Applications of Linear Regression in Pairwise Protein Structural Similarity Prediction

Austin Ream[2, *, +], **Rahul Alapati**[1, +], **Vineet Nayak**[1, +], **Anuj Gupta**[1, +] **and Sritika Chakladar**[1, +]

[1]Graduate Student, Computer Science and Software Engineering, Auburn University, 36830, USA [2] Undergraduate Student, Computer Science and Software Engineering, Auburn University, 36830, USA

[*] agr0012@auburn.edu

[+] these authors contributed equally to this work

ABSTRACT

As the sequence and the three-dimensional structure of more and more proteins are discovered in the laboratory and catalogued in databases, protein structure similarity prediction emerged as an important and urgent problem in computation biology. This report lays out the implementation of Linear Regression a linear modeling approach, that has been used to train a linear model to determine the similarity between two protein structures based on the position specific scoring matrices generated by PSI-BLAST, proportion of protein secondary structure classes generated by the naïve Bayes secondary structure predictor and the proportion of protein solvent accessibilities generated by the decision tree based solvent accessibility predictor. Two variants of gradient descent namely: Batch & Stochastic and MCLE estimation have been used to learn the weight vectors during model training. Accuracy of average squared error between the True and Predicted TM-Score are calculated to evaluate the classification performance of the model.

1. Introduction

A protein's 3D structure largely determines its functional properties. As a result, knowledge of the 3D structure of a protein can yield useful information about the functional properties of the protein. Structural similarity between proteins is a very good predictor of functional similarity. Since, a protein's amino acid sequence determines 3D structure, which in turn determines protein function, one might think that sequence similarity is also a very good predictor of functional similarity, but this turns out to be less important when compared with structural similarity. Vastly different amino acid sequences can yield very different structures, and similar sequences can sometimes yield dissimilar structures. Thus, sequence similarity is a far less reliable predictor of functional similarity when compared to structural similarity. Several measures of protein structure similarity have been proposed and used over the past few years, attempting to

assign to each pair of proteins a distance, presumably capturing the extent to which the two proteins resemble each other in structure, origin and function. TM-Score is an important and popular similarity measure used in protein structure similarity.

Linear regression is a way to explain the relationship between a dependent variable and one or more explanatory variables using a straight line. It is a special case of regression analysis. It transforms the input training dataset using a linear function to return a probability based upon the prediction threshold. The learning algorithm uses regression coefficient to reduce the prediction error. It discovers the best value of the coefficient based on the prediction error and iteratively updates them until the stopping criteria is met.

Gradient descent optimization is an iterative algorithm for finding the minimum of the cost function by taking steps proportional to the negative of the gradient at the current point. It has two variants: Batch and Stochastic Gradient Descent. Batch gradient descent calculates the gradient of the cost function by using the sum of the cost of each sample in every iteration. On the other hand, stochastic gradient descent is a stochastic approximation of the gradient descent optimization algorithm and it's an iterative method for minimizing the cost function over a single instance in the dataset. We use the gradient descent optimization algorithm to update the regression coefficient vector to reach convergence quickly.

In this project, we train a linear regression model to determine the similarity between two protein structures by predicting a similarity measure called TM-Score. We implement both the variants of gradient descent namely: Batch & Stochastic and MCLE estimation to learn the weight vectors during model training. We use these models to predict the TM-Score for a pair protein sequences.

2. Methods

The protein sequence pairs in Multi-FASTA format, the True TM-Score from the TM-align, the position specific scoring matrices, the relative solvent accessibilities and the secondary structure classes have been used to train and test the linear model. Gradient descent optimization algorithms and MCLE Estimation have been used to learn the weight vector. The learned weight vector is used in pairwise protein structural similarity prediction.

The input protein sequence pairs and their respective TM-Scores have been

curated into non-overlapping sets of Training (75%) and Test (25%) datasets using the simple random sampling without replacement.

Now, PSIBLAST and the nr database have been used to generate train and test PSSMs for protein sequences in train and test datasets, respectively.

