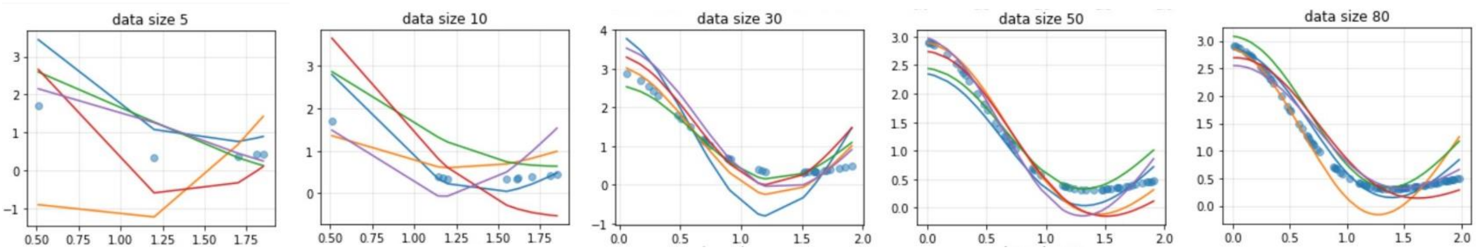
Machine Learning

Homework 2

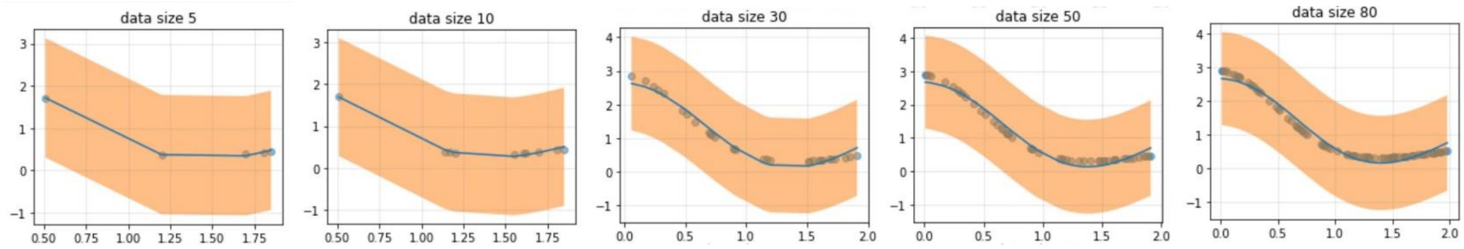
309551059 羅文慧

- ## Sequential Bayesian Learning

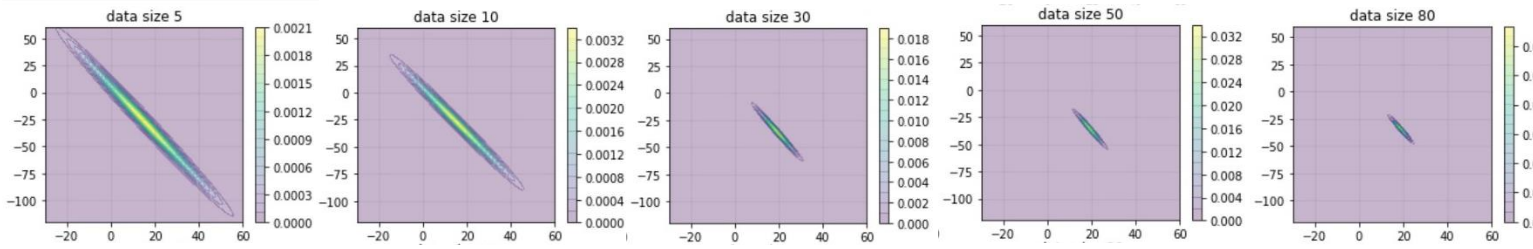
1. Plot five curves sampled from the parameter posterior distribution and N data points.



2. Plot the predictive distribution of target value t by showing the mean curve, the region of variance with one standard deviation on both sides of the mean curve and N data points.



3. Plot the prior distributions by arbitrarily selecting two weights.



4. Make some discussion on the results of different N in 1, 2 and 3.

Randomly choose 5 weights from the distribution made by m_N and S_N, and use these weights to predict their corresponding y. From the figures in (1), we can observe that the curve is more smooth when the data size increasing.

By calculating mean and standard deviation, we can plot figure (2). Mean curve is also more smooth when the data size increasing.

