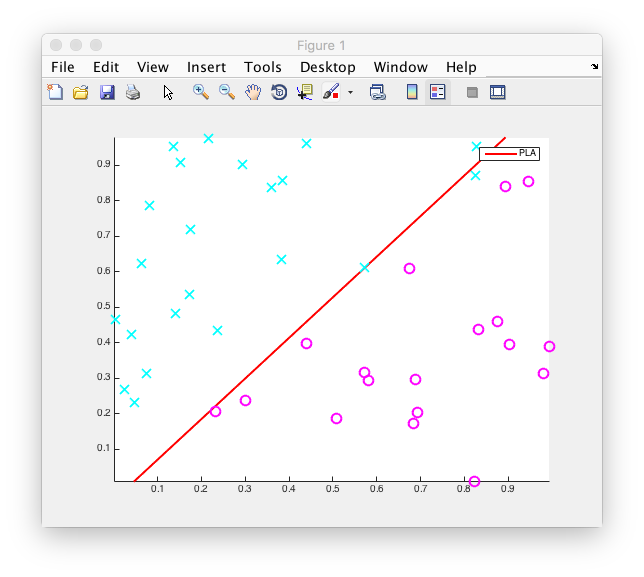
CS5402 Assignment 3

Wei Luo

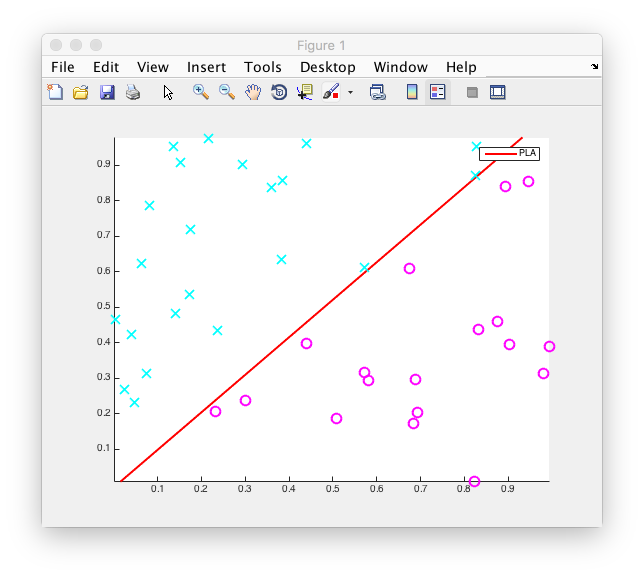
1. **Implement your Linear Regression algorithm on this dataset and obtain your W\_*LinearRessgion*. Apply your W\_*LinearRessgion* to the training dataset and compute your error rate.**



The error measure err\_sqr = 10.8583, it still has 1 error for classification and the error rate is 0.025.

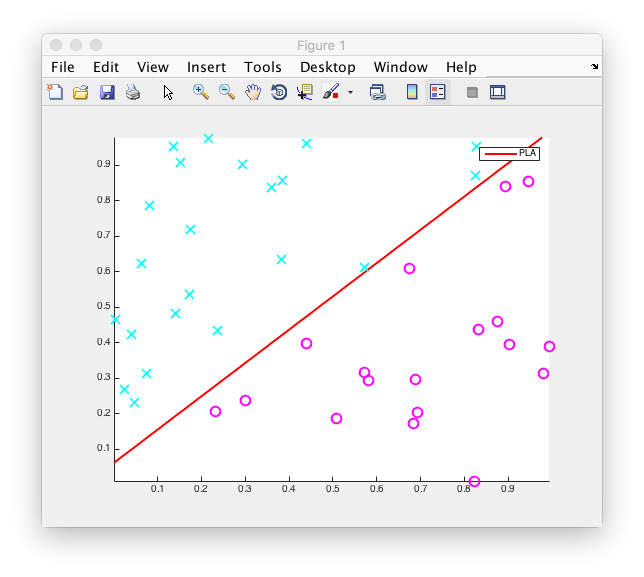
1. **Implement your Logistic Regression algorithm on this dataset and obtain your W\_*LogisticRessgion*. Apply your W\_*LogisticRessgion* to the training dataset and compute your error rate.**

First, I set the initial W\_*LogisticRessgion* as [0; 0; 0], the step size is 0.1 and the total iteration T = 3000.



