Sequence-Based Protein Classification:

a Naive Bayes Approach

Yijing Li / Zhenghong Ma / Weixin Li / Weinan Li

Abstract

The objective of this project is to build a model that enables classifying a protein into certain categories accurately and efficiently based solely on its sequence of amino acids. We acquired our data from the RCSB Protein Data Bank[1]. After selecting top-10 categories of proteins based on their functions, we managed to convert the amino acid sequence data into numerical arrays with Count-Vectorization method. The framework we used for classification is based on Naive Bayes, and we reached a best prediction accuracy of 89.2%. Furthermore, we have compared it and discussed with the results from other modern Machine Learning methods such as Adaboost[2] and Random Forest.

Background

Nowadays, high-throughput biological sequencing becomes faster and more economical. However, the efficiency of extracting key information is limited by low--throughput experimental characterization of proteins' properties such as x--ray crystallography (XRC) and cryo--TEM [3] which classify proteins without utilizing sequences of proteins. In order to take advantage of the mass information of sequences [4] and accelerate the process of experimental characterizations, our team decided to apply machine learning techniques for the classification [5] of protein function directly from primary sequence.

The advantages of using Machine Learning for classification are obvious: high-speed and low-labor-cost compared to the traditional experimental methods. Classifying proteins according to sequences enables machines to do classification once they get the data instead of using the experimental method that requires waiting for the process of chemical reactions. By utilizing computers without the supervision of scientists or researchers, we reduced the labor costs significantly.

Technical Roadmap

Data Analysis

1. Data Preparation

We used a protein dataset retrieved from Research Collaboratory for Structural Bioinformatics (RCSB) Protein Data Bank (PDB). It contains protein metadata which includes details on protein classification, extraction methods, experimental technique, etc. However, in this project we mainly focused on the sequence data. From the dataset, we observed that there are thousands of kinds of protein and many of them only take a small portion of the total. For a more straightforward solution to the problem, we only took top10 categories of proteins within the data, namely, hydrolase, transferase, oxidoreductase, immune system, lyase, hydrolase/hydrolase inhibitor, transcription, viral protein, transport protein and virus. Nevertheless, even the top10 categories of proteins do not possess uniform distribution and an imbalanced dataset could often cause the precision metric to be meaningless [6]. Thus, in the next step we up-sampled the minorities to make each category has the same amount of sequences. In the following processes, the data was separated as 90% of training set and 10% of the test set. And separate multi sequence of one protein as different sequence input.

2. Vectorization

One of the vital stages in our pipeline is converting a sequence of amino acids of data into features that are readable to machine learning models. Here we adopt the Count-Vectorization method that is frequently utilized in NLP problems to vectorize the sequences. Given a sequence of amino acids, it counts the frequency of each amino acid and then output a vector of frequency. However, the drawback of this approach is obvious, it failed to address the connection between adjacent amino acids and only the count of each amino acid will impact the output. In proteins, it is not the individual amino acids that determine the characteristics of the protein. We must also consider the secondary and tertiary structures that are formed via the bonds of amino acids, in other words, the arrangement of amino acids does matter. Hence, we need to take more than one of the continuous amino acids as our frequency counter, or short sequence, in this approach. By increasing this range, we make the kinds of short sequence much greater and hence result in sparser vector. The figure.1 below illustrates the vectorization process. In range size greater than 1, the features would be the count of short sequences like ‘AA, AM, AH’. Our analysis afterwards includes the investigation relationship of this range with the model performance.