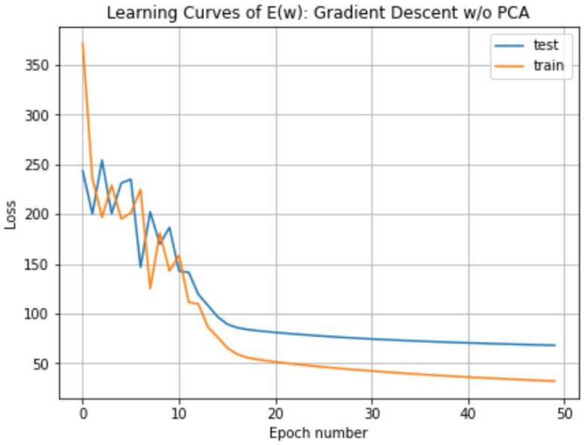
In figure (3), I choose w0 and w1 to plot the contour figure. As data size increasing, the region of prior distribution decreases.

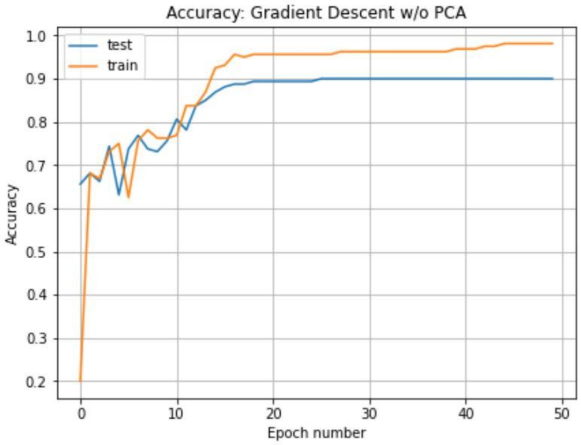
- ## Logistic Regression

1. **Set the initial weight vector wk = [wk1, . . . , wkF] to be a zero vector where F is the number of features and k is the number of classes. Implement batch GD, SGD, mini-batch SGD (batch size = 32) and Newton-Raphson algorithms to construct a multiclass logistic regression.**

   In this section, we use a subset of Fashion-MNIST, which includes 5 classes each containing 64 images. The dataset is randomly split into training:testing = 1:1. Each pixel is normaized between 0 and 1, and the label is preprocessed into one-hot encoding.

   (a) Plot the learning curves of E(w) and the accuracy of classification versus the number of epochs until convergence for training data as well as test data.

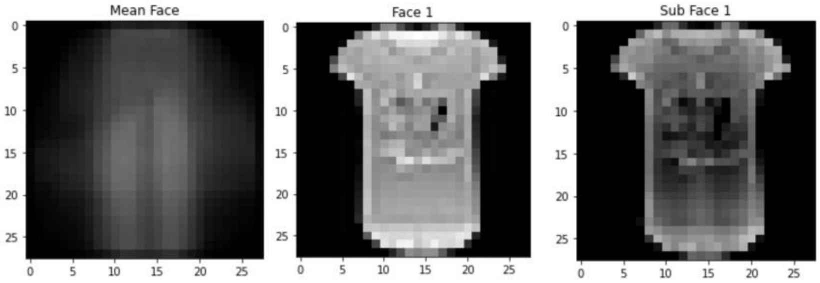




   (b) Show the classification results of training and test data.

