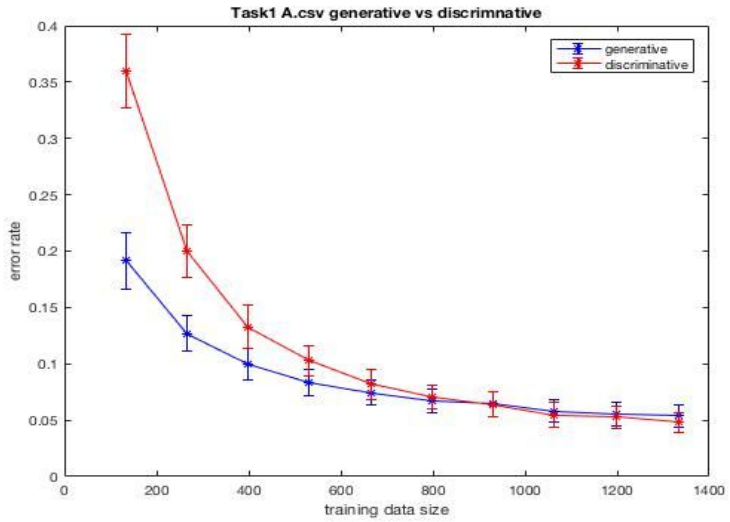
Report for Project 3

Zhaokun Xue

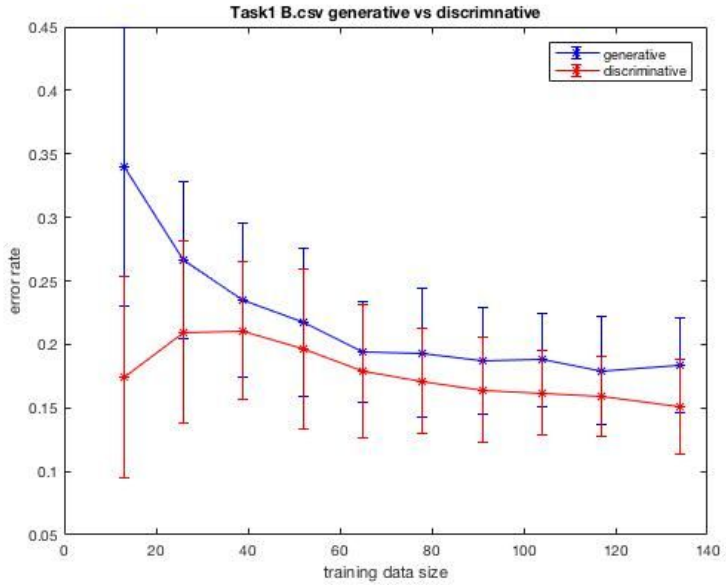
- **Task1**

  Results

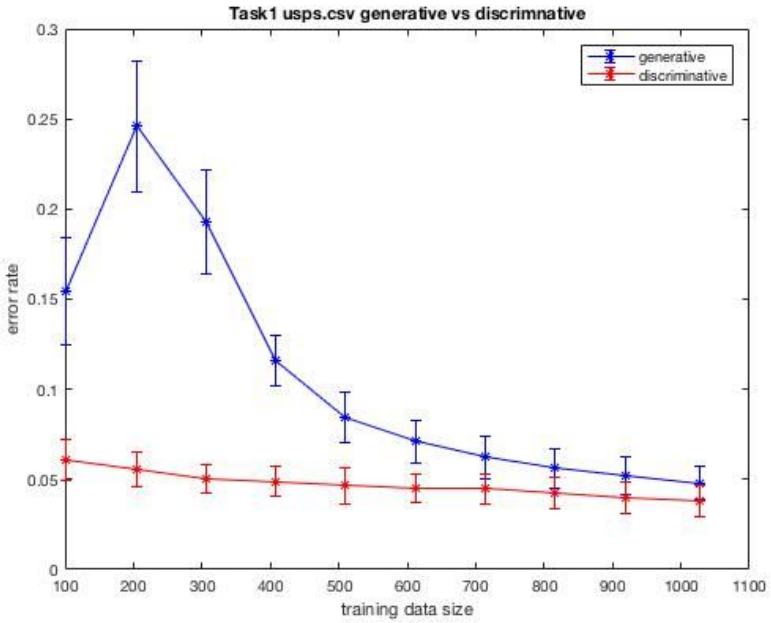
A.csv



B.csv



usps.csv



For task 1, I increase the training dataset size (0.1 * total training size) every time i.e. I work on training set size from 0.1 to 1 of total training set size. Therefore, there are 10 data points for each line in the graphs.

