ISYE 6740/CSE 6740/CS 7641: HW 6

90 Points Total v1.0

Due: 11:59am Dec. 4

Name: Sandesh Adhikary

GT ID: 903303329

GT Account: sadhikary6

Instruction: Please write a report including answers to the questions and the plotted figures. Please write the code in `MATLAB` and submit your code in a **'zip' file** via T-Square. You are only allowed to use existing package/library when the question suggests so. Your implementation should based on the starting code given. You can adjust it if needed and you can also add auxiliary functions. Your code is supposed to have explanatory comments.

1) Principle Component Analysis (30 points) In this problem, you will implement PCA by power's method and stochastic gradient algorithm. The starting code is provided in folder `Q1`. The file `Q1_main.m` should be able to walk you through the question. You should use the hyper-parameters provided in the code to answer the question. Feel free to play with the hyper-parameters, but be sure to change them back for the sake of grading script. The data file `Q1.mat` contains the true covariance matrix `S` (for generating data) and the generated data matrix `X`. Denote the leading eigenvector of `S` as w^*. Denote the estimator of w^* (based on `X`) as $\widehat{w}$.

(a) Get w^*. You should use `S`. You can use any MATLAB build-in functions if you need. (5 points)

```
v_real =

    0.0615
    0.2472
    0.4390
    0.1575
   -0.0048
    0.7214
    0.2601
    0.0540
    0.3556
   -0.0068
```

Figure 1: w^*

(b) Get $\widehat{w}$. You should use data `X`. You can use any MATLAB build-in functions if you need. (5 points)

```
v_est =

    0.0602
    0.2440
    0.4384
    0.1579
   -0.0052
    0.7203
    0.2594
    0.0491
    0.3621
   -0.0081
```

Figure 2: $\hat{w}$

(c) Recall that the power's method starts from an arbitrary vector v_0, then performs update

$$v_{k+1} = \frac{Av_k}{||Av_k||}$$

where A is the sample covariance matrix. Implement power iteration. Use $[1/\sqrt{d}, ..., 1/\sqrt{d}]$ as initialization, where d is the dimension. Plot the figure of $\log(\|w^* - v_k\|_2^2)$ and the figure of $\log(\|\widehat{w} - v_k\|_2^2)$ versus the number of iterations. (10 points)

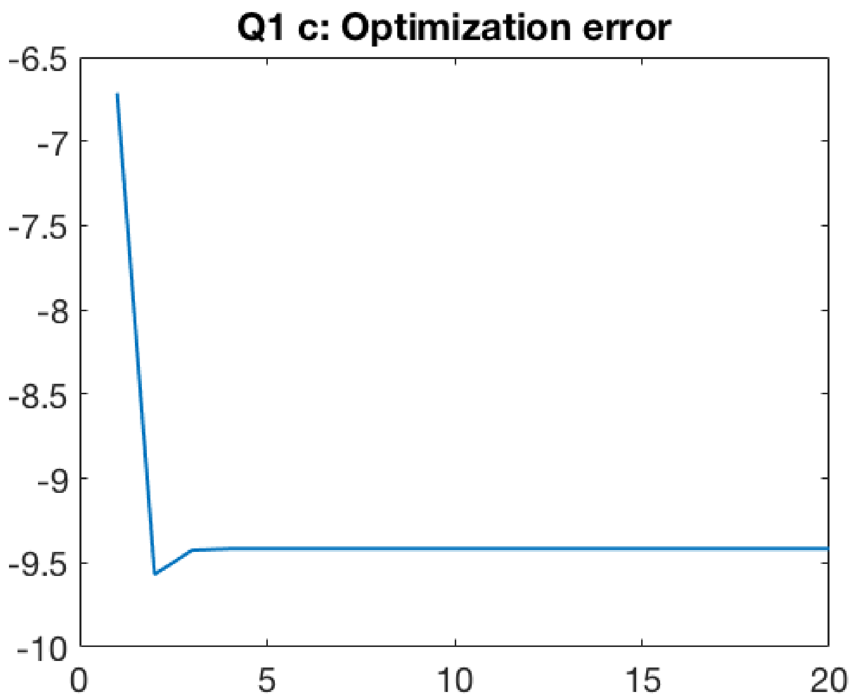

Figure 3: $\log(\|w^* - v_k\|_2^2)$ vs number of iterations.