BLAST (Basic Local Alignment Search Tool) is a sequence similarity search method, in which a query protein is compared to protein sequences in a target database to identify regions of local alignment and report those alignments that score above a given score threshold. Position-Specific Iterative (PSI)-BLAST is a protein sequence profile search method that builds off the alignments generated by a run of the BLAST program.

A sample PSSM for a sequence is shown below in figure 1:

>sequence
TIKVLFVDDHEMVRIGISSYLSTQSDIEVVGEGASGKEAIAKAHELKPDLILMDLLMEDMDGVEATT
QIKKDLPQIKVLMLTSFIEDKEVYRALDAGVDSYILKTTSAKDIADAVRKTSRGESVFEPEVLVKMR
NRMKKRAELYEMLTEREMEILLLIAKGYSNQEIASASHITIKTVKTHVSNILSKLEVQDRTQAVIYAF
QHNLIQ

A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
16	0	12	0	0	0	0	0	0	0	0	13	0	0	0	33	26	0	0	0
5	18	13	0	0	8	3	1	1	0	0	47	0	1	1	1	1	0	0	0
16	22	1	0	1	2	0	27	3	0	1	5	0	0	0	15	6	0	0	0
0	0	0	0	2	0	0	0	0	8	21	0	1	36	0	0	0	7	22	3
0	0	0	0	0	0	0	0	0	1	1	0	0	96	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0	41	2	0	0	0	0	0	0	0	0	56
10	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	87	0	0	1
3	0	0	0	0	0	0	97	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	98	0	0	0
0	0	0	97	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	98	0	0	0
6	2	11	29	0	0	28	23	0	0	0	0	0	0	0	2	0	0	0	0
8	0	0	0	3	0	0	0	0	17	0	0	0	0	0	3	0	0	0	68
0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	99	0	0	0
2	2	0	0	1	0	1	0	3	10	13	0	2	10	0	0	8	3	8	37
19	0	0	0	2	0	0	1	0	14	1	0	0	0	0	5	0	0	0	57
20	0	0	0	1	0	0	0	0	0	0	0	0	0	0	52	25	0	0	1
24	20	0	0	19	1	0	5	0	1	4	3	0	0	0	11	7	0	0	4
63	2	1	0	1	1	0	16	0	2	5	1	0	0	0	2	5	0	0	2
0	0	0	0	0	0	0	0	0	17	79	0	2	1	0	0	0	0	0	1
15	0	0	0	1	0	1	0	0	11	62	0	4	0	0	0	1	0	0	5
11	12	2	1	1	37	3	3	16	0	1	5	1	0	0	3	2	0	1	0
48	2	1	0	2	1	1	5	2	1	10	4	1	2	0	4	4	1	5	5
25	0	0	0	2	0	0	0	0	4	45	0	2	18	0	0	0	1	1	1
23	21	11	0	0	10	1	3	2	4	3	8	1	0	1	5	4	0	0	4
20	16	4	3	0	16	9	3	2	1	5	12	1	0	0	5	2	0	0	0
18	13	3	1	1	19	5	2	9	1	7	6	2	0	0	7	3	0	2	2
1	1	6	3	0	2	1	78	3	0	0	2	0	0	0	1	0	0	1	0
2	8	1	0	1	4	1	1	6	8	23	10	2	5	1	1	1	1	12	10
2	25	7	6	2	8	1	0	3	1	1	11	0	0	1	20	11	0	0	2
25	0	0	0	2	0	0	0	0	8	0	0	1	1	0	5	27	0	1	31

Figure 1. Sample PSSM for a sequence.

The PSSM contains 20 “weighted observed percentages rounded down” values for each residue in a protein sequence. For a protein sequence of N residues, there will be N * 20 PSSM Values. We consider the average of the 20 PSSM weighted observed percentages observed rounded down and scale them down to a scale between 0 and 1 by dividing them by 100. For a pair of proteins, we would have a total of 40 features i.e. 20 features/protein.

