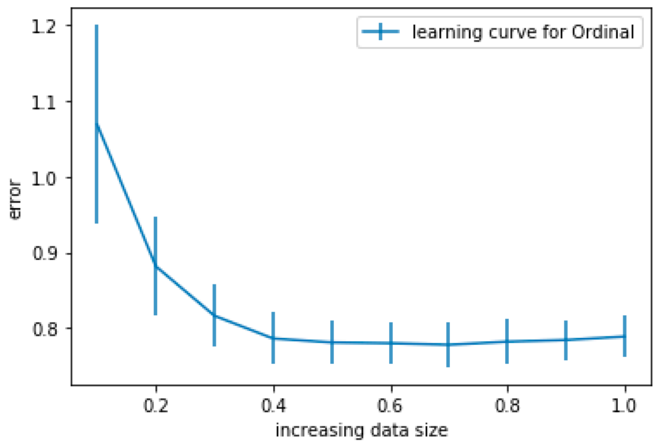
Programming project 3

Assignment Questions : (Here the values of position required for convergence and time taken for training data are in increasing data size from 0.1 … 1)

Method : As discussed in the assignment I have implemented as a separate optimization function and called for whichever method we are using . We calculate W map and then calculate the test error .

Graph for the ordinal regression dataset(AO) :

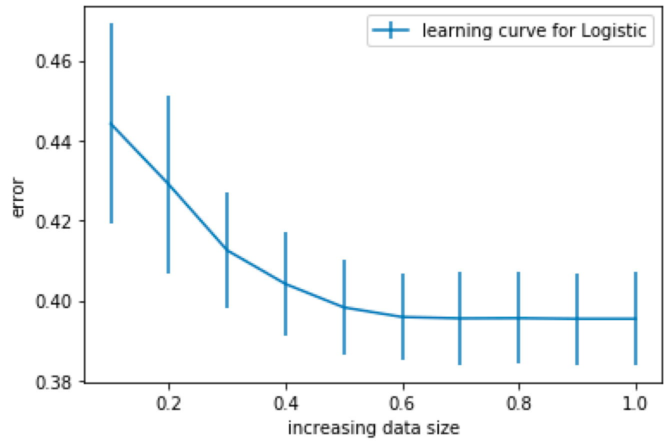



```
 average position for convergence for different training size :
[6.133333333333334, 5.166666666666667, 4.966666666666667, 4.866666666666666, 4.566666666666666,
4.333333333333333, 4.233333333333333, 4.033333333333333, 3.933333333333333, 3.8333333333333335]

 average time for convergence for different training size:
[0.004851552733316567, 0.00733066999997997, 0.009783047433332588, 0.012454332033325955,
0.014843783233345675, 0.016849290366675027, 0.02345780766667455, 0.026282275633305593,
0.030205697099995633, 0.034194833666704956]

 running time is 6.287846979000051
```

Explanation :
- Here the average number of positions required for convergence is decreasing and as the data size is increasing and training time is increasing as the data size increases which is as expected . Hence the learning curve we got is as expected and the total running time for the

ordinal regression is 6.28 seconds . Here we the increasing data size the error also reduces.

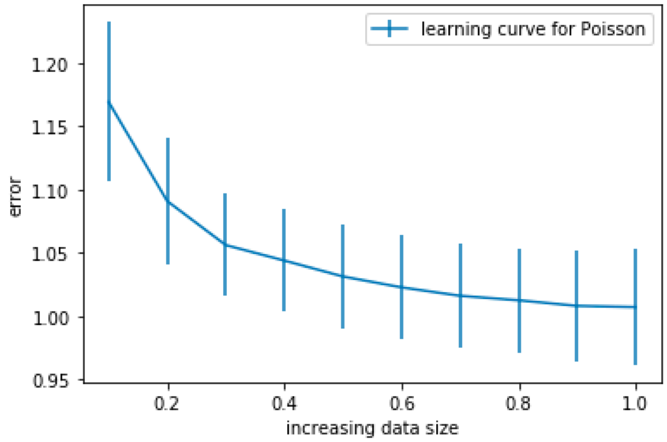
Graph for the A dataset (Logistic):




```
 average position for convergence for different training size :
[99.0, 99.0, 99.0, 99.0, 98.73333333333333, 52.53333333333333, 9.433333333333334,
0.5333333333333333, 0.0, 0.0]

 average time for convergence for different training size:
[1949.6361325451, 1949.6637270923336, 1949.6981043805667, 1949.7395001446, 1949.7874961618,
1949.8167489362336, 1949.8232599438334, 1949.8244866531, 1949.8253198077662, 1949.8262372131999]

 running time is 6.803252206000025
```

Explanation :
According to this graph, the steps required for convergence are decreasing quickly as soon as the data size is increasing and the time taken to train the data is not increasing significantly or we can say that it almost remains constant as we are getting convergence faster
And data size increasing the change in training time is less . The total time required to run the process is 6.8 seconds . With increasing data size error is also decreasing.