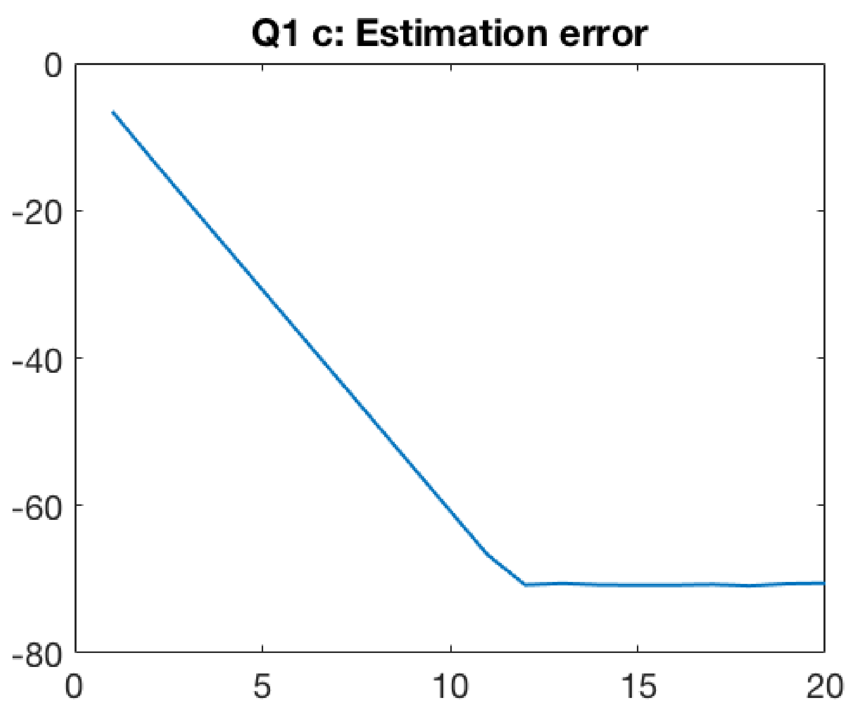

Figure 4: $\log(\|\widehat{w} - v_k\|_2^2)$ vs number of iterations.

(d) Recall that the stochastic gradient algorithm also starts from an arbitrary vector v_0, then performs update

$$v_{k+1} = \frac{v_k + \eta A_k v_k}{\|v_k + \eta A_k v_k\|},$$

where $A_k = \frac{1}{2}(X_i - X_j)(X_i - X_j)^\top$, and X_i and X_j are two data points you randomly sampled in each iteration. Please prove that A_k is an unbiased estimator of the sample covariance matrix. Use $[1/\sqrt{d}, ..., 1/\sqrt{d}]$ as initialization, where d is the dimension. Plot the figure of $\log(\|w^* - v_k\|_2^2)$ and the figure of $\log(\|\widehat{w} - v_k\|_2^2)$ versus iteration number. (10 points)

Hint: The eigenvectors are not unique. If w is an eigenvector of A, then $-w$ also is. To generate meaningful plots of (c) and (d), you may want to flip the sign of w^* or $\widehat{w}$.

Submission Requirement: For report, please include

(a) vector w^*;

(b) vector $\widehat{w}$;

(c) the figure of $\log(\|w^* - w^{(k)}\|_2^2)$ and the figure of $\log(\|\widehat{w} - w^{(k)}\|_2^2)$ with title "`Q1 c: Optimization error`" and "`Q1 c: Estimation error`", respectively;

(d) the figure of $\log(\|w^* - w^{(k)}\|_2^2)$ and the figure of $\log(\|\widehat{w} - w^{(k)}\|_2^2)$ with title "`Q1 d: Optimization error`" and "`Q1 d: Estimation error`", respectively, and the proof of unbiased estimator,

For code, put everything under the folder "`Q1`".