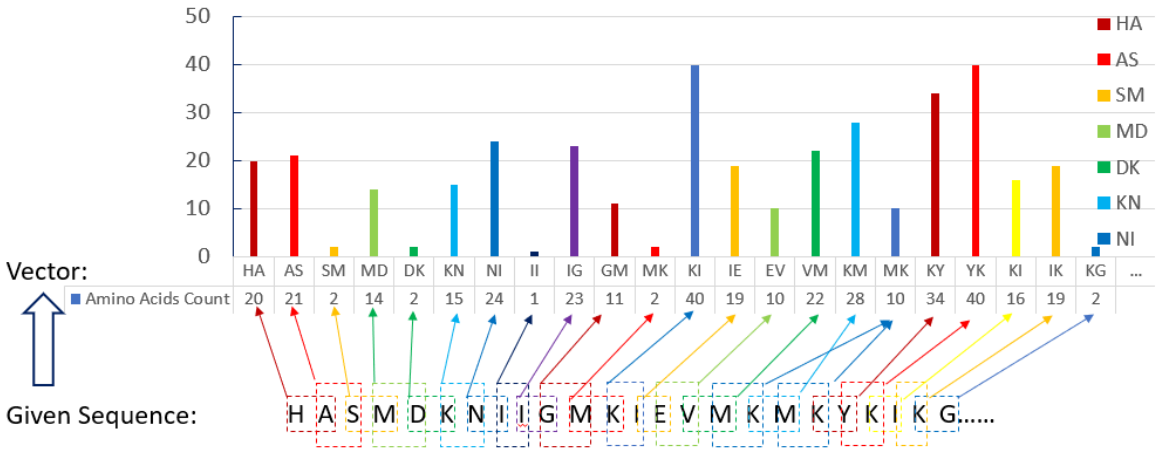


Fig.1 Illustration of the vectorization process, when we set word size to be 2 in sequence.

Main Algorithm

Now, this protein class is a multi-class classification problem. Here, we introduce Naive Bayes based classification algorithm because Naive Bayes has following properties:

1. Very simple, easy to implement, and the training time is fast;
2. Can be used for both binary and multi-class classification problems;
3. Highly scalable. It scales linearly with the number of predictors and the data points;
4. Not sensitive to irrelevant features;
5. Handles continuous and discrete data;

The algorithm mainly has three parts:

The Preparation Stage.

This phase is to make the necessary preparations for Naive Bayes classification. The main task is to determine the characteristic attributes according to the specific situation, here is to determine how many continuous amino acids can be formed an independent characteristic based on vectorization we mentioned above. Let's say each input have m characteristics, such that input is X = {a_1, a_2, a_3...a_m}. Because for this problem, we only focus on first 10 main kinds of class, so let the class label set to be Y = {y_1, y_2, y_3...y_{10}}. This stage is the vital stage in the naive Bayesian classification that needs to be completed carefully. The quality of the classifier is largely determined by the feature attributes, feature attribute division and training sample quality.

The Classifier Training Stage

The task of this stage is to generate a classifier. The main task is to calculate the frequency of occurrence of each category in the training set, the frequency of the characteristic *i* is P(yi). And then calculate the conditional probability estimates for each feature attribute partition. The conditional probability estimation of characteristic attribute x under category i is P(a_x|y_i). Based on Bayes rule and conditional probability of individual feature attributes under each category, we can calculate the conditional probability estimation of each category i under input x: P(y_i|X). The Bayes rule is:

$$P(y_i|x) = \frac{P(x|y_i)P(y_i)}{P(x)}$$

$$P(x|y_i)P(y_i) = P(a_1|y_i)P(a_2|y_i)...P(a_m|y_i)P(y_i) = P(y_i)\prod_{j=1}^{m} P(a_j|y_i)$$

In the end, compare each condition probability to get the max one, and the estimate class is the category of the max conditional probability.

$$P(y_k|x) = max\{P(y_1|x), P(y_2|x), ..., P(y_n|x)\}, x \in y_k$$

Application Stage

The task of this stage is to classify the classified items using the classifier. The inputs are the classifier and the item to be classified, and the output is the mapping relationship between the item to be classified and the category.