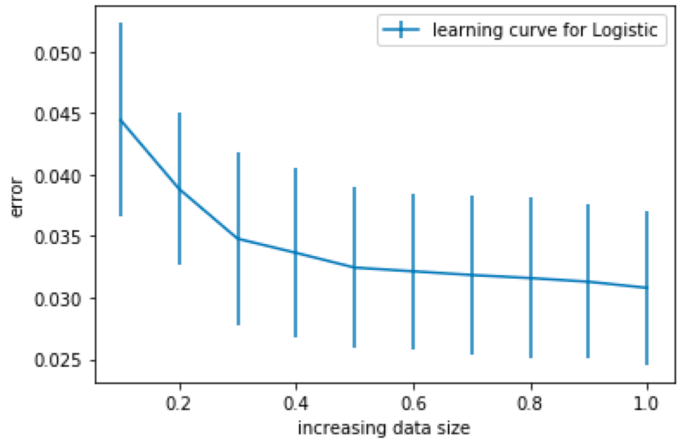
Graph for AP dataset (Poisson) :




```
 average position for convergence for different training size :
[6.3, 2.7, 2.1, 2.0, 2.0, 2.0, 1.9333333333333333, 1.6333333333333333, 1.4666666666666666,
1.3666666666666667]

 average time for convergence for different training size:
[0.002609372833361097, 0.0012677227333218373, 0.004365440966618432, 0.00569500626666013,
0.008087361633351975, 0.01228587673332792, 0.017633285333295133, 0.023392433533308575,
0.029394824733284017, 0.021825914733350753]

 running time is 4.318584527999974
```

Explanation :
According to this graph, the steps required for convergence are almost the same at 2 and as the data size is increasing, the time taken to train the data is increasing or we can say that the learning curve is as expected as the steps to converge remains constant and average time taken to train increases with increasing data size . The total time required to run the process is4.318 seconds seconds .. With increasing data size error is also decreasing.

Graph for USPS dataset(Logistic) :




```
 average position for convergence for different training size :
[99.0, 99.0, 99.0, 98.9, 91.86666666666666, 71.83333333333333, 63.9, 48.0, 43.1,
41.93333333333333]

 average time for convergence for different training size:
[2542.0260950567995, 2542.1983574747337, 2542.398927635533, 2542.6140373829667,
2542.830357909967, 2543.0156343301674, 2543.1922990942007, 2543.3329855932325, 2543.4760694349,
2543.6221225637337]
```

Explanation :
According to this graph, the steps required for convergence are decreasing as the data size is increasing, the time taken to train the data is increasing or we can say that the learning curve is as expected as the steps to converge increases and average time taken to train increases with increasing data size . The total time required to run the process is 34.3 seconds seconds . . With increasing data size error is also decreasing.

Conclusions :

Here the runtime will be different for all the models and graphs as the data is different.
Run time depends majorly on the data - (iterations required for convergence) , (complex calculations - for (r , d , t , and error)) , (length of the data set).

Here the usps logistic regression data takes the highest time because of the convergence of steps taken each time.

Extra credit Question’s Answer :

Here as the value of alpha increases the accuracy increases till some point and then the accuracy decreases or remains constant.

Approach for the method :

Here , this approach is inspired from grid search , linear search and binary search.
Here for one parameter we will have to provide the upper bound and lower bound of the parameter . Now we will calculate error for the given values of our hyper parameter. We will now randomly select a number between the given bounds and then calculate the error for that given randomly selected hyper - parameter. Now we will change the lower bound if our error of hyper parameter is less than the error of the lower bound and if the error of the hyper parameter is less than the upper bound than we will replace our upper bound with the randomly selected hyper parameter and repeat this process for around 70-80 times and we will get the approximate best value.

