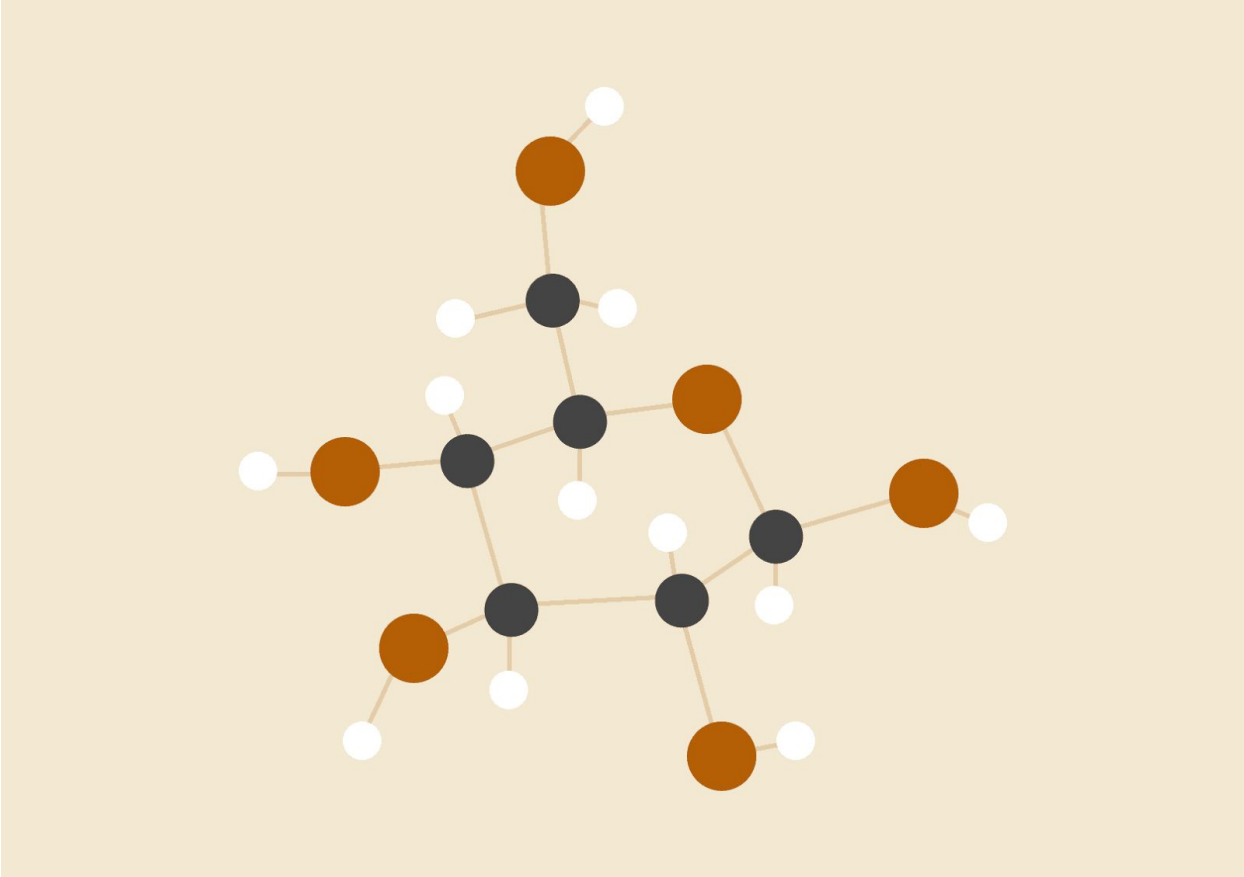
PROJECT-2 REPORT

Introduction to Machine Learning - CSE574



Jayant Solanki

UBIT Name: jayantso
Person Number: 50246821
Fall 2017 CSE

Swati Nair

UBIT Name: swatishr
Person Number: 50246994
Fall 2017 CSE

INTRODUCTION

This report contains the implementation details of Project 2 i.e. training a linear regression model on LeToR and synthetic dataset using closed form solution and stochastic gradient descent.

OBJECTIVES ACHIEVED

We have trained linear regression model using closed form solution and stochastic gradient descent.

- Firstly, we divided the given input and output data in three sets: Training (80%), Validation (10%) and Test (10%) using the random shuffle method.
- Then, we used grid search on M (number of clusters = number of basis functions) and λ (regularization coefficient).
- Using k-means clustering, derived M basis functions.
- Then, by three different learning techniques i.e. closed form solution, stochastic gradient descent without early stopping and SGD with early stopping, computed the weights for different values of M and λ on the training set.
- The model was then used to compute the validation set and root mean square error was computed.
- We tested the model on the testing set using M and λ for which we got least RMS error on validation set. The test error was computed which shows the amount of generalization achieved by learning.
- Above steps were done for both LeToR and synthetic datasets
- We created a tabulation data for the varying values of lambda and clusters K, then we chose the lambda and clusters number where we got the minimum value in the validation error. Lowest validation error was returned by our code for each cluster by using simple if-else comparison. This was our tuning of parameters for the calculation of the test errors.
- Closed form solution throws memory error when we increase the value of M. The best method is to use SGD with early stopping.
- Since the closed form performs matrix multiplication, this will have higher time complexity compared to SGD.

