EECS 349 (Machine Learning) Homework 6

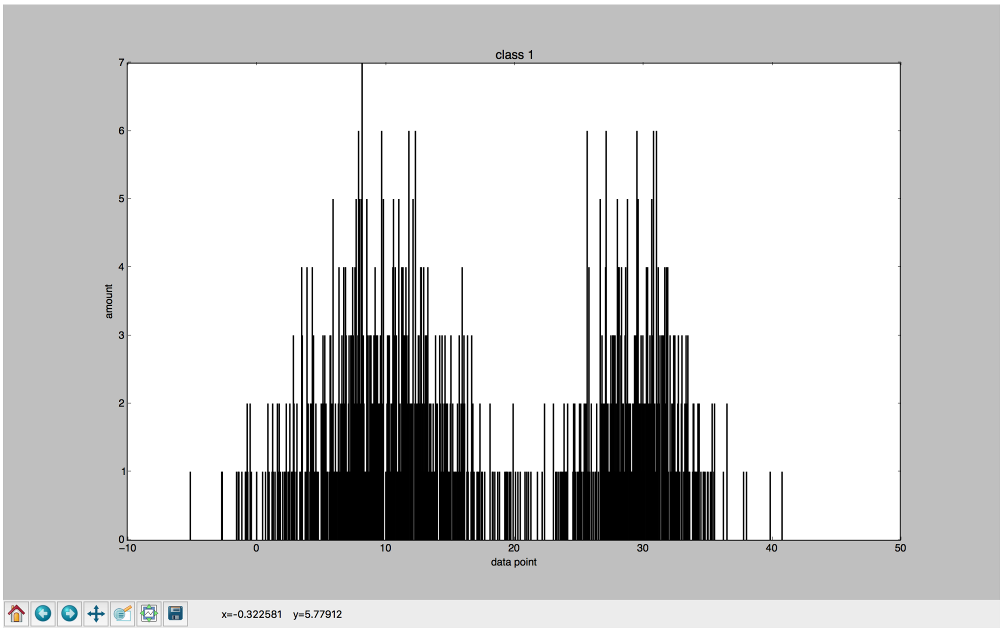
Xinyi Chen

1.

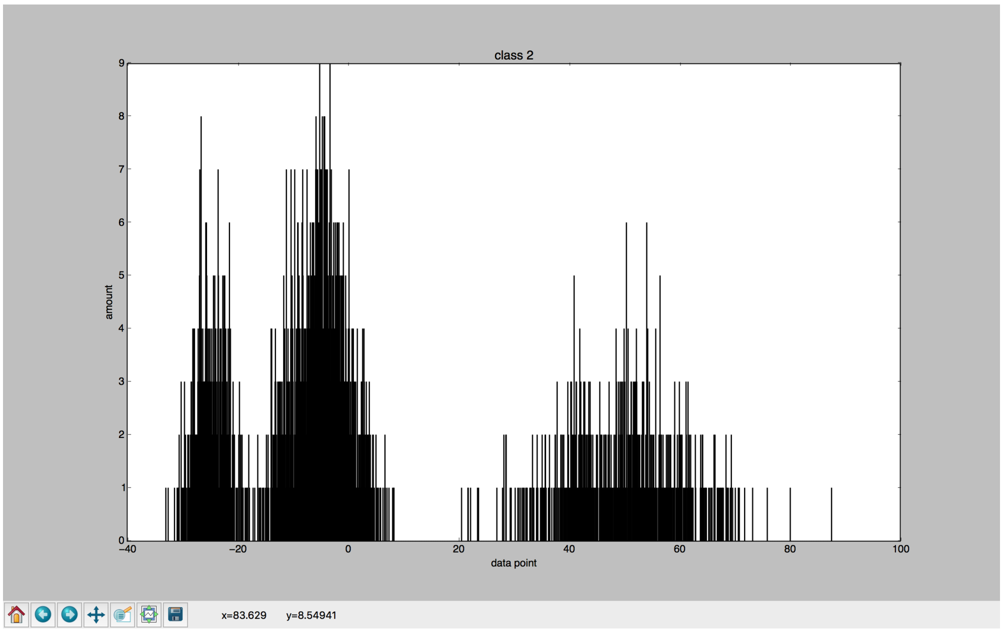
Code in *gmm_est.py*.