$\eta = 0.001$

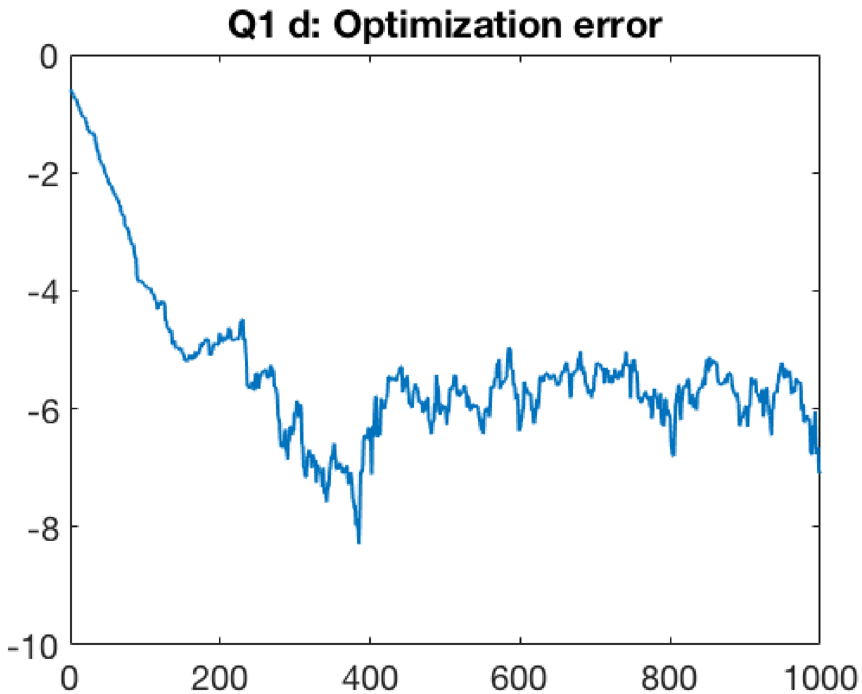

Figure 5: $\log(\|w^* - v_k\|_2^2)$ vs number of iterations with $\eta = 0.001$

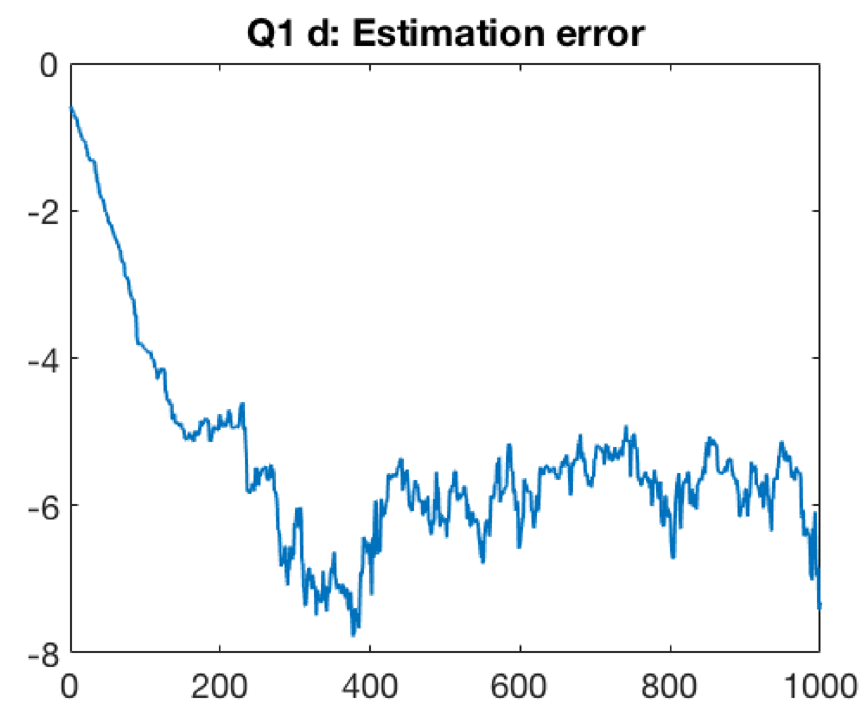

Figure 6: $\log(\|\hat{w} - v_k\|_2^2)$ vs number of iterations with $\eta = 0.001$

$\eta = 1/k$

We, now perform the algorithm again, but this time we set $\eta = 1/k$ where k is the iteration count. Therefore, η decreases with each iteration.