IMPLEMENTATION

Using **numpy's genfromtxt function,** the input and output data were read and stored into 'X' and 'Y' numpy array

Task - 1 : Split the input and output data into training (80%), validation (10%) and test set (10%) using train_test_split function of sklearn.model_selection. Let it be X_Train, X_Val, X_Test for inputs and Y_Train, Y_Val, Y_Test for outputs

Task - 2 : Use **grid search on hyperparameters** M and λ. Here, we have selected M ranging from 10 to 75 with step size 5 in case of LeToR data and 3 to 13 with step size 1 in case of synthetic data. λ ranges from 0.1 to 2 with step size 0.1

Task - 3 : Divide X_Train in M number of clusters and computed its centroids using **kmeans** function of scipy library. Also, computed covariance matrix of each cluster which will be the spreads in each basis function.

Task - 4 : Compute Design Matrix for training, validation and test sets using the centroids and spreads obtained from Task - 3. This design matrix is a vector of M Gaussian radial basis functions. M is determined on the cluster numbers using Kmeans clustering.

Task - 5 : For different λ values, compute M weights using either of the below three methods:

- Closed Form Solution: It takes λ, design matrix and output data as parameters and implements the closed form solution formula.
- Stochastic Gradient Descent without early stopping: Considered learning_rate = 1, minibatch_size=N (number of samples in training set), number of epochs=10000.
- Stochastic Gradient Descent with early stopping: In early stopping, for each epoch, we are calculating the validation RMS error. If the validation error in current epoch is lesser than the previous value, we consider the current epoch's weight. Also, suppose current validation error is greater than the previous one, we increment the counter j, which will be incremented on every incoming epoch cycles until a patience value (set to 10). Once it is equal to patience value, the SGD will be stopped and function will return the weights generated during the lowest validation error found till then.

Task - 6 : For each λ, we computed the RMS error for validation set (ErmsVal) and considered the weights, λ and M for the minimum ErmsVal. Thus, our model has been trained.

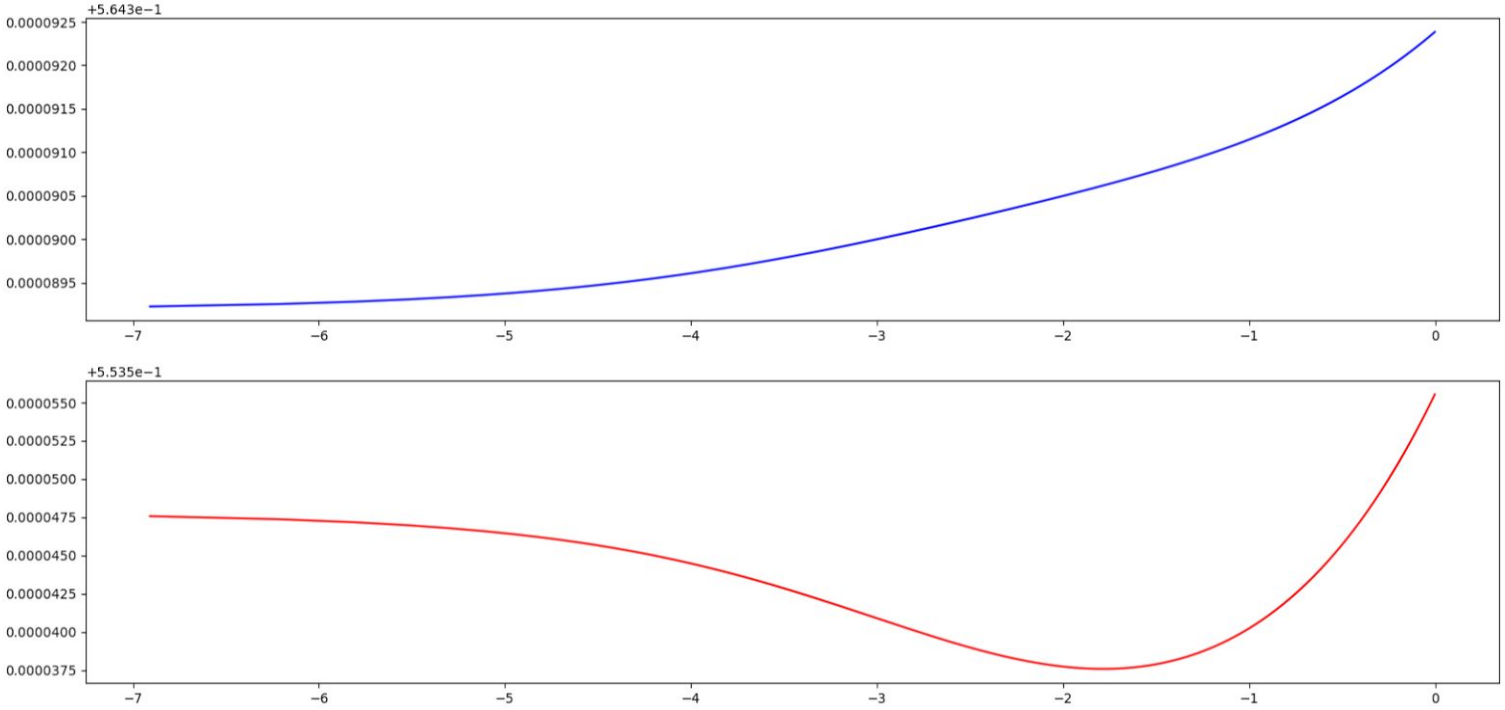
Task - 7: Trained Model is then ran on the Test set with the optimum values of hyperparameter obtained in Task - 6 and calculate the test error. It shows the generalization efficiency of our model.