Now, we use the previously developed naïve Bayes protein secondary structure predictor to predict the three-class secondary structure of the protein pair from their primary sequence. Then, we calculated the proportion of helix (H), strand (E) and coil (C) for each pair, resulting in a total of 6 features i.e. 3 features/protein.

Now, we use the previously developed decision tree-based protein solvent accessibility predictor to predict the two-class solvent accessibility of the protein pair from their primary sequence. Then, we calculated the proportion of exposed (E), buried (B) for each pair, resulting in a total of 4 features i.e. 2 features/protein.

Hence, we will have a total of 50 features all in the range of [0,1) for a protein pair.

The true TM-Score between a unique pair of protein 3D structures is calculated using the TM-Align program. The average of the two TM-scores reported by the TM-align program, one normalized by length of Chain_1 and other normalized by length of Chain_2, have been considered in feature generation.

A sample training instance is shown in Figure 2. Each training instance consists of 40 PSSM values, the proportion of H, E and C for each protein (6), the proportion of Exposed and Buried for each protein (4) and the corresponding True TM-Score for the protein pair.

A sample training instance:

```
>>> datast[0]
[0.07172566371681416, 0.051548672566371675, 0.04831858407079646, 0.06690265486725663, 0.007256637168141593, 0.028008849557522125, 0.0781858407079646, 0.05553097345132743, 0.034646017699115045, 0.07668141592920354, 0.1029646017699115, 0.07115044247787611, 0.022079646017699118, 0.033230088495575225, 0.035398230088495575, 0.04420353982300885, 0.05557522123893805, 0.008761061946902656, 0.029867256637168143, 0.07473451327433628, 0.0588, 0.04855, 0.0646, 0.056799999999999996, 0.0311, 0.03275, 0.04565, 0.0603, 0.04135, 0.060250000000000005, 0.09185, 0.049749999999999996, 0.0346, 0.04125, 0.03405, 0.0748, 0.05055, 0.0134, 0.0348, 0.07165, 0.3274336283185841, 0.3141592920353982, 0.3584070796460177, 0.205, 0.33, 0.465, 0.37610619469026546, 0.6238938053097345, 0.37, 0.63, 0.30155]
```

Figure 2. Feature generation (40 PSSM Values, 6 H, E, C proportions, 4 E, B proportions and True TM-Score).

2.1. Linear Regression Learning on Training Set:

Initially, we implemented the Stochastic gradient descent-based optimization algorithm and MCLE estimation to learn the weight vector of Linear Regression using the training set.

In Stochastic Gradient Descent, we begin by selecting a single training instance for each iteration using simple random sampling with replacement and initially all the weights are set to zero. In every iteration, we calculate a prediction based on the linear function using the training instance and the weight vectors as follows:

$$P(X) = \left(w_0 + \sum_{j=1}^{n} w_j x_j^{\ell} \right)$$

where, P(x) = Probability estimate

The weight vectors are updated in every iteration using the MCLE estimation as follows:

$$\forall i \quad \text{repeat}$$

$$w_i \leftarrow w_i + \eta 2 \sum_{\ell} x_i^{\ell} \left(Y^{\ell} - \left(w_0 + \sum_{j=1}^{n} w_j x_j^{\ell} \right) \right)$$

$$\text{assume} \quad x_0 = 1 \quad \text{for} \quad w_0$$

where,

w_i : Weights
η : learning rate
X_i : Features from the selected training instance
Y : Prediction
Assumption: $X_0 = 1$ for W_0

The learning algorithm uses gradient descent to discover best values of regression coefficients based on the error in prediction on training data in an iterative process, until convergence is reached. Given a dataset, we want to find the parameter vector '**w'** which maximizes the likelihood.

During the training, we use a variable learning rate (η) which follows an exponential function. The above process is repeated until the error in our prediction doesn't change over a few iterations, i.e. change in the weights is less than ϵ = 0.0005 for 10 iterations.