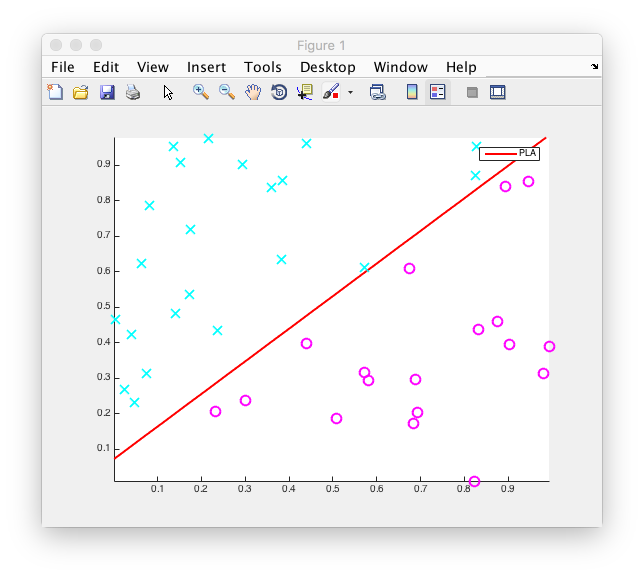
Actually, when it iterates 2313 times of all the dataset, the error rate of classification equal to 0. The final error measure err\_ce = 6.4836 and the error rate is 0.

Then I set the step size to 0.5. It is obviously that the change of W\_*LogisticRessgion* at beginning time became larger.

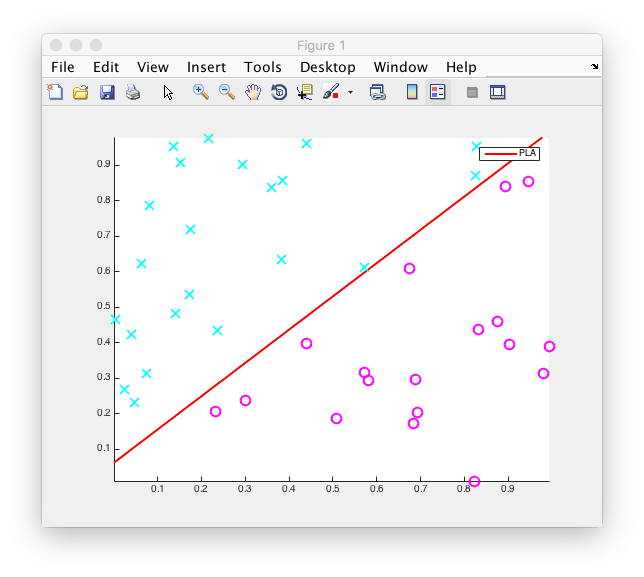


The time we need to make classification error rate to 0 is only 463. The error measure is 3.5317.

Then I set the step size to 1. The final error measure is 2.6389. The error rate is still 0.

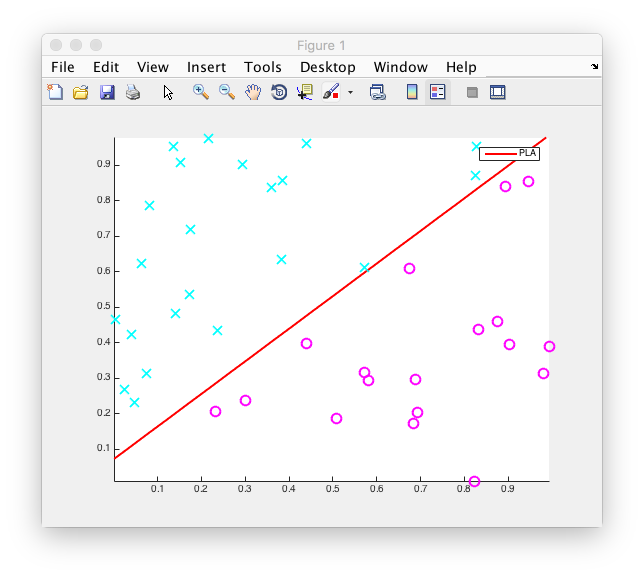


Second, I set the initial W\_*LogisticRessgion* as the W\_*LinearRessgion*, and the step size is 0.5.