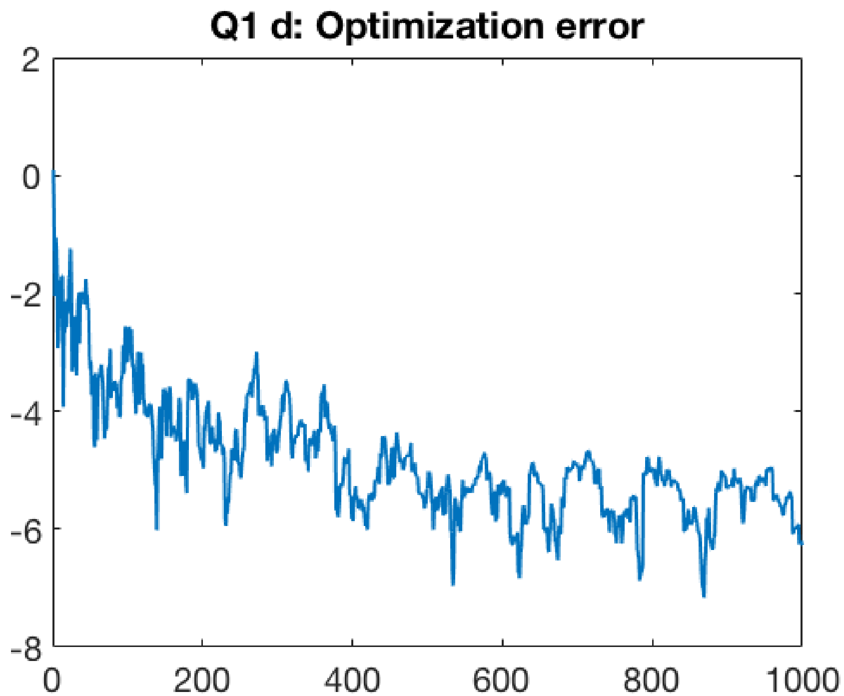

Figure 7: $\log(\|w^* - v_k\|_2^2)$ vs number of iterations with $\eta = 1/k$ where k is the iteration count.

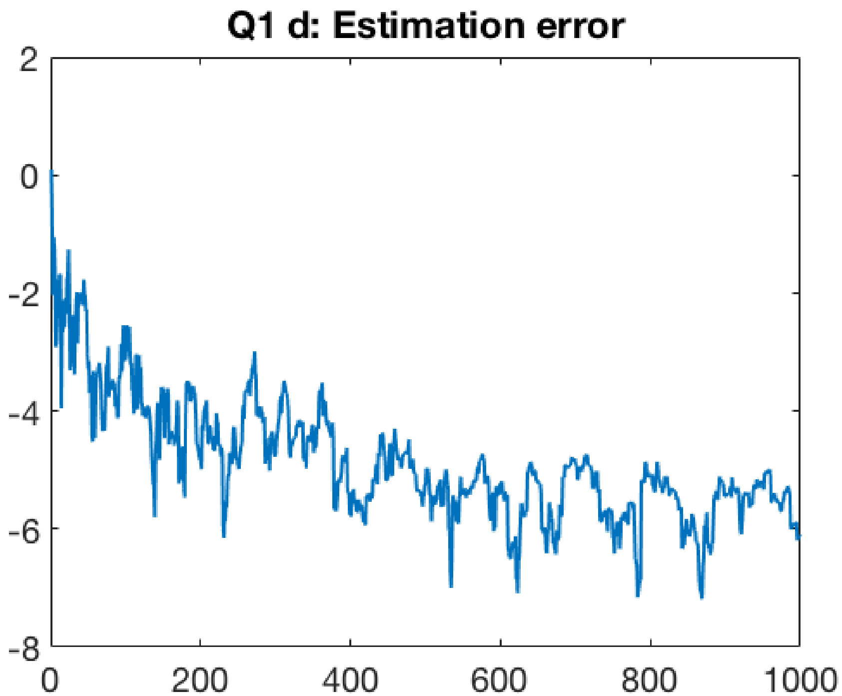

Figure 8: $\log(\|\hat{w} - v_k\|_2^2)$ vs number of iterations with $\eta = 1/k$ where k is the iteration count.

2) K-means Clustering (30 points) Implement K-means clustering based on the code provided in folder `Q2`. Two figures should be generated when you run `Q2_main.m` after your implement. Your report should include the figures for K=2, 3, and 4.

Submission Requirement: For report, please include the six figures generated.

For code, put everything under the folder "`Q2`".