Also, we have implemented batch gradient descent optimization algorithm in combination with MCLE estimation to learn the weight vector of Linear Regression using the training set.

In batch gradient descent, we begin by selecting all the instances in the training dataset and repeat the above-mentioned process of learning weights.

2.2. Linear Regression Classification on Test Set:

For classification, we use the weights learned as a part of training the linear model and the linear probability function to determine the TM-Score between the unique pair of protein 3D structures.
The TM-Score would be 0 if the proteins don't have any similarity between them, 1 when they are perfectly similar to each other, [0,1) otherwise.

The prediction is calculated using the following equation:

$$P(X) = \left(w_0 + \sum_{j=1}^{n} w_j x_j^{\ell} \right)$$

where, P(X) = Probability estimate

Accuracy:

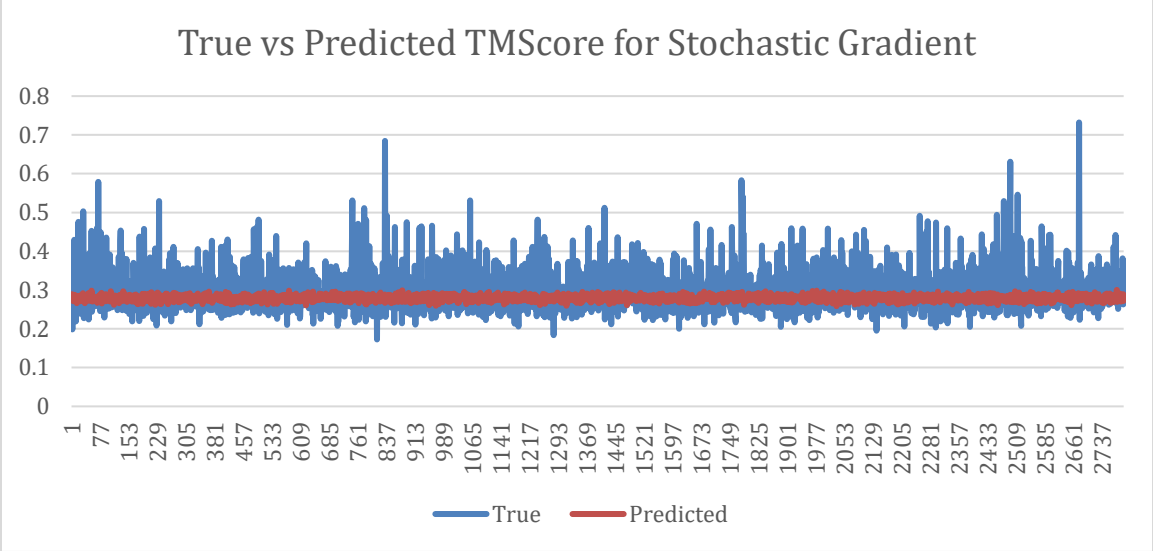
The accuracy of average squared error between True and Predicted TM-Score i.e. (true TM-score – predicted TM-score)^2 is calculated on the test set.

3. Results

The following is a sample output showing TM-Score between a pair of protein sequences.

```
anuj@anuj-pc:~/ext_storage/mystuff/studies/Auburn/08 COMP 7970 Computational Biology/gitrepo/project5/group1/src$ ./proj
ect5.py ../../../project3/apps/blast/bin/blastpgp /tmp/nr/nr weights_training_stochastic_lr_function.pickle ../../data/1
a3a.fasta ../../data/1guu.fasta
0.142270023252
```

The following figure shows the relation between true and predicted TM-Score calculated using the linear model trained using **Stochastic Gradient Descent:**