The time we need to minimize the classification error rate to 0 is 378. The error measure is similar with the first one, err\_ce = 3.5048.

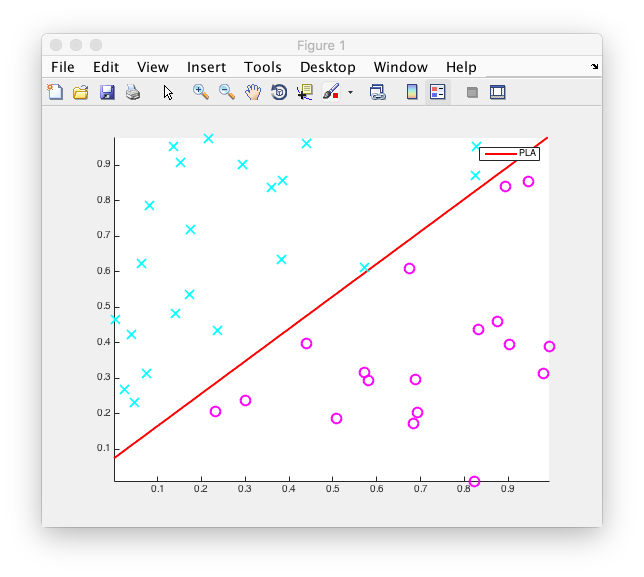
Then I set the step size to 1.



The time we need to minimize the classification error rate to 0 is only 189. The error measure is also similar with the first one, err\_ce = 2.6278.

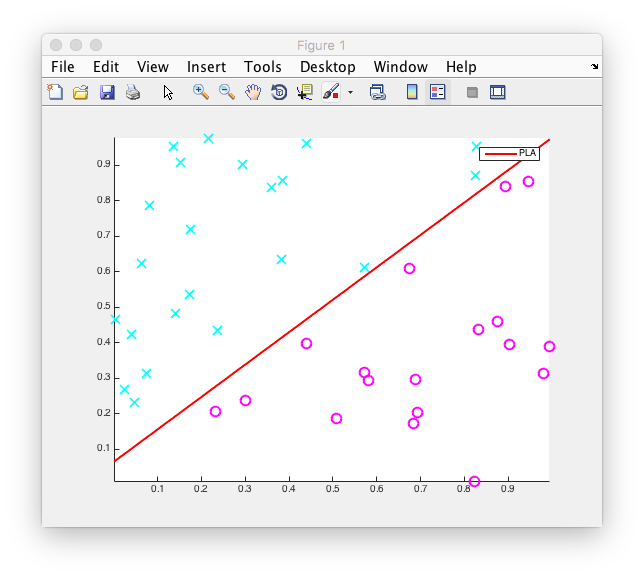
1. **Implement your Logistic Regression algorithm with Stochastic Gradient Descent (SGD) on this dataset and obtain your W\_*LogisticRessgionSGD*. Apply your W\_*LogisticRessgionSGD* to the training dataset and compute your error rate.**

Firstly, I set the initial W\_*LogisticRessgionSGD* as [0; 0; 0] and the step size is 0.1, the total iteration is 1000, the number of data we get from the dataset in one time is 40.



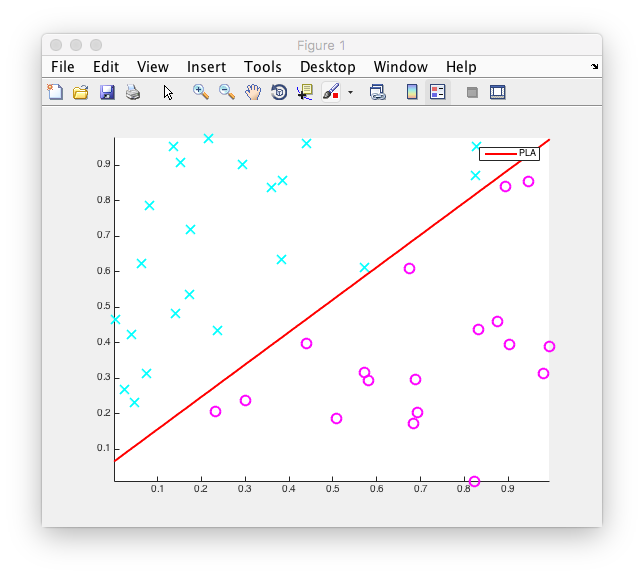
The classification error rate is 0, error measure is 3.3273.

