**Implementing Logistic Regression for Protein Contact Map prediction**

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

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

\* szc0098@auburn.edu

+these authors contributed equally to this work

**ABSTRACT**

Protein structure predictions from its amino acid sequence is useful in the study of protein functions. Protein contacts hold vital information for the understanding of the 3-dimensional protein structure. The protein contact prediction refers to the problem of predicting the spatial closeness of residue pairs in a folded protein structure. Machine learning approaches predict the probability of contact from a training dataset of protein structure and work quite well in predicting the outcome.

Our approach models the logistic regression algorithm to learn from the training dataset of protein sequence and its feature vector. We use the gradient ascent based optimization algorithm to learn the weight vector of logistic regression using the training set. Also, we compare the prediction accuracy of our implementation of stochastic gradient ascent with mini-batch gradient ascent algorithms.

**1. Introduction**

Protein structure is the three-dimensional arrangement of atoms in an amino acid-chain molecule1. There are different distinct levels of protein structure. The [primary structure](https://en.wikipedia.org/wiki/Primary_structure) of a protein refers to the sequence of amino acids in the polypeptide chain. Tertiary structure refers to the three-dimensional structure of monomeric and multimeric protein molecules. Secondary structure refers to highly regular local sub-structures on the actual polypeptide backbone chain.

Protein folding is the physical process by which a polypeptide folds into its characteristic and functional three-dimensional structure from random coil3. The primary structure of a protein, its linear amino-acid sequence, determines its native conformation4. The specific amino acid residues and their position in the polypeptide chain are the determining factors for which portions of the protein fold closely together and form its three dimensional conformation. Formation of a secondary structure is the first step in the folding process that a protein takes to assume its native structure. Characteristic of secondary structure are the structures known as α-helices and β-sheets2 that fold rapidly because they are stabilized by intramolecular hydrogen bonds. α-helices are formed by hydrogen bonding of the backbone to form a spiral shape. The β pleated sheet is a structure that forms with the backbone bending over itself to form the hydrogen bonds.

Prediction of a protein structure from its amino acid sequence is an important and challenging work. Not only can successful predictions provide a starting point for direct tertiary structure modelling, but they can also significantly improve sequence analysis and sequence-structure threading for aiding in structure and function determination7.

Protein contacts hold vital information for the understanding of the 3-dimensional protein structure. The protein contact prediction refers to the problem of predicting the spatial closeness of residue pairs in a folded protein structure. Contacts occurring between sequentially distant residues, i.e. long-range contacts, impose strong constraints on the 3D structure of a protein and are particularly important for structural analyses, understanding the folding process and predicting the 3D structure8. Even a small set of correctly predicted long-range contacts can be useful for improving ab initio structure prediction for proteins without known templates9.

Logistic regression is a classification algorithm which is used to assign observations to a discrete set of classes. It transforms the input training dataset using a logistic function to return a probability which is mapped to discrete response classes 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 ascent optimization is an iterative algorithm for finding the maximum of the cost function by taking steps proportional to the positive of the gradient at the current point. It has two variants: Batch and Stochastic gradient Ascent. Batch gradient ascent 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 ascent is a stochastic approximation of the gradient ascent optimization algorithm and it’s an iterative method for maximizing the cost function over a single instance in the dataset. We use the gradient ascent optimization algorithm to update the regression coefficient vector to reach convergence quickly.

In this project, we model the logistic regression algorithm to learn from the training dataset of protein sequence and its feature vector. We use the gradient ascent based optimization algorithm to learn the weight vector of logistic regression using the training set. We implement stochastic gradient ascent using Maximum Conditional Likelihood Estimation (MCLE) to train our model. Also, we compare the prediction accuracy of our implementation of stochastic gradient ascent with mini-batch gradient ascent algorithms.

**2. Methods**

The protein sequences in FASTA format and the corresponding .rr file holding residue-residue contact information have been used to implement the logistic regression algorithm on training set and uses the learned weight vector to calculate the probability all pairs of residue pairs on test dataset.