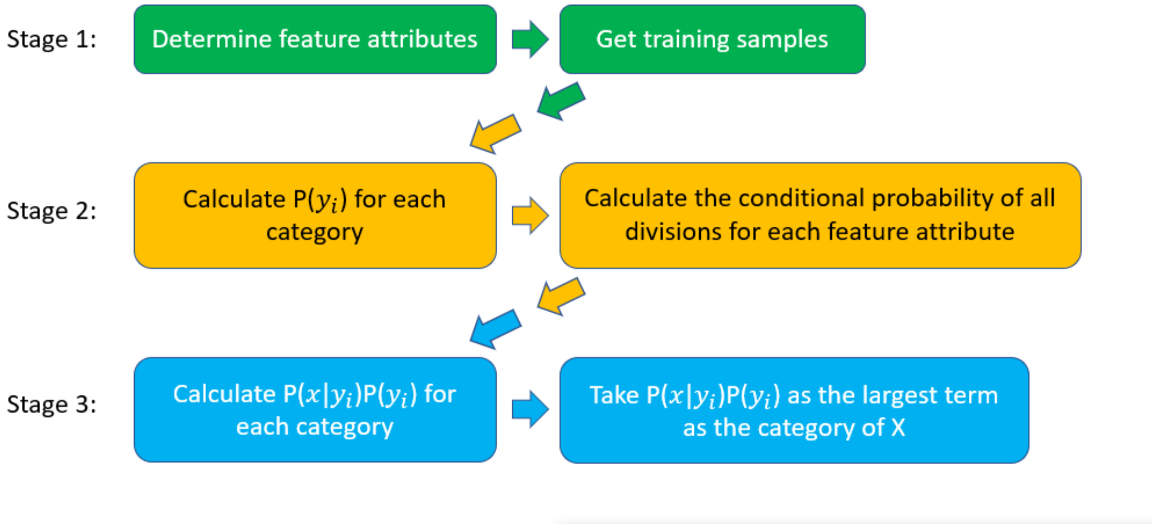
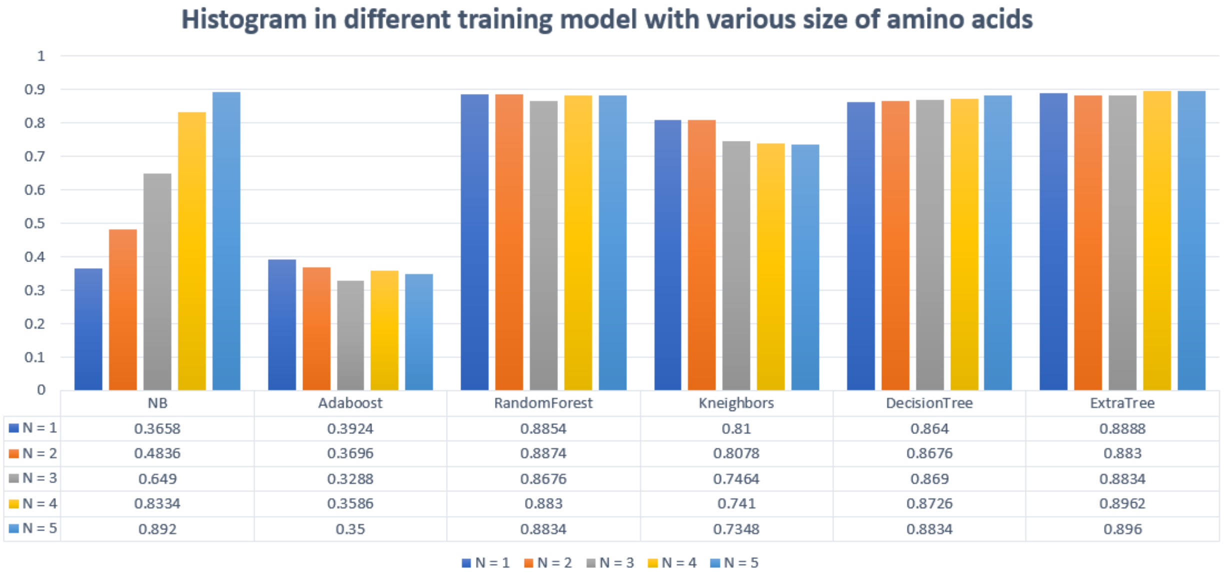

Stage 1:
Determine feature attributes
Get training samples
Stage 2:
Calculate $P(y_i)$ for each category
Calculate the conditional probability of all divisions for each feature attribute
Stage 3:
Calculate $P(x|y_i)P(y_i)$ for each category
Take $P(x|y_i)P(y_i)$ as the largest term as the category of X

Fig.2 Core pipeline of Naïve Bayes model

Results



	NB	Adaboost	RandomForest	Kneighbors	DecisionTree	ExtraTree
■ N = 1	0.3658	0.3924	0.8854	0.81	0.864	0.8888
■ N = 2	0.4836	0.3696	0.8874	0.8078	0.8676	0.883
■ N = 3	0.649	0.3288	0.8676	0.7464	0.869	0.8834
■ N = 4	0.8334	0.3586	0.883	0.741	0.8726	0.8962
■ N = 5	0.892	0.35	0.8834	0.7348	0.8834	0.896

Fig.3 The histogram that shows the performance for each model under different word size.

Figure. 3 shows the performance of different training model with various size of short amino acids. On the one hand, we found that only the Naive Bayes approach is sensitive to the length of short sequence we used in feature extraction. This is because this method mainly based on the calculation of conditional probability, which take each feature independently. If the size of the short sequence is too small, it may have fewer features than expected, and the features may influence each other, which means the features depend on others. So, the max value may be different with the correct label, and the accuracy becomes lower. As the length becomes longer, the correlation of each feature becomes smaller and the results get better. However, the size of sequence length cannot become too long, which may make lots of redundant features and let the training time becomes much longer than normal.

On the other hand, other approaches don't depend on this size, which means the training process don't rely on the independence of each feature. In the future work, we can simply utilize the simple amino acid as a simple feature to train the model.