Then I change the step size to 0.5.



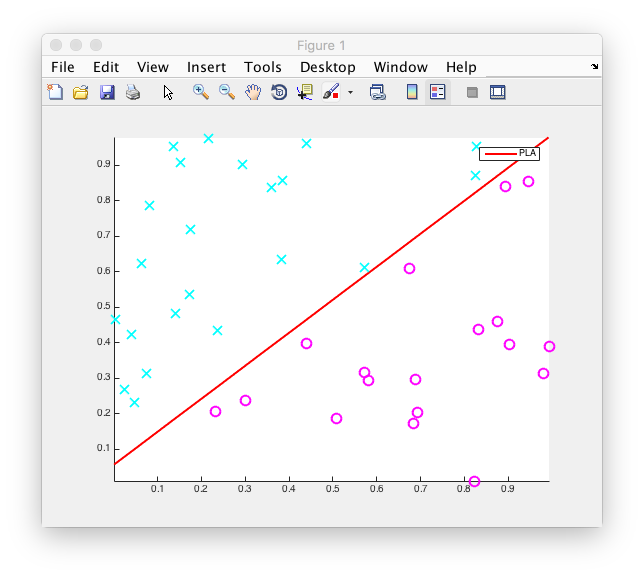
The error rate is 0 and the error measure is 1.4286.

Secondly, I change the initial W\_*LogisticRessgionSGD* to W\_*LinearRegression*, and the step size is 0.5.



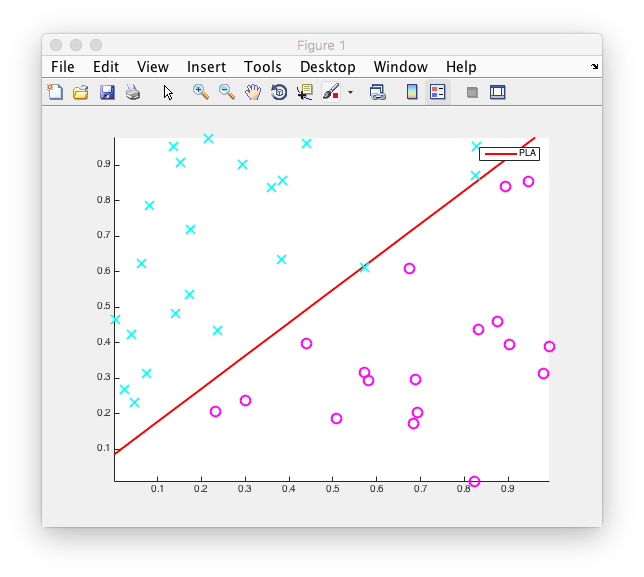
The classification error rate is 0, error measure is 1.4277, so similar with the first one. It seems that this initial W\_*LogisticRessgionSGD* might get the better result.

Then I change the step size to 1.



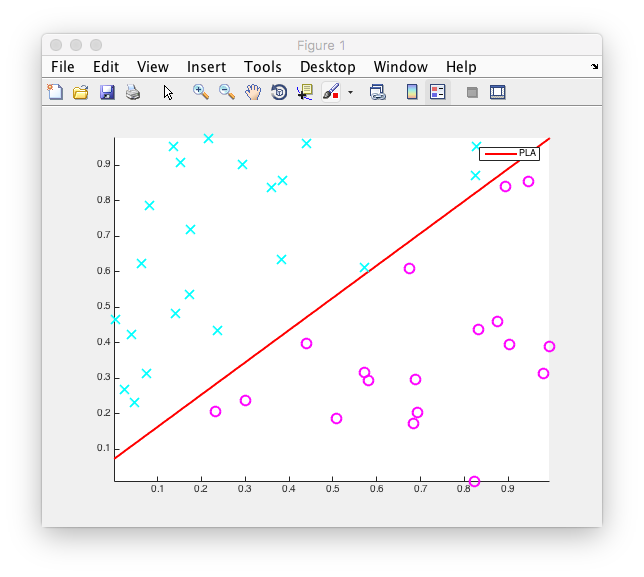
The classification error rate is 0, error measure is 0.9032.