We use the training dataset {<Xi , Yi>}, where Xi consists of values of amino acid attributes and Yi is the corresponding labels i.e. 0 for non-contact and 1 for contact to train our learning model. A sigmoid logistic function is used for mapping predicted values to probabilities. We performed stochastic gradient ascent using MCLE on the training dataset to update the regression coefficient iteratively, until convergence. The weight coefficient model and sigmoid logistic function is then used to classify the test dataset. Finally, we verify our predictions against the test data labels to measure the accuracy.

The input protein sequences .fa file and their respective .rr files have been curated into non-overlapping sets Training (75%) and Test (25%) datasets using simple random sampling with replacement.

**2.1. Feature Vector Generation**

We used NCBI PSIBLAST and the nr database, to generate PSSMs for the dataset. The input FASTA files are passed to the PSIBLAST and the corresponding PSSMs are generated. From the PSSM we will have 20 PSSM Values for each residue in a protein sequence, i.e. for a protein sequence of length N, we will have N\*20 PSSM values.

We use a sliding window of 5 around the central residue for feature generation. For this, we consider the PSSM values of two residues above and the PSSM values of two residues below the central residue, to generate a feature vector of 100 PSSM values for the central residue. For terminal residues, we consider rows of 20 values of -1 for generating the feature vector.

Now, we combine 100 PSSM values for each pair of residues (i, j), where j > i +5, in a protein sequence i.e. for a protein sequence of length N, we will have N\*200 PSSM values. After the N \* 200 feature vectors are generated for all the vectors, we will use the .rr file to determine the label of the pair of residues. The .rr file lists the pair of residues that are in contact, i.e. the distance between their C-beta atoms (C-alpha in case of glycine) is less than 8 Angstroms (Å). We consider all other pairs to be in non-contact and add a label of zero.

**2.2. Logistic Regression Learning on Training Set**

The logistic regression model takes real-valued feature vector as input and makes a prediction as to the probability of the input belonging to the response classes (class 0 for non-contact or class 1 for in contact). If the probability of prediction is greater than threshold (=0.5), we can take the output as a prediction for one response class (class 1), otherwise the prediction is for the other class (class 0). The output is transformed into a probability using the logistic function.

The logistic function used is:

P(z) = Probability estimate

z = input to the function (𝑤0 + ∑i wixi )

e = base of natural log

The learning algorithm uses gradient ascent 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:

* + ηXi(Y –P(Z))
* : Weights
* η : learning rate
* Xi  : Features from the selected training instance
* Y : Prediction
* P(z) = Probability estimate

Our assumption here is X0 = 1 for Wo.

We implemented stochastic gradient ascent using MCLE to train our model.

We begin by selecting a single training instance for each iteration using simple random sampling with replacement and initializing the weight vectors to zero. In every iteration we use the cost gradient of a single instance. Given each training instance, we calculate a prediction using the current values of the weight coefficients and calculate new coefficient values based on the error in the prediction. During the training process, we use a variable learning rate (η) which follows an exponential function and varies from 0.045 to 0.009405, i.e. the learning rate is high at higher gradient value and reduces as gradient decreases and we are closer to the maxima.

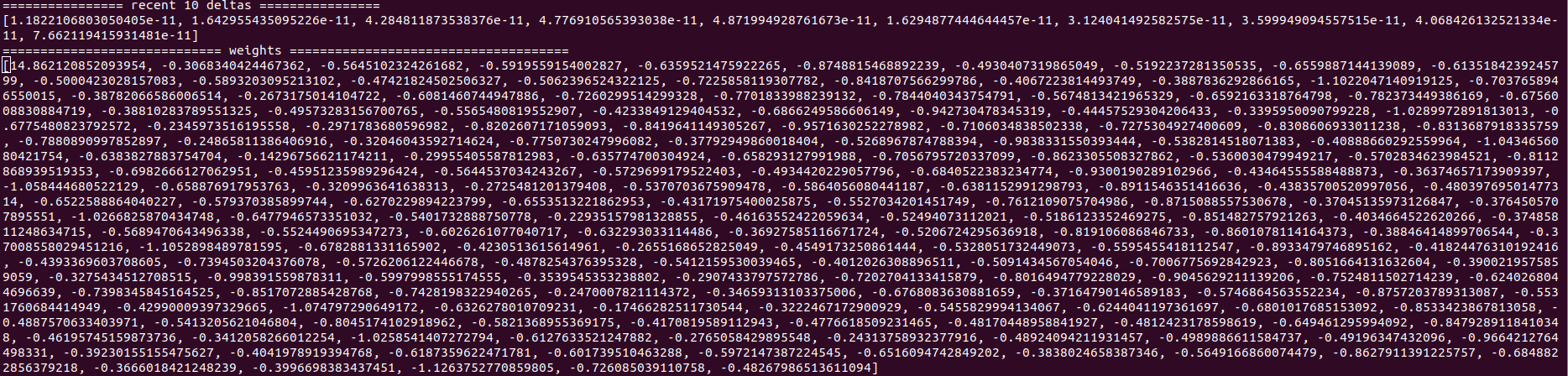
This process is repeated until the error in our prediction doesn’t change over a few iterations, i.e. the change in the weight coefficient is less than ϵ = 0.0005, for 10 iterations

