CSE474/574: Introduction to Machine Learning (Fall 2017)

Project 2: Linear Regression

Name	**Anirban Chatterjee**
Person Number	**50249214**
Name	**Vikram Singh**
Person Number	**50247207**

In this project, we had to train a linear regression model to evaluate two datasets (LeToR and Synthetic) and find out the root-mean-squared error for a set of test data via closed form solution and stochastic gradient descent.

The common parameter for both the models is the design matrix which we get from the formula:

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^\top \Sigma_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)\right)$$

Our goal is to minimize the error for our model. This can be done by tuning the hyperparameters. Hyperparameters control the distribution of the output and hence the more accurate we can make them, the less error we get. This is mostly done by hit-and-trial method.

The hyper parameters chosen by us are as follows:

For Closed Form

* Cluster Size = 10

* Regularization Parameter = 0.01

For Stochastic Gradient descent

* Cluster Size = 10

* Regularization Parameter = 0.01

* Learning rate = 0.001

* Number of epochs = 1000

Using the training set provided we tuned our hyper parameters. We then retrieved the model parameters from the validation set. The model parameters obtained are optimized due to our tuned hyper parameters. The validation set error and the training set error achieved are minimum.

Considerations: From the following project we learned that during the k-means clustering the cluster size should neither be too high, to avoid over fitting nor the cluster size should be too low which results in under fitting which does not generalize well. Hence to get an optimum balance, our choice of cluster size is inherently important. Our cluster size of 10 was an optimum fit which avoided over fitting as well as underfitting.

One of the problems that can occur is the problem of overfitting. In our case, we tried to minimize this by taking the size of the training set to be large and by setting the learning rate to be as small as 0.001. The learning rate should also be selected in such a way, so that it's not too large so that it overshoots out minima and neither should it be too low so that our algorithm does not take too much time in computing the minima. The learning rate of 0.001 selected by us neither overshoots the minima nor does it make the computational time too large. The same we learned about selecting the mini batch size.

Conclusion: From the project, we learned that closed form solutions are preferable when the dataset is small. For larger datasets, the transpose of the matrix may not be available, or it may be too expensive to compute. For SGD, sometimes the computation cost is less than closed form for large datasets. Also, for most cases, SGD converges at the global minimum than the local minimum.

The results obtained are as follows:

UBIT ID: **anirbanc, vsingh25**

Person Number: **50249214, 50247207**

```
Synthetic/Closed: Design matrix for test set:
[[ 1.    0.956961   0.96068013 ..., 0.96372571          0.96500899
   0.96984918]
[ 1 0.96104556 0.96618566 ..., 0.97132213 0.96119238
0.97767605]
[ 1.0.94343632 0.9466423 ..., 0.95491618 0.93940765
 0.96875491]
...,
[ 1. 0.97647708 0.975517..., 0.98317991 0.97415517
  0.9668724 ]
[ 1.0.96708853 0.97407741 ..., 0.97561035 0.95804717
  0.97018641]
[ 1.0.96496405  0.95787749 ...,  0.96880639  0.96160823
   0.96600767]]

Synthetic/Closed: Closed form solution for test set:
[-31.12560536 -16.5498026   -7.09490998 ...,  -
2.75164356   3.47666718
  22.39830914]
Synthetic/Closed: Test set error: 0.021331136083870387
Synthetic/SGD: Design matrix for test set:
```

```
[[ 1.         0.956961    0.96068013 ...,  0.96372571  0.96500899
   0.96984918]
 [ 1.         0.96104556  0.96618566 ...,  0.97132213  0.96119238
   0.97767605]
 [ 1.         0.94343632  0.9466423  ...,  0.95491618  0.93940765
   0.96875491]
 ...,
 [ 1.         0.97647708  0.975517   ...,  0.98317991  0.97415517
   0.9668724 ]
 [ 1.         0.96708853  0.97407741 ...,  0.97561035  0.95804717
   0.97018641]
 [ 1.         0.96496405  0.95787749 ...,  0.96880639  0.96160823
   0.96600767]]

Synthetic/SGD: Closed form solution for test set:
[-31.12560536 -16.5498026   -7.09490998 ...,  -
2.75164356   3.47666718
  22.39830914]

Synthetic/SGD: Test set error: 0.021331136083870387
Letor/Closed: Design matrix for test set:
[[ 1.         0.956961    0.96068013 ...,  0.96372571  0.96500899
   0.96984918]
 [ 1.         0.96104556  0.96618566 ...,  0.97132213  0.96119238
   0.97767605]
 [ 1.         0.94343632  0.9466423  ...,  0.95491618  0.93940765
   0.96875491]
 ...,
 [ 1.         0.97647708  0.975517   ...,  0.98317991  0.97415517
   0.9668724 ]
 [ 1.          0.96708853  0.97407741 ...,  0.97561035  0.95804717
   0.97018641]
 [ 1.          0.96496405  0.95787749 ...,  0.96880639  0.96160823
   0.96600767]]
```

Letor/Closed: Closed form solution for test set:

```
[-1.86611131  0.47292497  0.06860419 ...,  0.68334269 -6.41279497
 -0.1434949 ]
```

Letor/Closed: Test set error: 0.015532992661697324

Letor/SGD: Design matrix for test set:

```
[[ 1.          0.91460313  0.88525072 ...,  0.85946503  0.93502994
   0.84780847]
 [ 1.          0.92770546  0.88467108 ...,  0.86402275  0.9552207
   0.85259124]
 [ 1.          0.89728631  0.87243186 ...,  0.84772824  0.90809806
   0.83125126]
 ...,
 [ 1.          0.93886716  0.80862354 ...,  0.95861731  0.94741332
   0.86821859]
 [ 1.          0.97684147  0.78112925 ...,  0.88479508  0.9362568
   0.83094374]
 [ 1.          0.96779707  0.82799945 ...,  0.91968126  0.93700796
   0.89026833]]
```

Letor/SGD: Closed form solution for test set:

```
[-1.86611131  0.47292497  0.06860419 ...,  0.68334269 -6.41279497
 -0.1434949 ]
```

```
Letor/SGD: Test set error: 0.015532992661697324
```