ANSWER

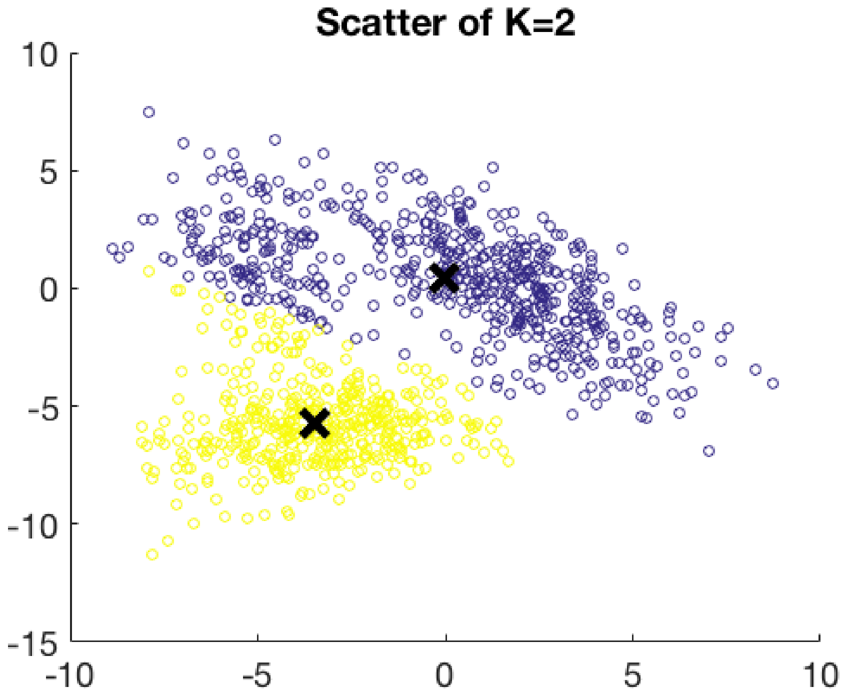

Figure 9: Scatter plot for K = 2

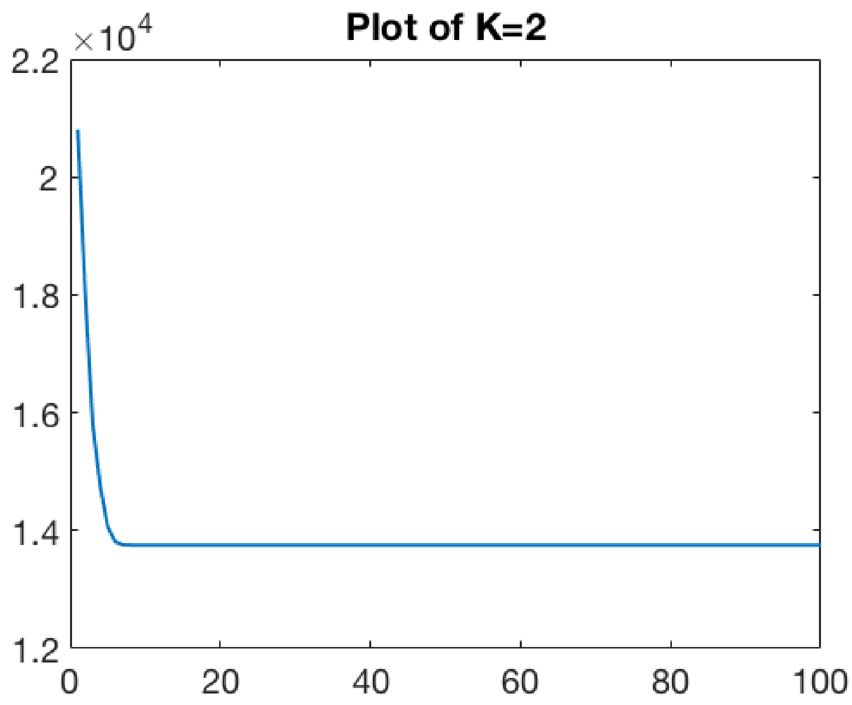

Figure 10: Loss plot for K = 2

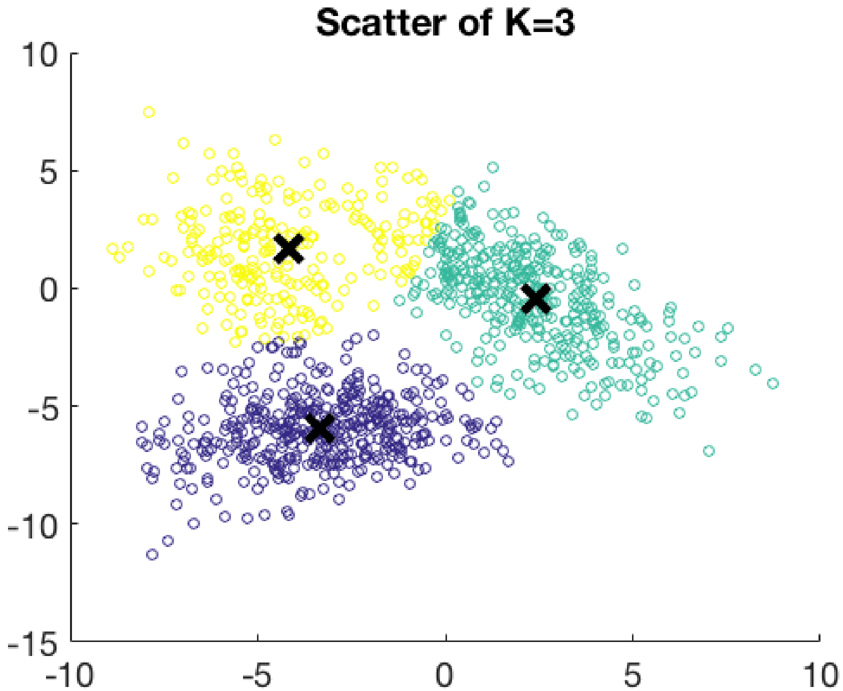

Figure 11: Scatter plot for K = 3

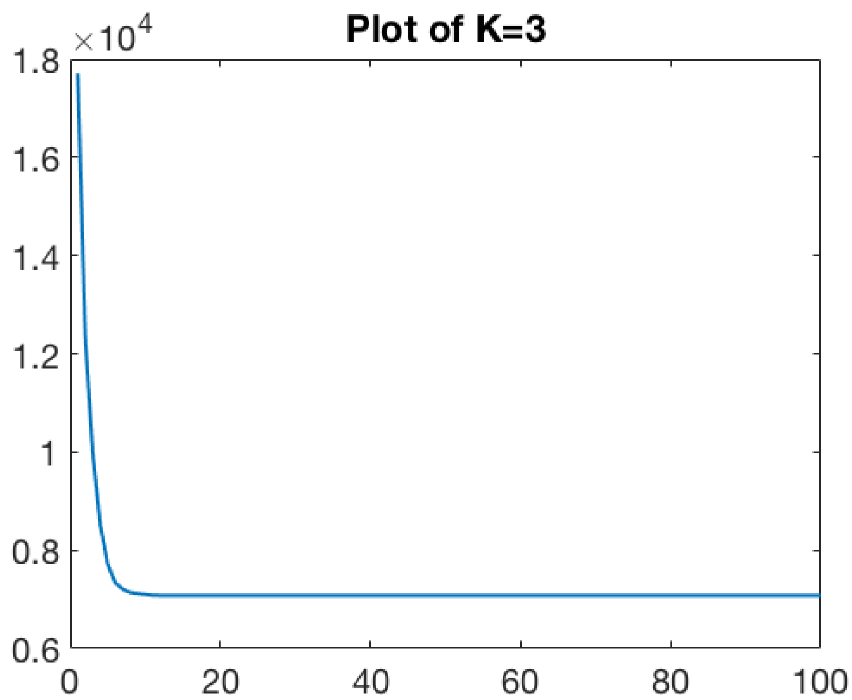