Now for tuning two parameters :
To get the most optimum answer we will have to do grid search which will take a lot of time . A better way to find in almost all cases the best hyper - parameters is to set the upper bound of the parameters we got from the method mentioned above and then do the linear search for very less range of parameters as we know that the best value is within this range . Doing so we will get the best set of the hyper parameters for the given data . The accuracy of the ordinal for S=1 and alpha =10 was around 62% and the accuracy after tuning the parameters with alpha = 19 and S = 1 is around 70% which we got from performing the above mentioned method for tuning hyper parameters.

Here the main objective of performing the operation is that we can get the approximate best upper bound of the the hyper parameter and considering an error of +-5 we can then do linear search so in total if alpha [0,1000] and S[1,50] . If we do linear search we will need approx 50000 iterations to find the best value for the parameters.

If we perform the above mentioned method we will need 300 iterations

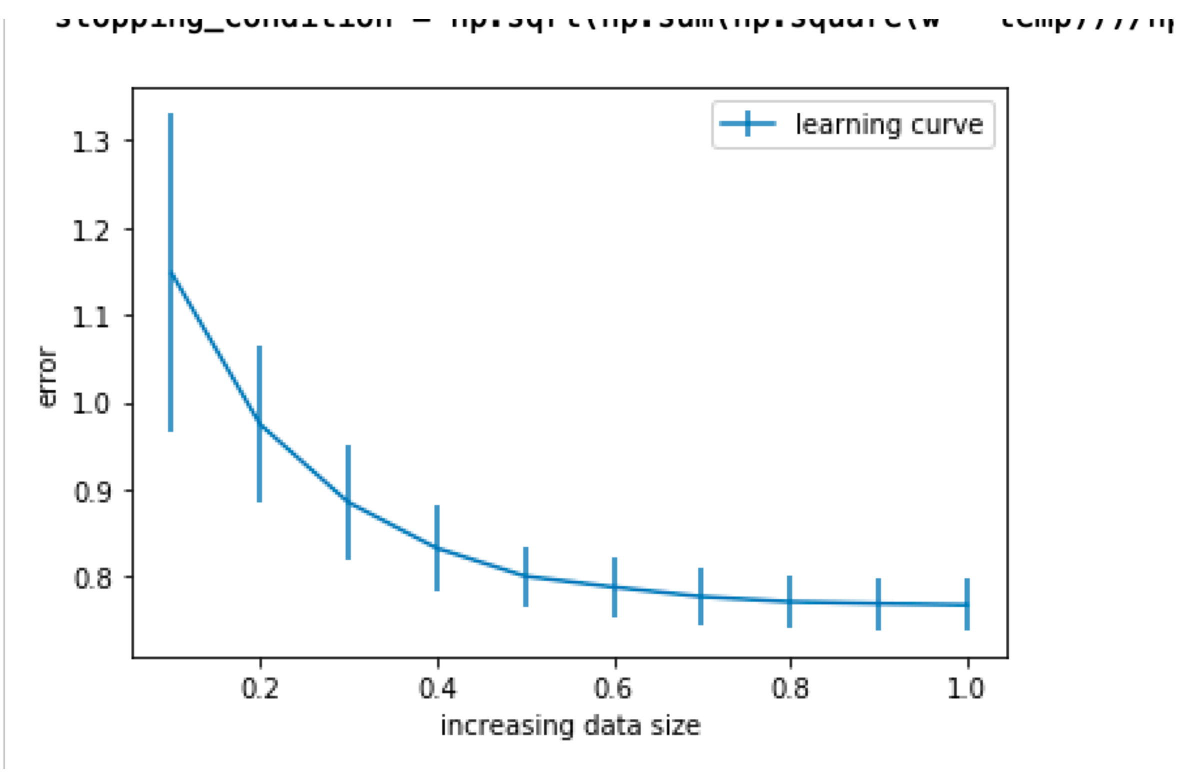
for first value , 300 for second and 100 iterations for performing linear search . That is we will be able to find accuracy almost similar to the best solution within 700 iterations which is significantly very less compared to linear search for all the parameters.

Pseudo code :

We will take one parameter as constant and find the best value of the other parameter. We will do this by repeating the same experiment 10 times and then taking the min error for each iteration and then again taking the minimum we found in all those 10 iterations. From this we will get our in this case our alpha value. After that for that value we will find Our S by similar method .

Now we will consider the lower bound and the upper bound to be +5 and -5 of the found out best value. We will now do linear search for this method and then find the best alpha and S value.

After tuning parameters :



According to the method mentioned above this is the kind of graph I am getting for ordinal regression .

Here for the sake of simplicity for verification I am splitting my train data as first 66% of the given dataset and my test data as the last 33% of the given dataset.