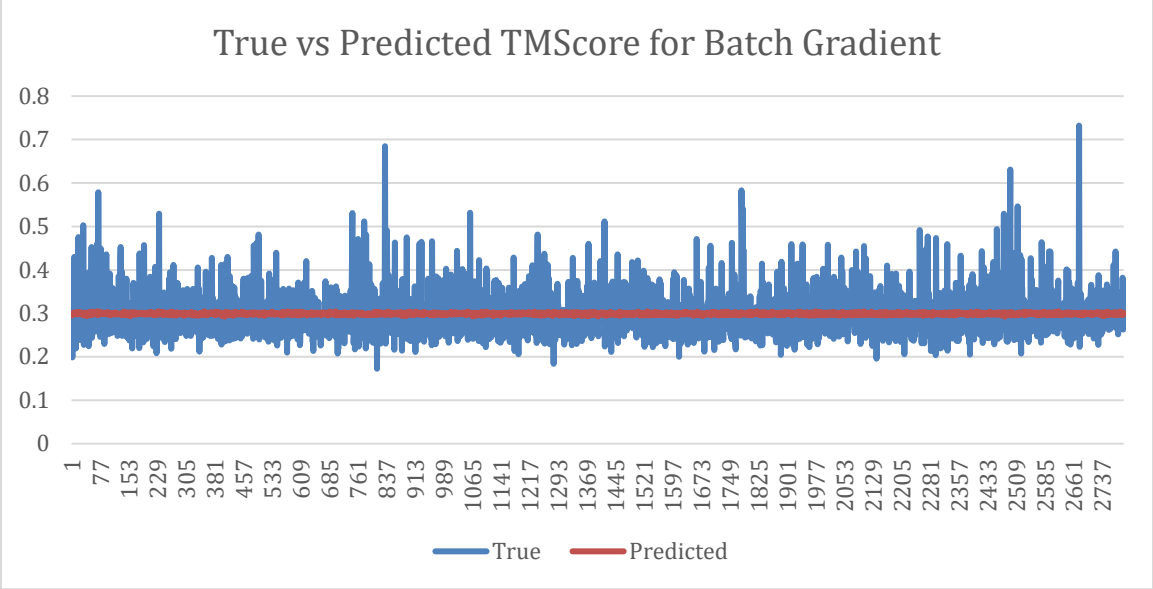
The **Pearson correlation** between True and Predicted TM-Score in case of **stochastic gradient descent** is **0.199420869** and the **average squared error** is **0.00280689756911**. The low Pearson correlation between the True and Predicted TM-Score is due to the fact that the True TM-Score are always in the range of 0.2 to 0.4 and as a result of that, the trained model always predicts TM-Score in the range of 0.2 to 0.3.

The following are the values of the weights learned using stochastic gradient descent:

Stochastic																				
0.10222649	0.016506	0.001207	0.000343	0.008278	-0.00374	0.003649	0.006788	0.007188	0.004108	0.006413	0.016146	0.003705	0.003614	0.006197	-0.00086	0.005821	0.004822	-0.00089	0.000348
0.01216988	0.019104	0.005981	0.000708	0.004437	-0.00585	0.003798	0.004649	0.005834	0.00266	0.006318	0.019899	0.002072	0.003798	0.003669	0.004018	0.00745	0.004013	-0.0008	0.001225
0.0084042	0.061681	0.00562	0.034925	0.060486	0.017281	0.02446	0.038801	0.063425	0.033344	0.068882									

The W0 bias weight 0.10222649 dominates all the other 50 weights.

The following figure shows the relation between true and predicted TM-Score calculated using the linear model trained using **Batch Gradient Descent:**



The **Pearson correlation** between True and Predicted TM-Score in case of **batch gradient descent** is **-0.005268338** and the **average squared error** is **0.00249980835431**. The negative Pearson correlation between the True and Predicted TM-Score is due to the fact that the True TM-Score are always in the range of 0.2 to 0.4 and as a result of that, the trained model always predicts TM-Score in the range of 0.2 to 0.3.