Thirdly, I change the number of data we get from one time to 1, and the total number of iteration change to 40000 (1000 \* 40), the step size to 0.1, the initial W\_*LogisticRessgionSGD* to [0; 0; 0].



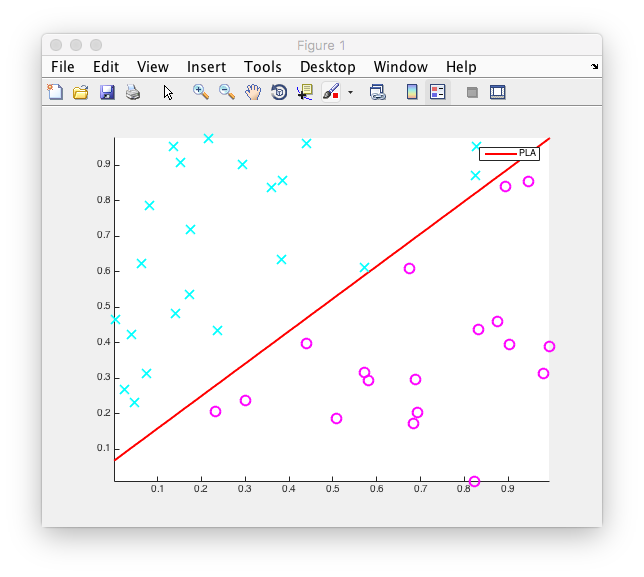
The error rate is 0.025. And the error measure is 3.5372. The result is obviously not as good as the first experiment which we set the number of data we get from one time is 40. Because in the first experiment every dataset has same opportunity to be trained.

Then I set the initial W\_*LogisticRessgionSGD* to W\_*LinearRegression*, and the step size to 0.5.



In this time, the result become better. The error rate turns to 0, and the error measure is 1.4428. The result is not as good as the experiment we set the number of dataset we get in a time as 40, but it is acceptable.

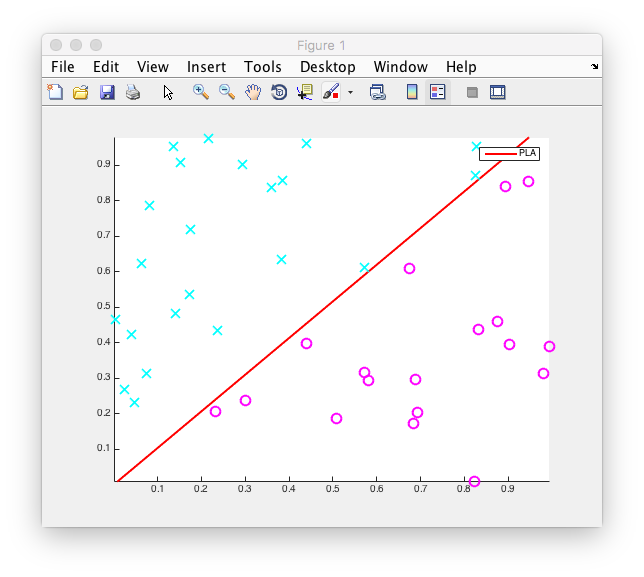
Finally, I change the number of dataset we get in a time to 5. And the total iteration change to 8000 (40000 / 5).



The result becomes a little better, the error measure smaller to 1.4247. Because the training dataset is randomly selected from the whole dataset, the result might change every time. Besides, the total number of dataset is only 40, which I believe is not enough to judge how many random data we get to train in a time is better.

**Compared with PLA algorithm**

In the PLA algorithm, we set the initial w with [0; 0; 0] the total iteration is only 5 (actually is 4, the last iteration is to make sure that no more error for this w), to turn the error rate to 0.



Then we change to use W\_*LinearRegression* as initial w. The total iteration is 2 and of course the w we get is different with the first one, but the error rate is 0.