Stanislav Listopad
67644205
CS 284A
Fall 2016

Secondary Protein Structure Prediction With Logistic Regression and Random Forest Classifier

Abstract

Secondary protein structure prediction can be successfully accomplished utilizing machine learning algorithms. In this project, logistic regression and random forest classifier methods were used to predict secondary protein structure of an input amino acid sequence. Both methods were able to provide over 60% accuracy. Best accuracy was achieved when utilizing a large number of input samples (>100), large sliding window (k >= 17), and all available features (amino acid sequence label, N- and C- terminals, relative and absolute solvent accessibilities, and amino acid sequence profiles).

1. Introduction

The goal of this project is to predict secondary protein structure of an amino acid sequence. Additional information, such as amino acid profiles, N- & C-terminals, and relative & absolute solvent accessibility, can be used to enhance the prediction accuracy rate.

2. Methodology

The following *general methodology* is used to predict the secondary protein structure:

1) Analyze the dataset

2) Select a machine learning (ML) algorithm

3) Shape the dataset for the selected ML algorithm

4) Approach the problem in 4 stages:

A) Single Feature & 1 to 1 Mapping

B) Multi Feature & 1 to 1 Mapping

C) Single Feature & K to 1 Mapping

D) Multi Feature & K to 1 Mapping

5) Fine tune the algorithm (regularization, # of samples, and etc…)

6) Optional: Repeat steps 2-5 with a different ML algorithm

1) The raw dataset (cullpdb+profile_6133.npy.gz) has dimensions 6133 x 39900. There are 6133 proteins in the dataset. Each protein is assigned 700 amino acid labels. There are 57 features recorded for each amino acid. The amino acids (700) and their features (57) are combined together to form all the values that describe a single protein (700x57 = 39900).

- Features 0 through 21 are amino acid labels: A, C, E, D, G, F, I, H, K, M, L, N , Q, P ,S, R, T, W, V, Y, X, NoSeq. Only one of these labels is one and the rest are zero for any given amino acid in any given protein (one hot representation). These features [0, 21] must be used in input for secondary structure prediction.
- Features 22 through 30 are secondary structure labels: L, B, E, G, I, H, S, T, NoSeq. Only one of these labels is one and the rest are zero for any given amino acid in any given protein (one hot representation). These features describe the output of the ML algorithm.
- Features 31 through 56 are N- and C- terminals [31, 32], relative and absolute solvent accessibility [33, 34] and amino acid sequence profiles [35, 56]. These features can be used to improve secondary protein structure prediction.

2) I selected **logistic regression** (LR) as my first machine learning algorithm. Logistic regression is fast, simple to implement, and it can be accurate depending on the problem.

3) I have reshaped the dataset in two ways.

- Change the 2D raw data matrix (6133 x 39900) into a 3D matrix (6133 x 700 x 57). The data is easier to manipulate in this format. Each dimension has a clear meaning: 6133 proteins, 700 amino acids, 57 features.
- Change the one-hot-representation for amino acid labels and secondary protein structure labels into decimal representation. The first 22 features are replaced by a single feature: a decimal value that ranges from 0 to 21. The features 22 through 30 are replaced by a single decimal value that ranges from 0 to 8.

The resulting dataset is a 6133 x 700 x 28 matrix. The first index is an amino acid label. The second index is a secondary structure label. The remaining features are N- & C- terminals, relative and absolute solvent accessibility, and amino acid sequence profiles. I saved this dataset as "cullpdb+profile_6133_3D_decimal.npy" file.

Additionally, I have removed the no-sequence values from testing data when running machine learning algorithms in order to provide meaningful accuracy scores.

4) *Single Feature* approaches (4A & 4C in *general methodology*) rely solely on the first feature (amino acid label) to predict secondary protein structure.

Features [2, 27] are used in conjunction with amino acid label [0] in *Multi Feature* approaches (4B & 4D in *general methodology*).

1 to 1 Mapping approaches (4A & 4B in *general methodology*) use one amino acid sequence label to predict one protein secondary structure label.

K to 1 Mapping approaches use (4C & 4D in *general methodology*) use K neighboring amino acid sequence labels to predict one secondary protein structure label.

Approach 4A is the least informed approach. In this approach a single feature (one amino acid label) is used to predict one secondary structure label.

Approach 4B is more informed. In this approach 27 features are used to predict one secondary structure label.

Approach 4C utilizes K amino acid sequence labels (K features) to predict one secondary structure label. The K value has an impact on quality of prediction.

Approach 4D utilizes K amino acids' features. Each amino acid has an amino acid label & 26 other features. Therefore, each amino acid has 27 features. A total of K * 27 features are used to predict one secondary structure label.