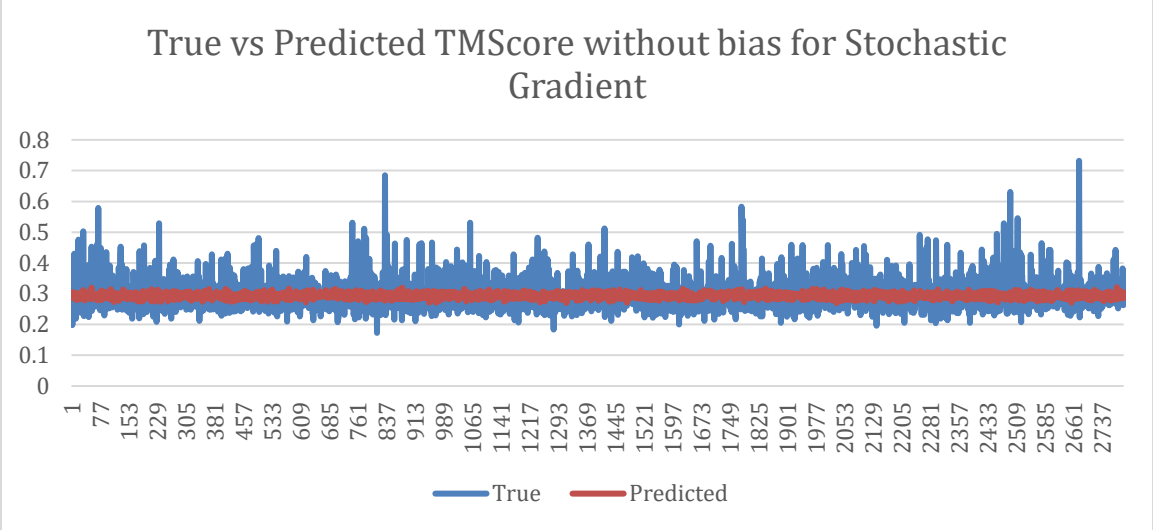
The following are the values of the weights learned using batch gradient descent:

Batch																				
0.10354398	0.008703	0.005283	0.003984	0.006322	0.001608	0.003913	0.007522	0.007719	0.002414	0.006157	0.009209	0.006438	0.002289	0.004003	0.004166	0.005817	0.005379	0.001242	0.003225
0.0078316	0.008478	0.005712	0.003942	0.006255	0.001468	0.003894	0.007291	0.007579	0.002307	0.006325	0.009521	0.006568	0.002402	0.004009	0.00443	0.005713	0.005255	0.001131	0.003126
0.00783962	0.03273	0.038256	0.032558	0.032736	0.037802	0.033006	0.035206	0.068338	0.035801	0.067743									

The W0 bias weight 0.10354398 dominates all the other 50 weights.

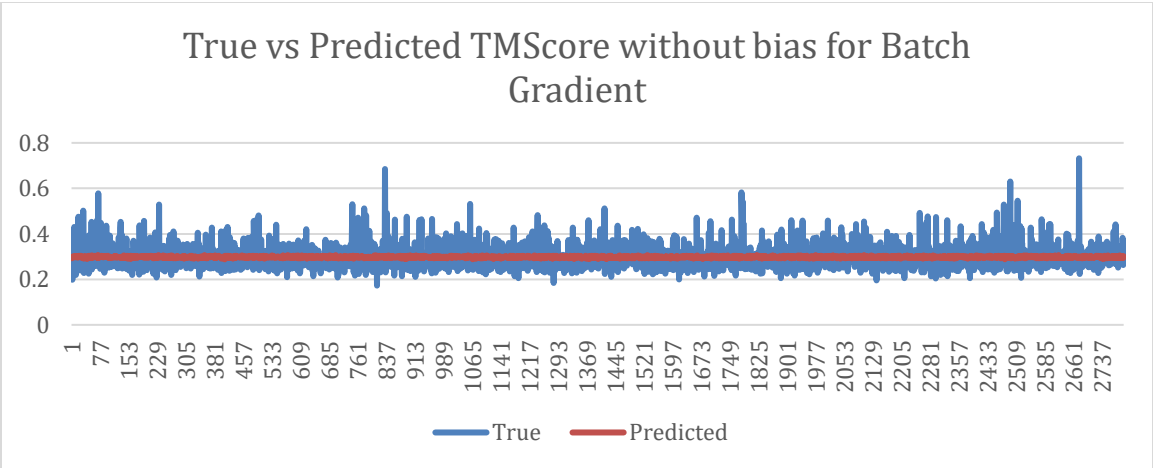
We performed an experiment, by ignoring the bias weight W0, while training the model to nullify the W0 domination in TM-Score prediction.