**2.3. Logistic Regression Classification on Test Set**

We classify each pair of residue in the test dataset as in contact (1) or non-contact (0) by using the weight model derived from the training dataset. For determining the test label, we use the prediction function which takes as input the weight model from the training data and features of test dataset. The predicted probability is compared to the actual label derived from the corresponding .rr file. Accuracy was calculated on the top L/10, L/5, L/2 predicted contacts to evaluate the classification performance averaged over the proteins in the test dataset. Here, L is the length of the protein sequence and top contacts are the contacts predicted with high probabilities

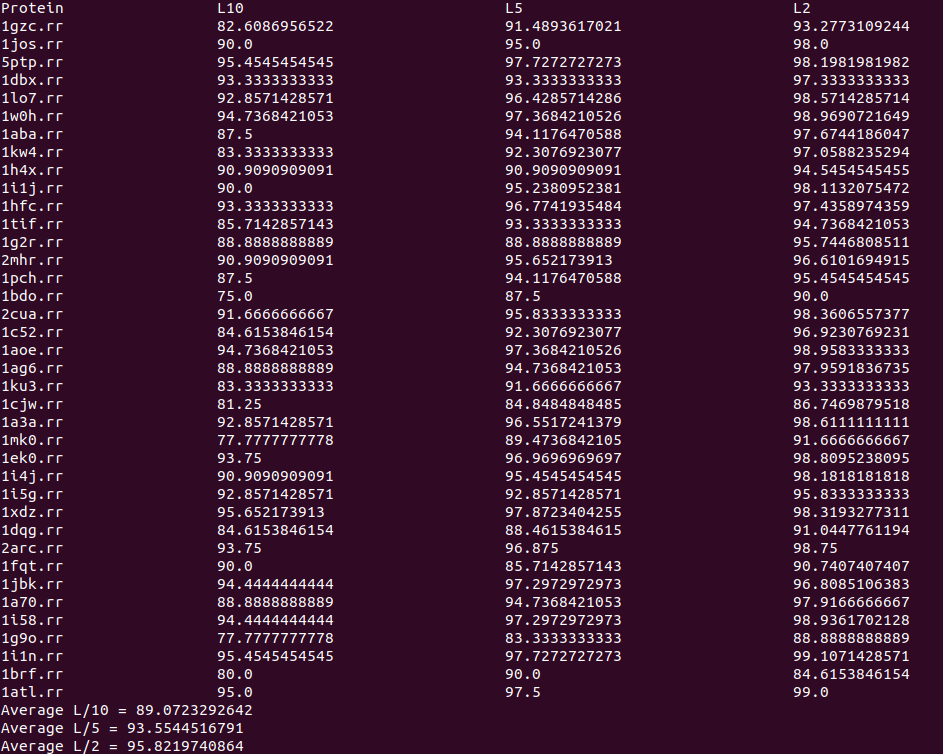
**3. Results**

The output of our training model is a learned weight vector which is used to calculate the probability all pairs of residue pairs (i, j) to be in contact given a single FASTA formatted test protein sequence.