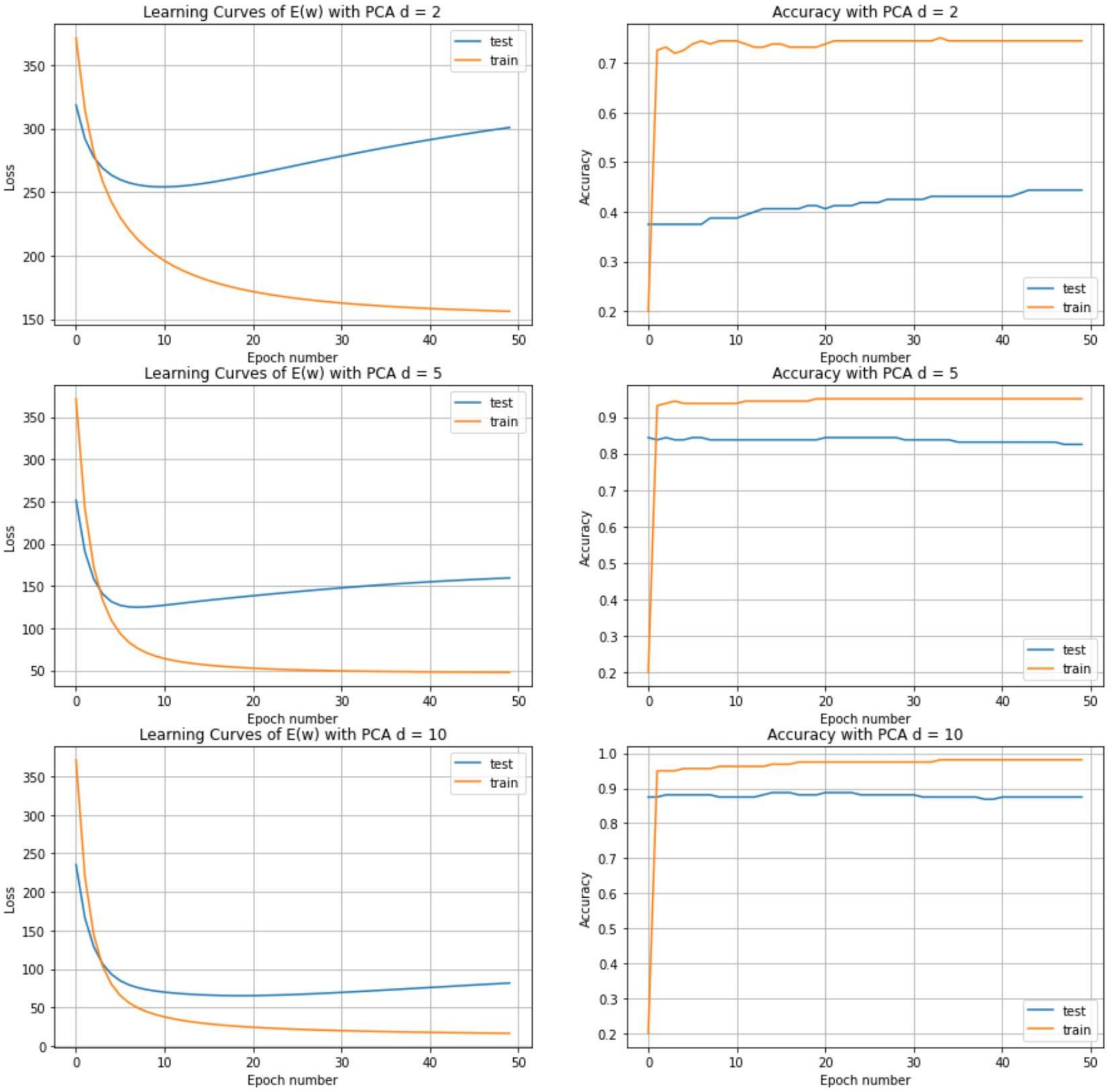
```
TYPE: Gradient Descent w/o PCA
Training Accuracy: 0.98125
Test Accuracy: 0.9
```

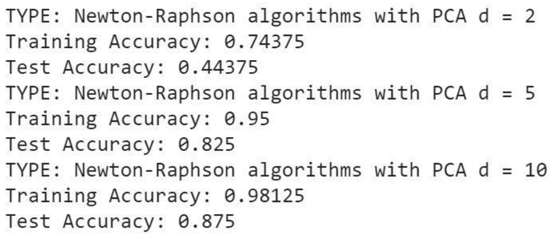
2. **Use principal component analysis (PCA) to reduce the dimension of images to d = 2, 5, 10.**



   Before using PCA, calculate pixel-wise mean to the training dataset. And plot the "mean face", which represents the pixel-wise density in dataset. The above pictures show the image before/after subtracts the mean face.