The following figure shows the relation between true and predicted TM-Score calculated using the linear model trained using **Stochastic Gradient Descent without Bias:**



The **Pearson correlation** between True and Predicted TM-Score in case of **stochastic gradient descent without bias** is **0.208526353** and the **average squared error** is **0.00242337981829**.

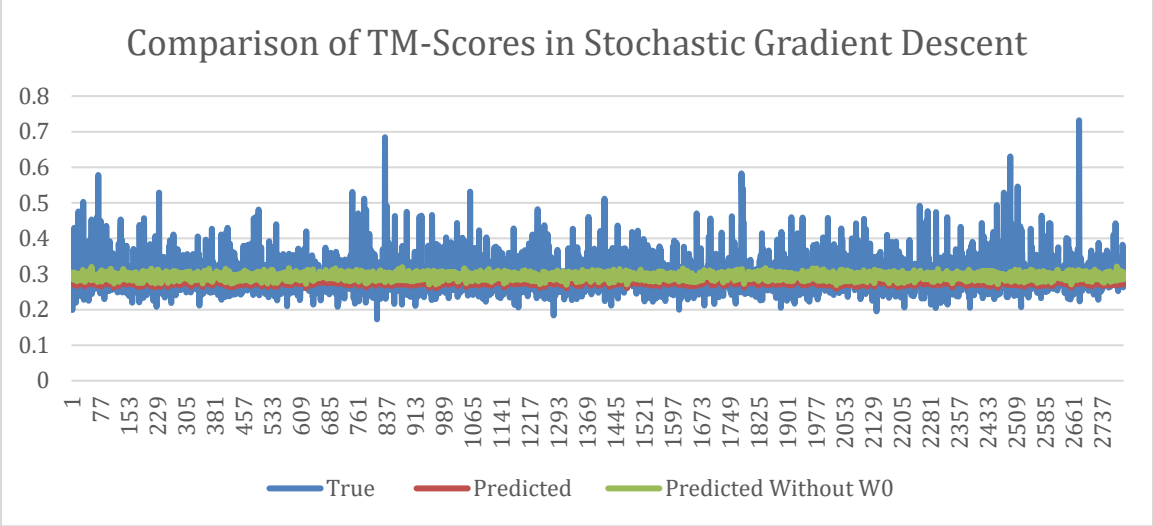
The following figure shows the relation between true and predicted TM-Score calculated using the linear model trained using **Batch Gradient Descent without Bias:**