IMPROVEMENTS

For efficient parameter tuning, below improvements were done:

1. Use of Early Stop to stop SGD process in case the validation error was not decreasing anymore.
2. Averaging the outputs of K-Means clustering:While performing K-Means on the training data set, we consider the average of the centroids generated by running kmeans 10 times
3. Randomly shuffling of input and output data:
    a. Used train_test_split on input and output dataset, which divides the data set into training, validation and test set randomly. This helps in efficient training of model

Graph for Training and Validation RMS Error versus ln λ : For K = 3(Synthetic data),



DATASHEET

Comparison table for Synthetic Data:

Sr.	M = no. of clusters	erms(CF)				Erms(SGD)			Erms(SGD with early stopping)	
		Lambda	Training	Test	Lambda	Training	Test	Lambda	Training	Test
1	3	0.1	0.7658	0.7699	0.1	0.7568	0.7592	0.1	0.7667	0.7661
2	5	0.1	0.7647	0.7709	0.1	0.7557	0.759	0.1	0.7596	0.7633
3	7	0.1	0.755	0.7679	0.1	0.7617	0.7662	0.1	0.762	0.7664
4	9	0.1	0.7574	0.7672	0.1	0.7569	0.7608	0.1	0.7623	0.7653
5	11	0.1	0.7595	0.7675	0.1	0.7559	0.758	0.1	0.7559	0.7579
6	13	0.1	0.7557	0.7653	0.1	0.7565	0.7653	0.1	0.7688	0.773
7	15	0.1	0.7581	0.7736	0.1	0.7568	0.7611	0.1	0.7623	0.7661
8	17	0.1	0.753	0.7722	0.1	0.7609	0.7653	0.1	0.7686	0.7729
9	19	0.1	0.7498	0.76	0.1	0.7619	0.767	0.1	0.76	0.7675
10	21	0.1	0.7574	0.7783	0.1	0.7638	0.7697	0.1	0.7586	0.7622
11	23	0.1	0.755	0.7751	0.1	0.7644	0.7683	0.1	0.7604	0.7659
12	25	0.1	0.7488	0.7794	0.1	0.7631	0.7707	0.1	0.7614	0.7675
13	27	0.1	0.7515	0.7837	0.1	0.7629	0.7709	0.1	0.7607	0.765
14	29	0.1	0.7531	0.7782	0.1	0.7671	0.7734	0.1	0.7678	0.7699

Comparison table for LeToR Data:

Sr.	M = no. of clusters	erms(CF)				Erms(SGD)			Erms(SGD with early stopping)	
		Lambda	Training	Test	Lambda	Training	Test	Lambda	Training	Test
1	10	0.7	0.5708	0.5741	0.1	0.5708	0.5741	0.1	0.5708	0.5741
2	15	0.8	0.5708	0.5741	0.1	0.5708	0.5741	0.1	0.5708	0.5741
3	20	0.6	0.5708	0.5741	0.1	0.5708	0.5741	0.1	0.5708	0.5741
4	25	0.1	0.5708	0.5741	0.1	0.5708	0.5741	0.1	0.5707	0.5739
5	30	0.7	0.5708	0.5741	0.1	0.5708	0.5741	0.1	0.5707	0.5739
6	35	0.3	0.5707	0.5738	0.1	0.5708	0.5741	0.1	0.5708	0.574
7	40	1	0.5708	0.5741	0.1	0.5706	0.5735	0.1	0.5706	0.574
8	45	0.1	0.5707	0.5742	0.1	0.5707	0.5738	0.1	0.5708	0.5741
9	50	1.5	0.5708	0.574	0.1	0.5708	0.5738	0.1	0.5708	0.5741