2.

plot of class 1 in *gmm_train.csv*:

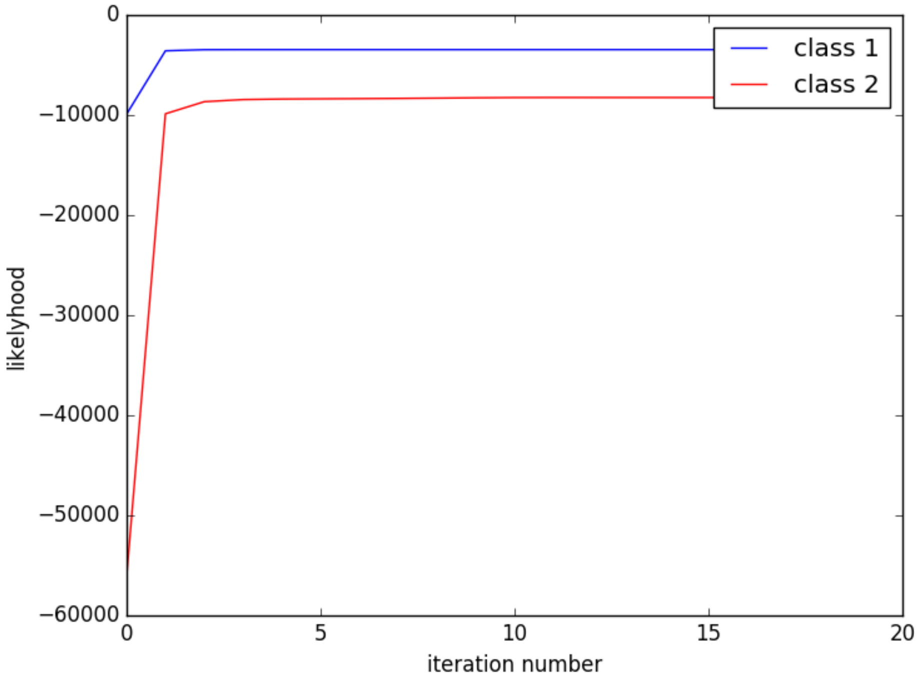


plot of class 2 in *gmm_train.csv*:



As we can see in these two pictures, class 1 has two Gaussian distributions(k=2, wt1 = [0.5, 0.5], mu1 = [10.0, 30.0]), class 2 has three Gaussian distributions(k=3, wt2 = [0.2, 0.3, 0.5], mu2 = [-25.0, -5.0, 50.0]).

log-likelihood values for the first 20 iterations:



Yes I think my program has converged, because the likelihood doesn’t change when iteration number > 5.

The final value of class 1:

```
mu = [9.7748859238684336, 29.582587183078552]
sigma^2 = [21.92280456596362, 9.7837696120162487]
w = [0.5976546303982766, 0.40234536960677603]
```

The final value of class 2:

```
mu = [-24.822750467609133, -5.0601577749460143, 49.624444718976456]
sigma^2 = [7.9473429252426451, 23.322655648051306, 100.02433752568446]
w = [0.20364949715643121, 0.49884300505742218, 0.29750751768089628]
```