Figure 12: Loss plot for K = 3

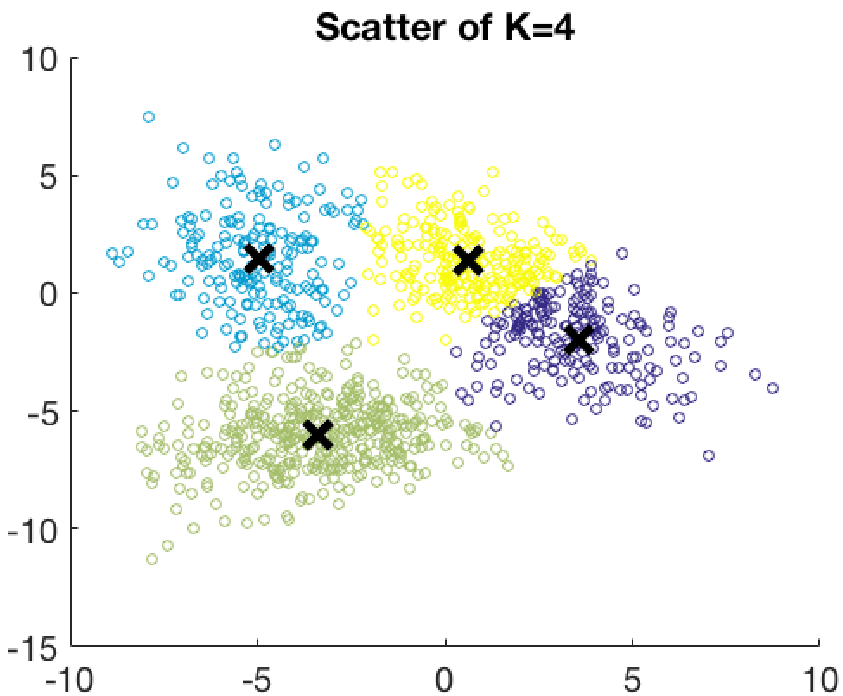

Figure 13: Scatter plot for K = 4

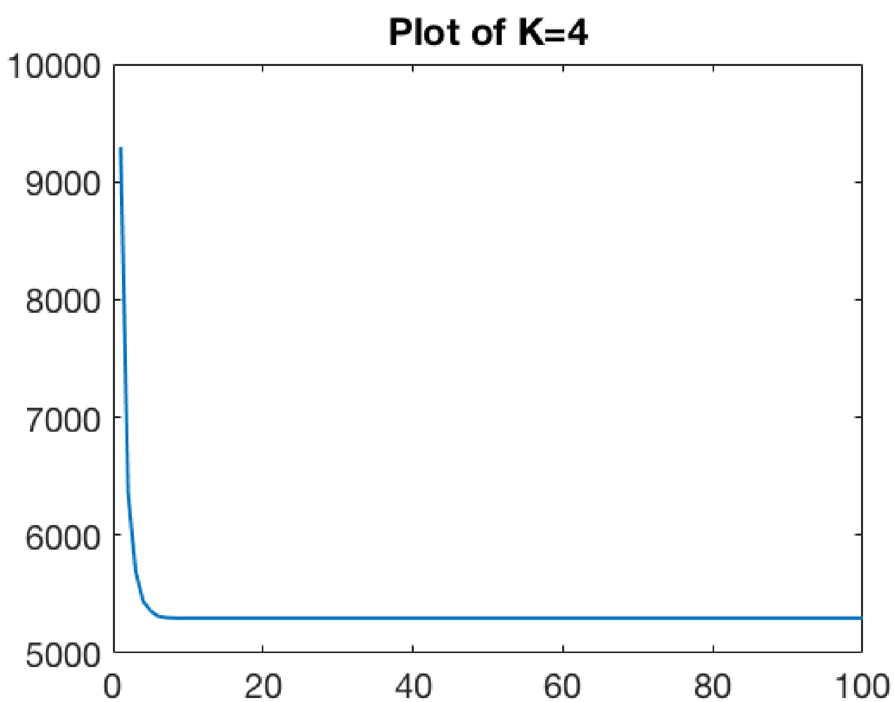

Figure 14: Loss plot for K = 4

3) Adaboost with Decision Tree (30 points) In this problem, you will implement Adaboost and random forest. The file `Q3_main.m` should be able to walk you through the question. You are allowed to use any MATLAB build-in functions.

(a) Implement an Adaboost with decision stump. Plot the ROC curve. (15 points)