5) Machine learning algorithm parameters (ex: number of input & test samples) affect the quality of secondary structure prediction. Best values for these parameters can be found through testing.

6) I have also predicted protein secondary structure using random forest classifier with decimal dataset.

3. Results & Discussion

For the following **logistic regression** benchmarks assume number of input samples (N) is 100, the number of test samples (M) is 272, and regularization constant C is 1.

Logistic Regression 4A:

Accuracy: 31.93% Runtime: 1.016s

Stanislav Listopad
67644205
CS 284A
Fall 2016

Logistic Regression 4B:

Accuracy: 46.69% Runtime: 10.26s

Logistic Regression 4C:

K = 3; Accuracy: 35.08% Runtime: 2.16s

K = 17; Accuracy: 36.16% Runtime: 12.54s

Logistic Regression 4D:

k = 3 Accuracy: 52.48% Runtime: 31.72s

k = 17 Accuracy: 60.91% Runtime: 243.53s

Additional features such as amino acid sequence profiles significantly improve prediction accuracy. Best performance is achieved when additional features are combined with the sliding window (K to 1 mapping) method.

Performance of *single feature* **logistic regression** methods with varying N (10, 100, 1000) and C (0.1, 1.0, 10, 100) values.

Single Feature LR	N = 10	N = 100	N = 1000
K = 1, M = 272, C = 1.0	29.9%	31.9%	34.17%
K = 3, M = 272, C = 1.0	31.9%	34.6%	35.13%
K = 17, M = 272, C = 1.0	33.5%	36.03%	36.28%

Accuracy improves as number of training samples is increased.

Single Feature LR	C = 0.1	C = 1.0	C = 10	C = 100
K = 1, M = 272, N = 100	29.9%	31.9%	34.17%	35.4%
K = 3, M = 272, N = 100	32.6%	34.6%	35.08%	35.15%
K = 17, M = 272, N = 100	35.5%	36.03%	36.16%	36.17%

Accuracy improves to a point as regularization parameters are increased. The regularization parameter is inverse of the regularization penalty. This suggests that for *single feature* LR methods overfitting is not a significant concern.

Multi Feature LR	C = 0.1	C = 1.0	C = 10	C = 100
K = 3, M = 272, N = 100	52.38%	52.48%	52.47%	52.45%

However, this is not the case for *multi feature* LR methods. The number of input features for these methods is K * 27. Therefore, overfitting is a more significant concern. Large C values can decrease performance of *multi feature* LR methods due to overfitting.

Best **logistic regression** run (M = 272, N = 500, C =5.0, K =17, Multi Feature)

Accuracy: 61.86% Runtime: 1070s

Stanislav Listopad
67644205
CS 284A
Fall 2016

For the following **random forest** benchmarks assume number of samples (N) is 100, number of test samples (M) is 272, and number of trees in a forest (n_estimators) is 100.

Random Forest 4A:

Accuracy: 36.2% Runtime: 1.062s

Random Forest 4B:

Accuracy: 44.8 % Runtime: 10.1s

Random Forest 4C:

K = 3; Accuracy: 41.37% Runtime: 1.48s

K = 17; Accuracy: 41.37% Runtime: 6.039s

Random Forest 4D:

k = 3 Accuracy: 54.31% Runtime: 19.58s

k = 17 Accuracy: 61.2% Runtime: 64.02s

Similarly to logistic regression, the random forest algorithm performs best with all available features and a large sliding window (K = 17) are used.

Performance of *single feature* **random forest** methods with varying n_estimator (10, 100, 1000) and N (10, 100, 1000) values.

Single Feature RF	n_estimator=10	n_estimator=100	n_estimator=1000
K = 1, M = 272, N = 100	37.9%	38%	37.9%
K = 3, M = 272, N = 100	41.5%	41.4%	41.4%

K = 17, M = 272, N = 100	37.6%	43.3%	44%

Accuracy improves as the number of trees used in random forest is increased.

Single Feature RF	N = 10	N = 100	N = 1000
K = 1, M = 272, n_estimator = 100	35.6%	37.9%	38.23%
K = 3, M = 272, n_estimator = 100	33.9%	41.4%	42.6%
K = 17, M = 272, n_estimator = 100	37.6%	43.3%	46%

Accuracy improves as the number of input samples to the random forest classifier is increased.

Best **random forest** run (M = 272, N = 500, n_estimator =100, K =17, Multi Feature) Accuracy: 62.04% Runtime: 462s

4. Conclusion

The random forest classifier has a slightly higher accuracy and a significantly smaller runtime than logistic regression when predicting protein secondary structure. Both methods are capable of achieving >60% accuracy.