RESULTS

LeToR DATA OUTPUT

For M : 35
LeToR Data: Printing Closed Form Solution...
[3.13954825e-01 2.90625979e+00 2.22838250e-39 -5.07461366e-07
6.34588372e-02 -1.36363431e-07 1.89511408e-10 -2.70682832e-02
-5.83122466e-62 -1.41844699e+00 1.94128180e-15 3.34810283e-57
-1.45789731e-06 1.13820758e-04 -2.78201003e-10 -6.78771661e-05
3.92891057e-09 2.01974616e-07 -1.13916935e-04 -1.26302153e-14
-9.67496697e-01 -1.38499734e-06 -7.20655601e-07 -6.88578003e-06
3.32336881e-03 -1.64542128e-01 8.17969338e-01 -4.32097750e-10
-2.55556638e-05 4.54507873e-03 -7.59428491e-02 3.57975683e-03
1.28463913e-04 1.24794876e-10 -9.62803550e-75 1.70444526e-06]

At lambda : 0.1000 and M = 35, **Test Error using Closed Form: 0.5742**

LeToR Data: Printing SGD Solution without early stopping...
[3.14342810e-01 1.24777863e-01 3.96417663e-41 -6.94069103e-10
1.12259775e-03 -2.42597368e-09 3.37100623e-12 -2.79362502e-02
-1.03849350e-63 -9.74041169e-02 3.45340009e-17 5.95445001e-59
-2.60150877e-08 2.03620765e-06 -4.44845293e-12 -1.11225167e-06
8.27373181e-11 3.47903211e-09 -1.75770668e-06 -2.24992992e-16
-1.79535826e-02 -2.46691965e-08 -1.17961217e-08 -1.24739300e-07
7.09784255e-05 -2.94448474e-03 1.47088066e-02 -7.69522917e-12
-4.50766776e-07 8.12704150e-05 -1.37787385e-03 5.99235913e-05
2.26669424e-06 2.21978695e-12 -1.72378291e-76 3.02661437e-08]

At lambda : 0.1000 and M = 35, **Test Error using SGD: 0.5741**

LeToR Data: Printing SGD Solution with early stopping...
[3.14342810e-01 1.24777863e-01 3.96417663e-41 -6.94069103e-10
1.12259775e-03 -2.42597368e-09 3.37100623e-12 -2.79362502e-02
-1.03849350e-63 -9.74041169e-02 3.45340009e-17 5.95445001e-59
-2.60150877e-08 2.03620765e-06 -4.44845293e-12 -1.11225167e-06
8.27373181e-11 3.47903211e-09 -1.75770668e-06 -2.24992992e-16
-1.79535826e-02 -2.46691965e-08 -1.17961217e-08 -1.24739300e-07
7.09784255e-05 -2.94448474e-03 1.47088066e-02 -7.69522917e-12
-4.50766776e-07 8.12704150e-05 -1.37787385e-03 5.99235913e-05
2.26669424e-06 2.21978695e-12 -1.72378291e-76 3.02661437e-08]

At lambda : 0.1000 and M = 35, **Test Error using SGD with early stopping: 0.5741**

SYNTHETIC DATA OUTPUT

For M : 11
LeToR Data: Printing Closed Form Solution...
[8.95091906e-01 -2.92259503e-01 3.49739068e+00 -1.43060626e+00
1.29644473e+01 -4.56982560e+00 -1.09094243e-01 -1.01419999e-02
2.11890913e+00 -5.28208514e+00 8.21929899e+00 1.60219116e+00]

At lambda : 0.1000 and M = 11, **Test Error using Closed Form: 0.7677**

LeToR Data: Printing SGD Solution without early stopping...
[0.89997967 -0.17534375 3.46534897 -0.81175888 5.92446217 -0.48385571
0.30189332 0.59668458 1.77420626 0.59189354 6.14484625 1.27445618]

At lambda : 0.1000 and M = 11, **Test Error using SGD: 0.7649**

LeToR Data: Printing SGD Solution with early stopping...
[0.89997967 -0.17534375 3.46534897 -0.81175888 5.92446217 -0.48385571
0.30189332 0.59668458 1.77420626 0.59189354 6.14484625 1.27445618]

At lambda : 0.1000 and M = 11, **Test Error using SGD with early stopping: 0.764**9

SOFTWARE/HARDWARE USED

- Sublime Text 3, Python 3 Environment based upon Anaconda, Ubuntu 16 System, Intel core i3 processor.
- Python libraries: numpy, openpyxl, matplotlib, scipy, sklearn

REFERENCES

1. Ublearns
2. Stackoverflow.com
3. Python, Numpy and scikit documentations.