- **What can you observe on the performance of the algorithms?**
  - Observations for A.csv:

    When training set size is small, generative method has a better performance than discriminative method. As the training size grows, they two converge to almost the same optimal performance. And as the training size increases, their standard deviations decrease. Discriminative method has a slightly better optimal solution.

- Observations for B.csv

  For this dataset, discriminative method always has a better solution than generative method. And as the training set size grows, they converge to almost the same performance, and discriminative method converges to a slightly better solution. And their standard deviations decrease with the increase of training set size.

- Observations for usps.csv

  For this dataset, discriminative method always has a better performance than generative method. For generative method, it starts from a relatively high error rate, and finally converge to the optimal solution. And its standard deviation decreases as the training set increases. As for discriminative method, it could reach to a really good performance, even though the training set size is really small. And as the training size grows, its performance just slightly float around the optimal solution. And discriminative method always has a small standard deviation.

- **What can you conclude from these observations?**

  Based on these observations, we can make the following conclusions:

1. From the observations on A.csv, we can conclude that when we work on dataset that is linearly separable but does not conform to the Gaussian generative model, we should apply generative method, as the discriminative method has a requirement to use a Gaussian prior.

2. From the observations on B.csv, we can say that when the dataset is generated from Gaussian and linearly separable, we should use discriminative method, because it always has a better performance with a smaller standard deviation than generative method. And discriminative method has less computation work, as it has fewer parameters to be calculated.

3. From the observations on usps.csv, when we only have a small size of training set, we should use discriminative method, as it has much better performance on small size training dataset. And it has a smaller standard deviation. Another reason we should firstly consider to use discriminative is that it has a simpler computation process, as we need to calculate fewer parameters in this method.

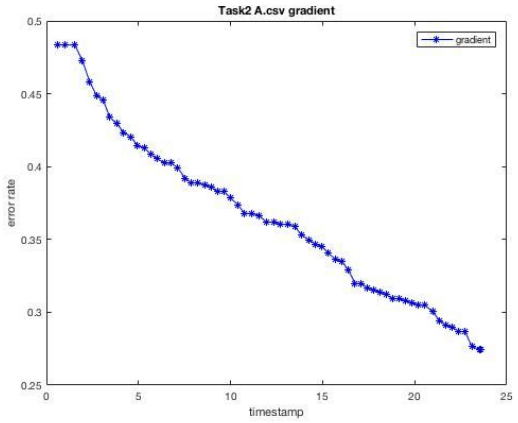
   All in all, when we have an unknown training dataset, it seems discriminative method is a better choice. Discriminative method has a better performance on small size dataset and it finally would converge to a good optimal solution. And we have less computation work to do with discriminative. And it provides a smaller standard deviation. If we know that the training dataset is not Gaussian, and the training dataset size is relatively small, generative method should be a better choice.