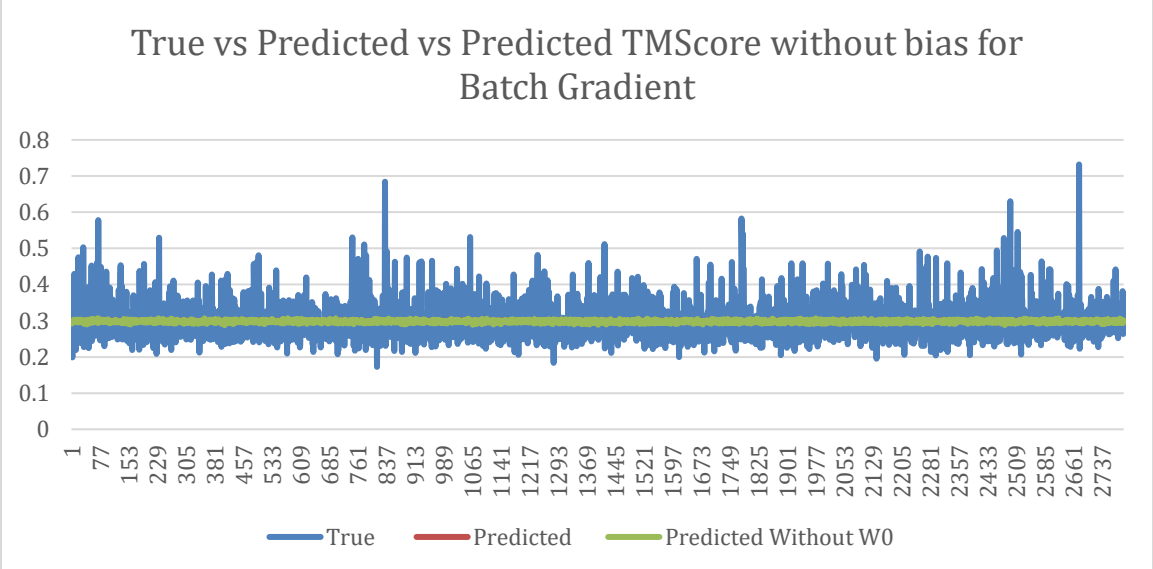
The **Pearson correlation** between True and Predicted TM-Score in case of **batch gradient descent without bias** is **-0.01191422** and the **average squared error** is **0.00250920669869**.

But, even after ignoring the bias weight, the Pearson correlation is still low. This is due to the fact that the True TM-Scores are all in the range of 0.2 to 0.4 and due to that our trained model is forced to predict TM-Scores in the range of 0.2 to 0.3.

The following figure shows the relation between True, Predicted TM-Scores with and without the bias in case of **Stochastic Gradient Descent**:



The following figure shows the relation between True, Predicted TM-Scores with and without the bias in case of **Batch Gradient Descent**:



From the above plots, we can observe that the predicted TM-Scores are always in the range of 0.2 to 0.3 and the bias weight dominance has no effect on the prediction.

5. Discussion

We performed different experiments as discussed above to improve the accuracy and the Pearson correlation between True and Predicted TM-Scores predicted using our linear regression model. Our Pearson correlation of 0.20 was the highest in the case of Stochastic Gradient Descent. Also, our average squared error was consistently around 0.002.

5. References

1. Wikipedia contributors. (2018, March 21). Protein structure. In Wikipedia, The Free Encyclopedia. Retrieved 19:23, March 26, 2018, from https://en.wikipedia.org/w/index.php?title=Protein_structure&oldid=831674619
2. Pauling, L., Corey, R. B., & Branson, H. R. (1951). The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. Proceedings of the National Academy of Sciences, 37(4), 205-211.
3. Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., & Walter, P. (2002). The shape and structure of proteins.
4. Anfinsen, C. B. (1973). Principles that govern the folding of protein chains. Science, 181(4096), 223-230.
5. Zhang, H. (2004). The optimality of naive Bayes. AA, 1(2), 3.
6. Schmidler, S. C., Liu, J. S., & Brutlag, D. L. (2000). Bayesian segmentation of protein secondary structure. Journal of computational biology, 7(1-2), 233-248.
7. Robles, V., Larrañaga, P., Peña, J. M., Menasalvas, E., Pérez, M. S., Herves, V., & Wasilewska, A. (2004). Bayesian network multi-classifiers for protein secondary structure prediction. Artificial Intelligence in Medicine, 31(2), 117-136.
8. Di Lena, P., Nagata, K., & Baldi, P. (2012). Deep architectures for protein contact map prediction. Bioinformatics, 28(19), 2449-2457.
9. Tress, M. L., & Valencia, A. (2010). Predicted residue–residue contacts can help the scoring of 3D models. Proteins: Structure, Function, and Bioinformatics, 78(8), 1980-1991.
10. Yang Zhang, Jeffrey Skolnick. TM-align: a protein structure alignment algorithm based on the TM-score. Nucleic Acids Research, Volume 33, Issue 7, 1 January 2005, Pages 2302–2309.