**Figure 1.** Weight vector with balanced classes using Stochastic gradient ascent. ϵ = 0.0005 and η = variable from 0.045 to 0.009405

The prediction accuracy calculated is shown below:



**Figure 2.** Prediction accuracy using stochastic gradient ascent. ϵ = 0.0005 and η = variable from 0.045 to 0.009405

**Table 1.** Prediction accuracy using stochastic gradient ascent.

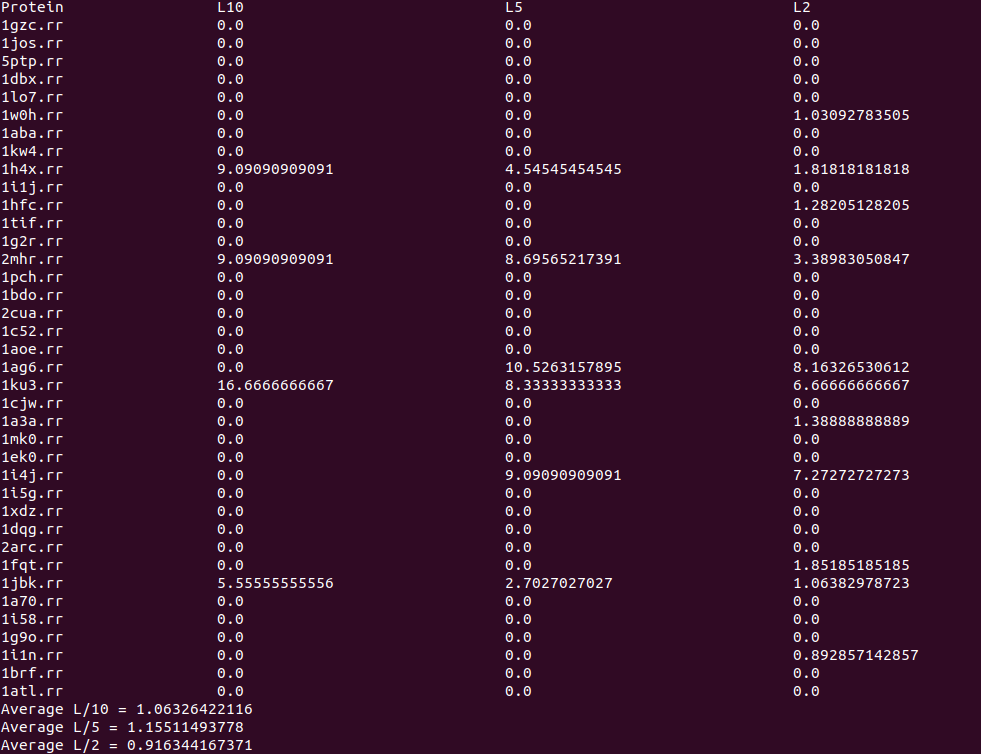
|  |  |
| --- | --- |
| Predicted Contacts | Accuracy |
| L/10 | 89.07 |
| L/5 | 93.55 |
| L/2 | 9.82 |

To compare our results, we performed a mini-batch gradient ascent using a batch size of 500 on the training dataset.



**Figure 3.** Weight vector with balanced classes using mini-batch gradient ascent. ϵ = 0.0005, batch size = 500 and η = 0.01.

The prediction accuracy calculated is shown below:



**Figure 4.** Prediction accuracy using mini-batch gradient ascent. ϵ = 0.0005, batch size = 500 and η = 0.01.

**Table 2.** Prediction accuracy using min-batch gradient ascent.

|  |  |
| --- | --- |
| Predicted Contacts | Accuracy |
| L/10 | 1.06 |
| L/5 | 1.15 |
| L/2 | 0.91 |

**4. Discussion**

**Class imbalance** : Since the number of non-contact residue pairs was quite high we had a class imbalance and the prediction accuracy was 97%. **Normalization of data set :** The PSSM feature values are not normalized. So without normalizing the feature set it became difficult to determine the learning rate and stopping criteria.**Batch gradient ascent:** To perform the batch gradient ascent, we need to calculate the gradient of the cost function by summing the cost of each sample. So if we have a million samples, we have to loop through a million times just to move a single step towards the maximum.

**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.