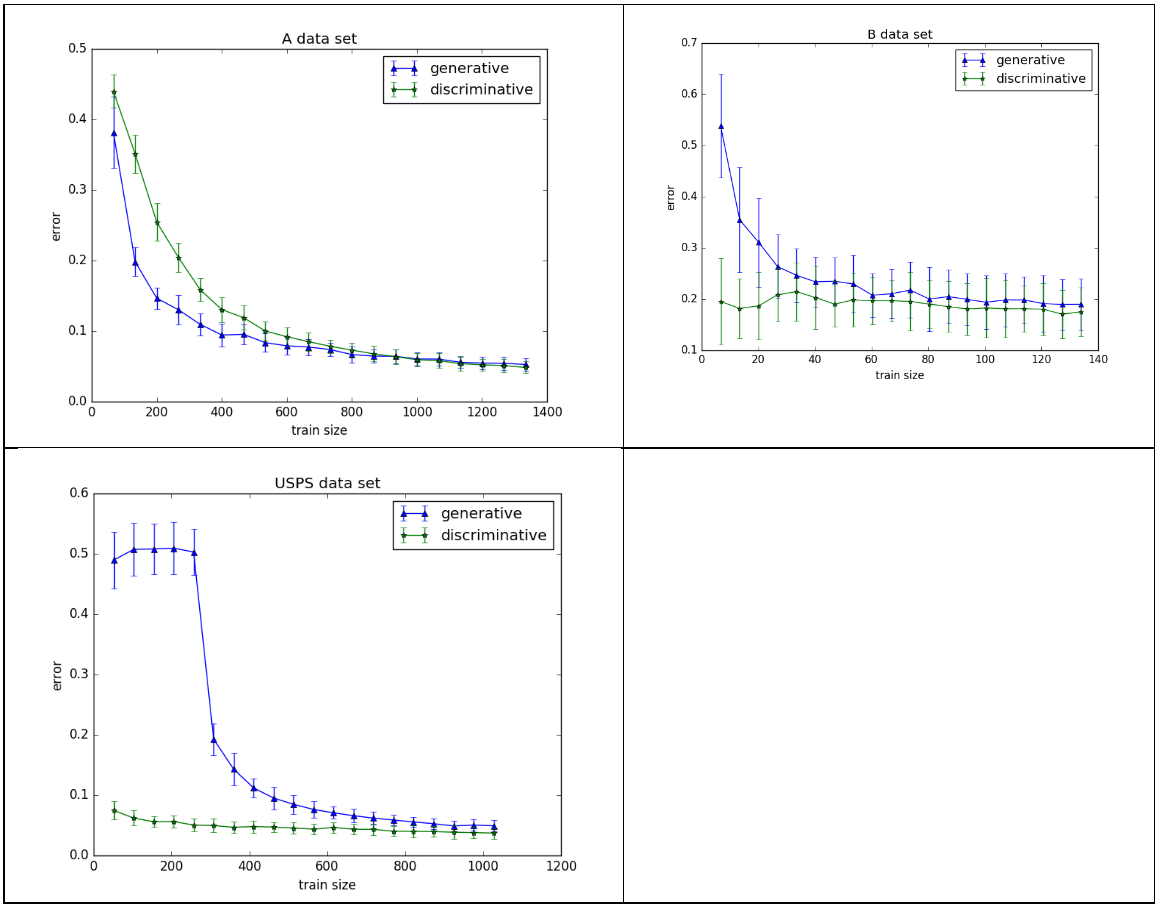
Comp136 HW3 Report

**Beibei du**

## Task1: Generative vs. Discriminative




- **What can you observe on the performance of the algorithm?**

  For all of these three data sets, if train size is large enough, discriminative model performs better then generative model, when the train size is small, data set A, generative model performs better, for the other two data sets, discriminative model performs better.

  Dataset A which does not conform to the Gaussian generative model but the data is linearly separable, from the figure we can find that when train size is small, generative performs better, but when train size is big, even discriminative model performs better than generative model, but generative performs also not bad.