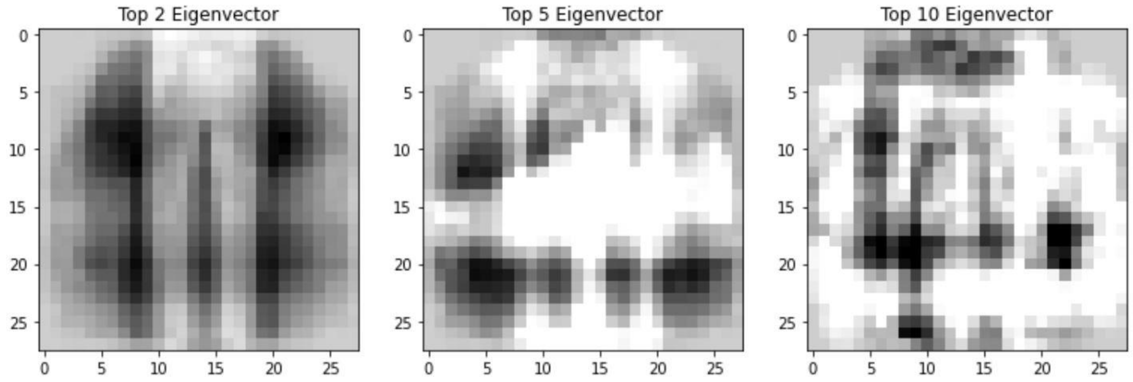
(a) Repeat 1 by using PCA to reduce the dimension of images to d.





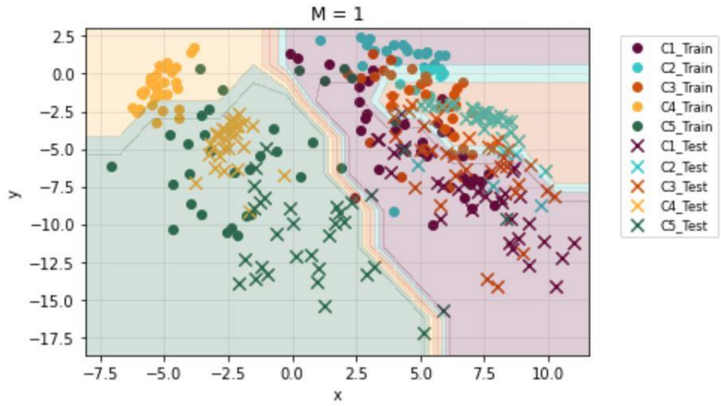
```
TYPE: Newton-Raphson algorithms with PCA d = 2
Training Accuracy: 0.74375
Test Accuracy: 0.44375
TYPE: Newton-Raphson algorithms with PCA d = 5
Training Accuracy: 0.95
Test Accuracy: 0.825
TYPE: Newton-Raphson algorithms with PCA d = 10
Training Accuracy: 0.98125
Test Accuracy: 0.875
```

(b) Plot d eigenvectors corresponding to top d eigenvalues.

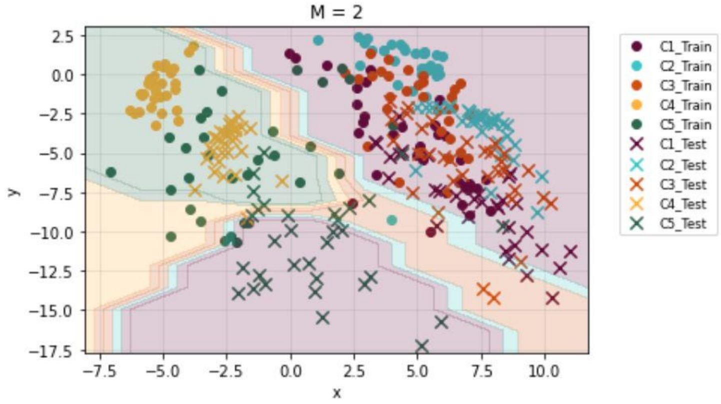


3. What do the decision regions and data points look like on the vector space?

(a) Plot the decision regions and data points of the images on the span of top 2 eigenvectors by using PCA to reduce the dimension of images to 2.



(b) Repeat 3(a) by changing the order from M = 1 to M = 2.



4. **Make some discussion on the results of 1, 2 and 3.**

   From figure (1), the learning curves of both train/testing in first 10 epochs oscillate intensely, then the learning curve of testing tend to converge but always higher than the learning curve of training. The accuracy curves of both train/testing also oscillate intensely in first 12 epochs, then the accuracy curve of training tend to converge but always higher than the accurcy curve of testing.

   From figure (2), the learning curves of training tend to converge when epoch increase, while the testing ones only drop in early stages and don't converge. It may due to the training dataset in the scenario is too small which cause overfitting problem. The accuracy curve improves significantly from d=2 to d=5, but there's no huge improvement from d=5 to d=10, which may due to the first few components provide more information than the last ones.

   From figure (3), the decision regions in M=1 seems more reasonable than M=2.