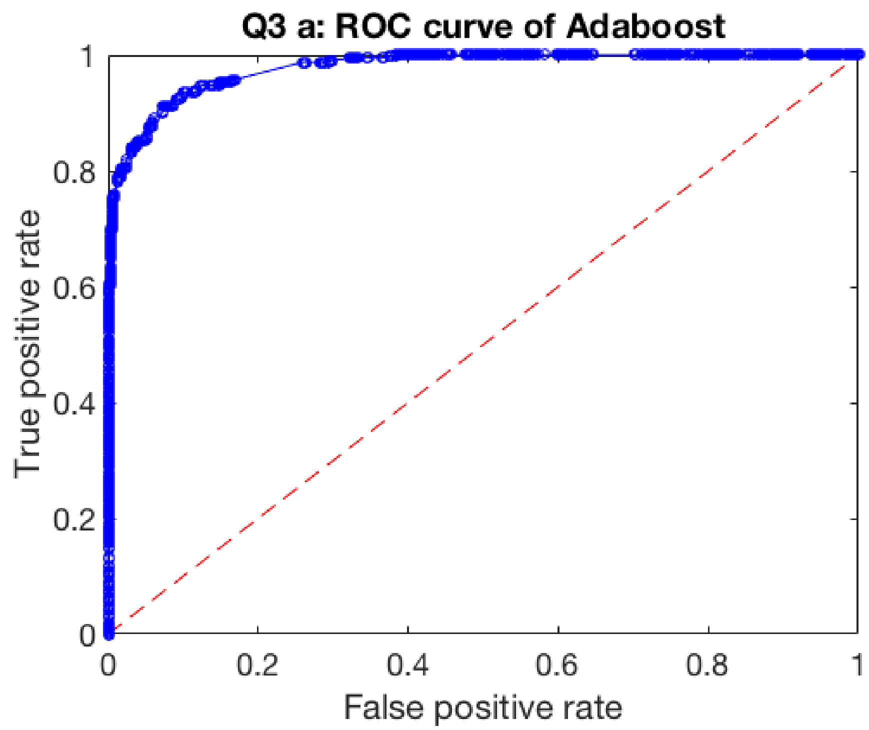

Figure 15: Adaboost using decision stumps. NumTrained = 100.

(b) Implement a random forest with decision tree. Plot the ROC curve. (15 points)

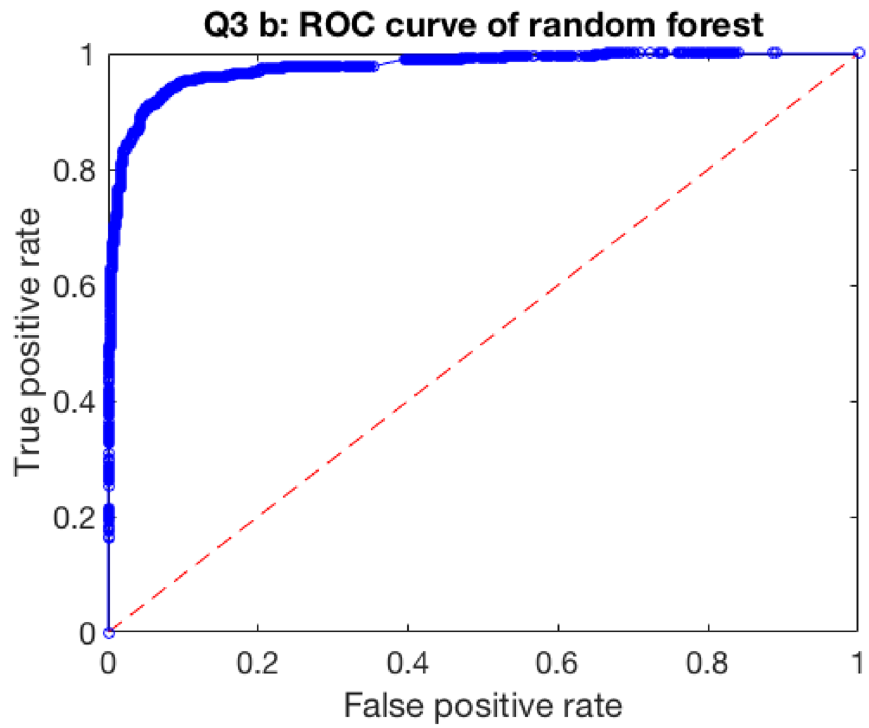

Figure 16: Random Forest using 100 trees.

Submission Requirement: For report, please include

(a) the ROC plot with title "`Q3 a: ROC curve of Adaboost`";

(b) the ROC plot with title "`Q3 b: ROC curve of random forest`".

For code, put everything under the folder "`Q3`".

Structure of your submission

```
Your submission
├── HW6_report_<Your-last-name>_<Your-first-name>.pdf
└── HW6_code_<Your-last-name>_<Your-first-name>.zip
    ├── Q1
    │   ├── Q1_main.m
    │   ├── oja_method.m
    │   ├── power_method.m
    │   └── Other auxiliary functions you might need
    ├── Q2
    │   ├── Q2_main.m
    │   ├── myKmeans.m
    │   └── Other auxiliary functions you might need
    └── Q3
        ├── Q3_main.m
        ├── myROC.m
        └── Other auxiliary functions you might need
```