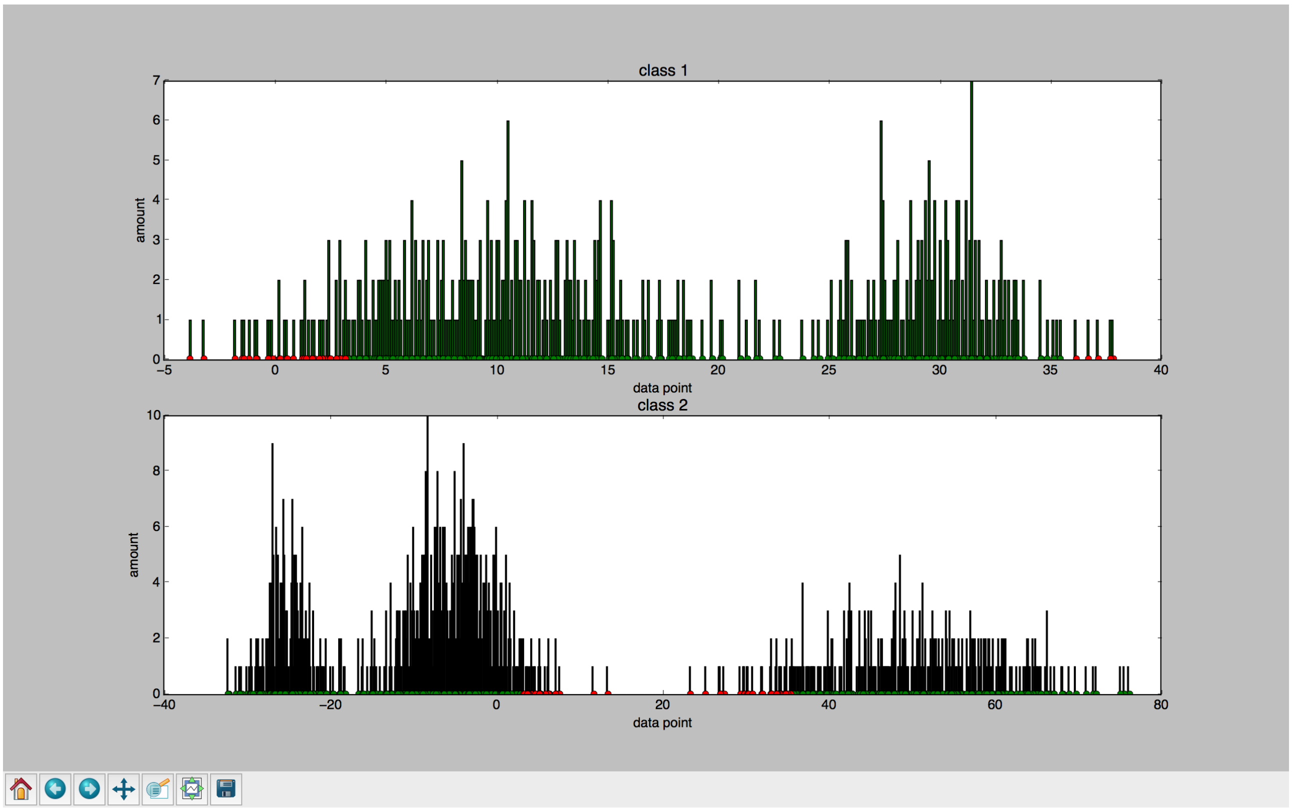
3.

Please run python *gmm_classify.py* <path/to/gmm_train.csv>

4.

prior error rate: 33.33%, GMM error rate: 6.13%

histogram:



As we can see in the histogram, histograms are the original data of *gmm_test.csv*, green data points are correctly classified data points, red ones are misclassified data points(it means in class 1, if a data point classified as class 1, it will be green. If classified as class 2, it will be red. And in class 2, if a data point classified as class 2, it will be green. If classified as class 1, it will be red)

We can see that GMM has lower error rate than prior probability, so GMM is more accurate. But in edge of Gaussian distribution, GMM is more likely to give a wrong classification.

5.

Yes, we can use closed-form solution to find the parameters, so there's no need to use Expectation Maximization. Because we already know about how many Gaussian distributions we have and data points they generated, so we know every detail about

Gaussian distributions, we can calculate them directly, so there's no need to find the parameters by using Expectation Maximization.

6.

Yes, we must use Expectation Maximization to find the parameters for the GMM. Because we only know how many Gaussian distributions, but we can't calculate the parameters directly because we don't know the data points they generated, so we must learn these parameters by using Expectation Maximization.