- **Task 2**

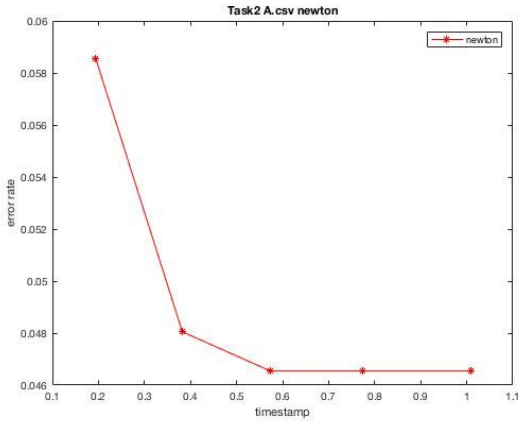
  Results

A.csv

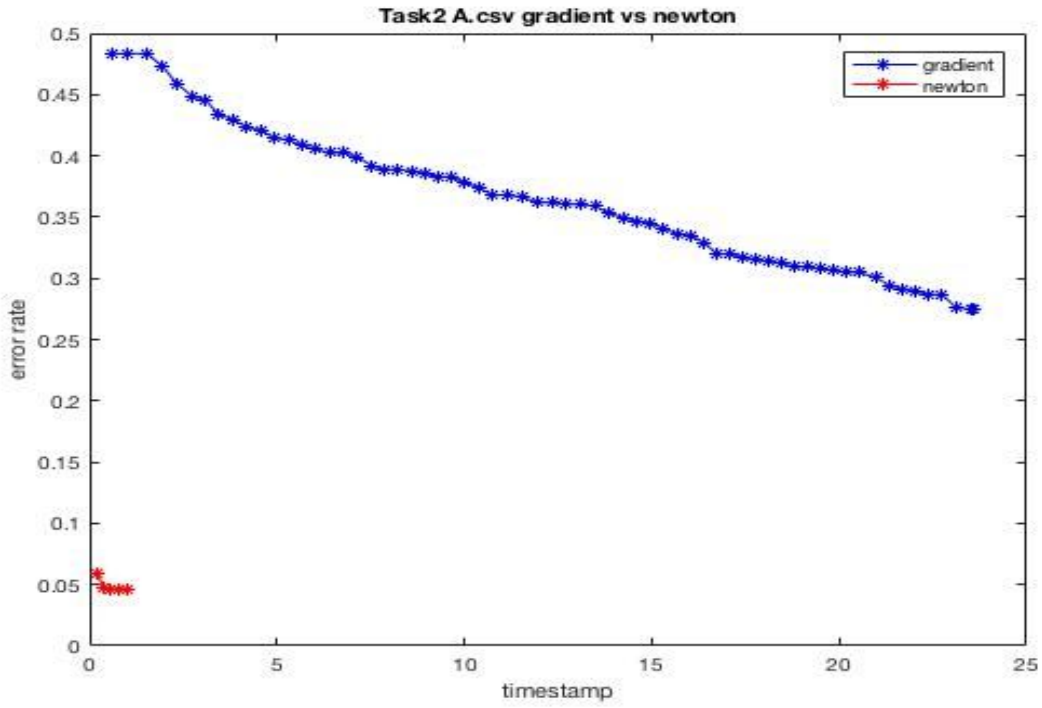
Gradient



Newton

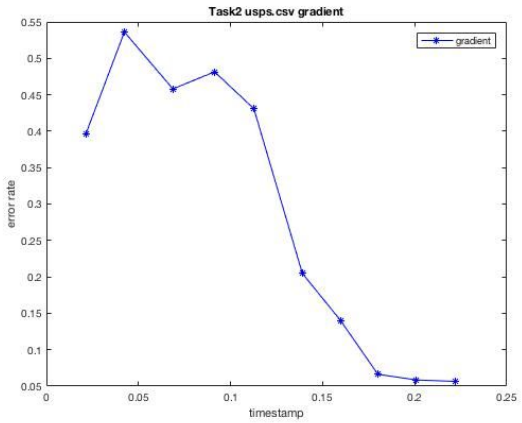


Combine results

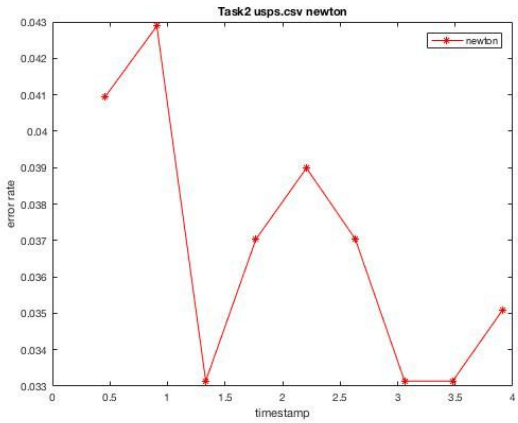


usps.csv

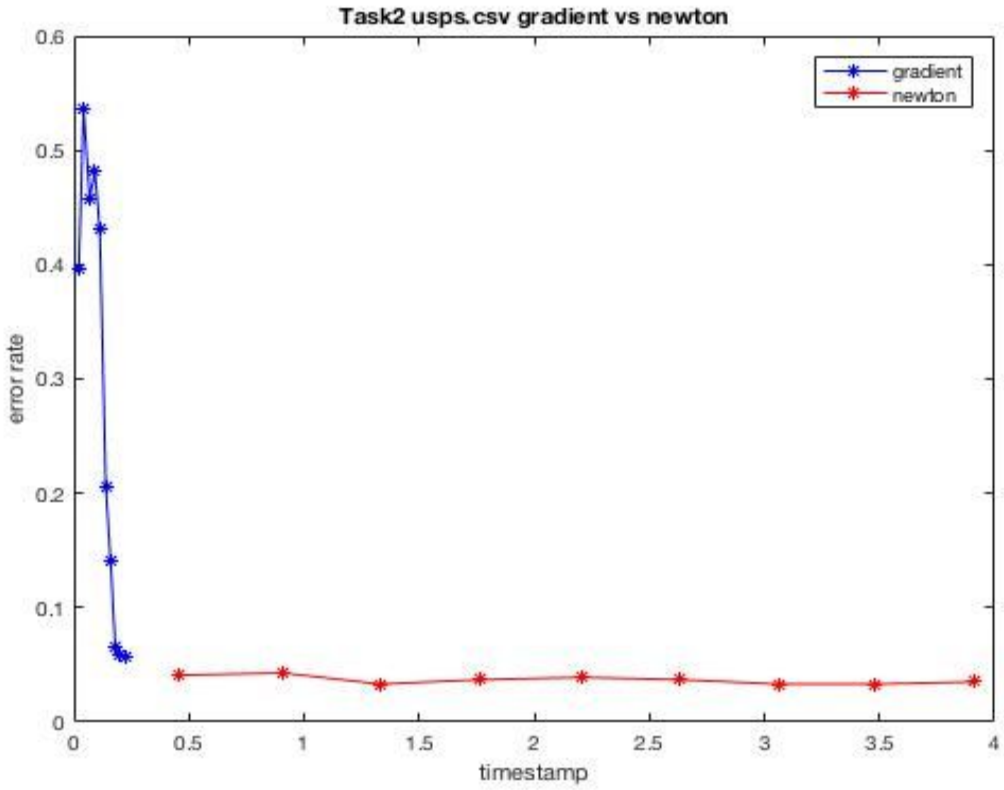
Gradient



Newton



Combine



For A.csv, I store w after every 30 updates for gradient method, and at last, I store the optimal w for gradient method.

- **What can you observe on the performance of the algorithms?**
  - Observations for A.csv

    For gradient method, it converges to its optimal solution after 1896 times iterations. Newton converges to its optimal solution after 5 updates. Newton method could reach a really good solution by only one update. After 1896 updates, gradient method still has a much worse performance than newton method. Newton method takes a much shorter running time to reach its optimal solution. And its performance just floats slightly around the optimal solution for each update.

  - Observations for usps.csv

    For gradient method, it converges to its optimal solution after 10 updates. And newton method converges to its optimal solution after 9 updates. They two have almost the same optimal performance. Gradient method takes a much shorter time to finish all updates. The time for updating gradient method 10 times is even shorter than 1 update for newton method. However, newton method could reach to a really good performance only after 1 update. And then its performance just floats slightly around the optimal performance.

- **What can you conclude from these observations?**

  Based on the previous observations, we can reach the following conclusions:

  1. When we have a reasonable size of training set, Newton method could always give us a really good performance only after 1 or few

updates. And it could usually converge to its optimal performance in a few updates.

2. The performance of gradient method is really sensitive to the form of dataset. If the dataset is not generated by Gaussians, it usually has a really bad performance with worse error rate, more updates to converge, and takes a longer running time. If the data is well-fitting for the algorithm, gradient method would have a much better performance. It could converge to its optimal performance in a few updates and with a much shorter running time as its computation is really cheap.

In a nutshell, if we have no idea about the training data we work on and we do not care about the running time, we should consider to use newton method first, as it can always give us a relatively good optimal solution in a few updates. On the other hand, if we have enough knowledge about our training dataset, and we know gradient method could work well on this dataset, we should apply gradient method. Because it can reach to almost the same optimal solution as newton method, but with a much shorter running time. For another situation, if we do not really care about the exactly optimal solution, and we just want to get a roughly optimal solution, I think we can definitely think about always using newton method and stop it after 1 update. Because based on our experiments, we notice that newton could give a really good solution for only 1 update.