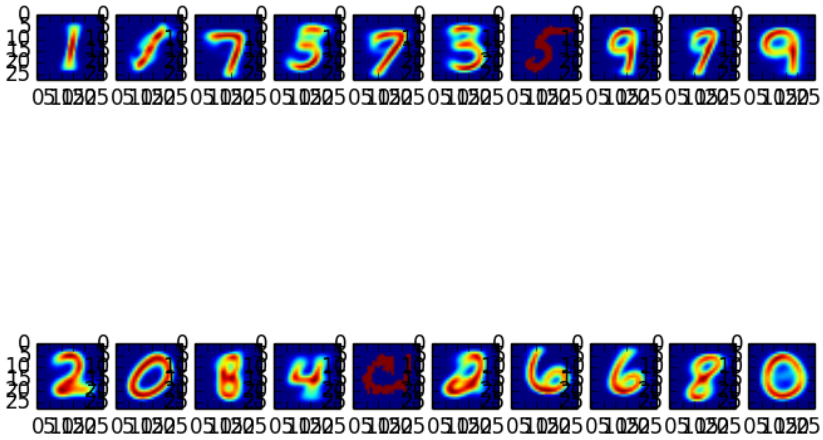

Figure 3: k=10

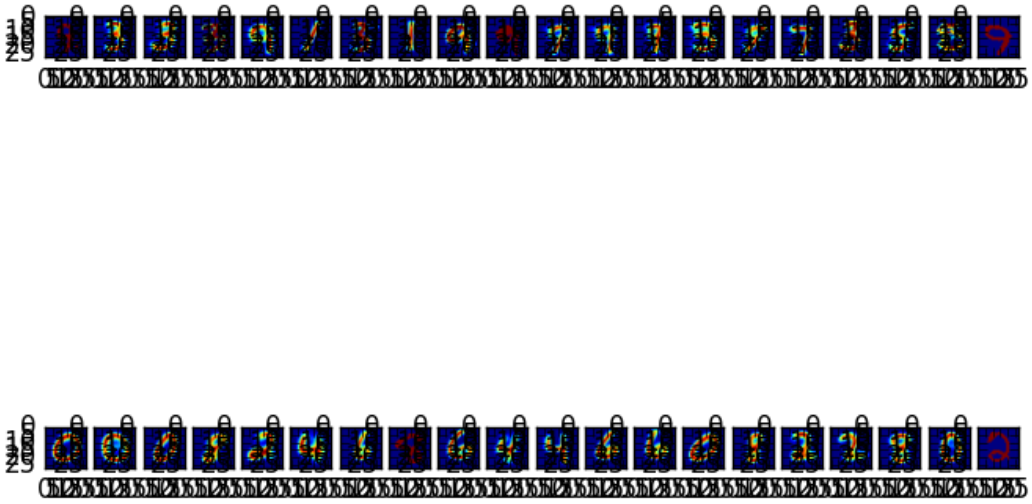

Figure 4: k=20