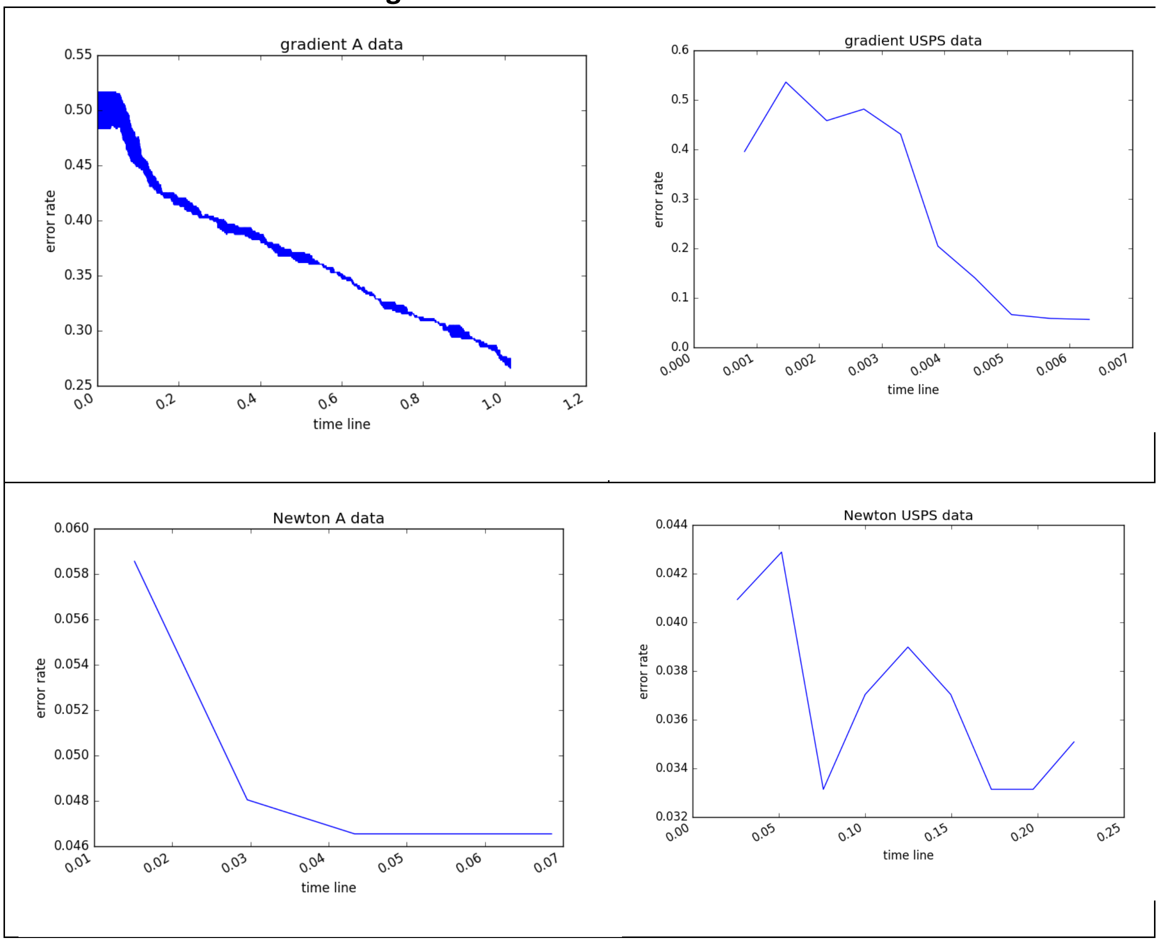
  Dataset B which diverges even further from the generative model and is somewhat linearly separable. From the figure, we can find that discriminative model performs better for any train size, generative model error rate decrease with the increase of train set size, also standard derivation of dataset B is bigger than dataset A and USPS according to the figure.

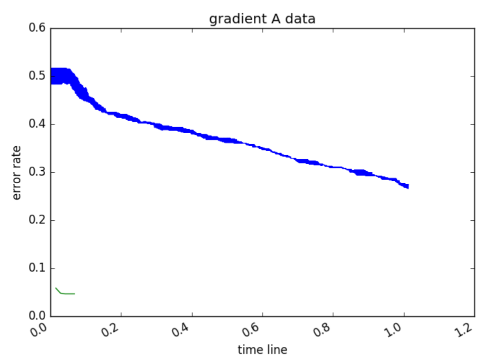
Dataset USPS, discriminative model has a low error rate with smaller derivation even the train size is small, and generative model we can find that there is a big drop of error rate when it arrives at some train size. Finally, the error rate of two models can be almost same.

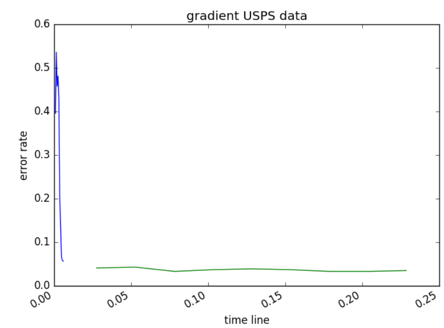
- What can you conclude from these observations?

  Discriminative model performs better (lower error rate), especially when the train size is small. Generative model performs better based on a list of assumption as like data set A. So if we know that the data are drawn from uniform distributed data and not conform to Gaussian but linearly separable, can use generative model according to A figure, but in the real world, for random data set, discriminative is a good choice according to USPS figure.

**Task2: Newton's method vs. gradient ascent**






- What can you observe on the performance of the algorithm?

| | Gradient A | Newton A | Gradient USPS | Newton USPS |
|---|---|---|---|---|
| counter | 1896 | 5 | 10 | 9 |
| Time spent | 1.0273s | 0.068577s | 0.0065s | 0.0232269s |
| final error rate | 0.2748 | 0.0465 | 0.05653 | 0.0350 |

Dataset A:

Gradient method needs to run 1896 times to converge with total time 1.0273 seconds. Newton method only needs to run 5 times to converge with total time 0.068577 seconds, much faster than Gradient A method. Also, for final error rate, Newton method is much lower than gradient method.

Dataset USPS:

Gradient method needs to run 10 times with total time 0.0065s, and newton method needs counter 9 with total time 0.02322s, for this data set, gradient USPS method is much faster than Newton method. The final error rate for both methods is relatively low.

- What can you conclude from these observations?

For dataset like A does not conform to the Gaussian generative model but is linearly separable, number of features is not very big (#Feature A dataset is 60), in this case it's better to use Newton method, since it's fast (even needs computation of inverse Hessian, but Matrix sigma is not very big) and also has a good performance;

For dataset like USPS, it depends. If want to save time and relatively good performance, it's better to use Gradient method, if want much better performance, it's better to use Newton method. Considering dataset USPS, the number of features (# features: 256 is) large, matrix of Sigma (Matrix size: 256 * 256)is large. Newton method is expensive since the computation of the inverse Hessian is demanding, so it needs more time, but Gradient method don't need that computation, so it saves time.