Figure. 4 shows the confusion matrix of Naive Bayes model when setting short sequence range size to be 5. We can observe that the model performs generally the same among the ten categories, there is no significant bias exists as a result of data balancing.

Table.1 shows the results of our best Naïve Bayes model, we can tell not from the precision rates but also from the recalls that the model behaves generally uniform among the ten categories.

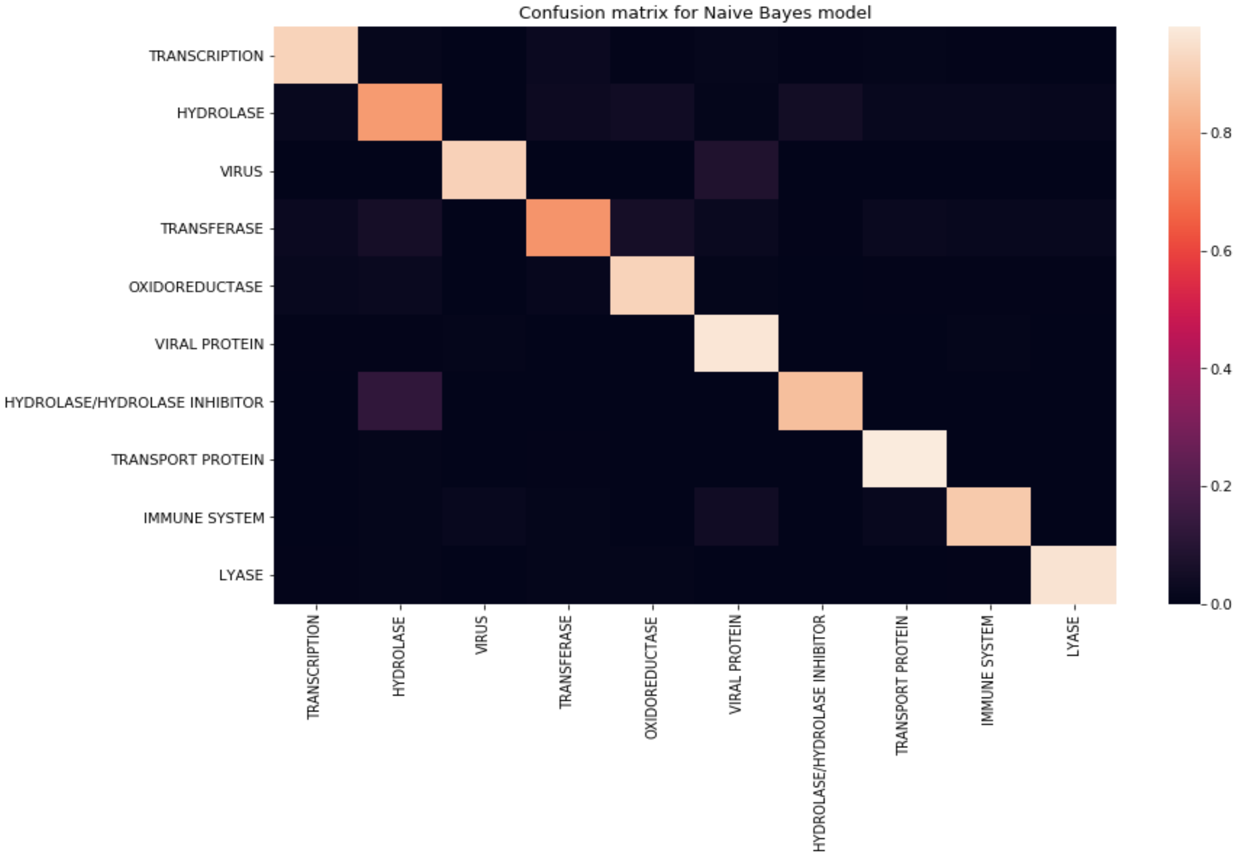


Fig.4 The confusion matrix for our best Naïve Bayes model.

	precision	recall	fl-score	support
HYDROLASE	0.74	0.78	0.76	479
TRANSFERASE	0.9	0.76	0.82	589
VIRAL PROTEIN	0.81	0.97	0.88	440
TRANSPORT PROTEIN	0.92	0.98	0.95	469
LYASE	0.95	0.96	0.95	467
TRANSCRIPTION	0.91	0.92	0.91	491
VIRUS	0.96	0.91	0.94	543
OXIDOREDUCTASE	0.89	0.92	0.9	514
HYDROLASE/HYDROLASE INHIBITOR	0.94	0.87	0.9	524
IMMUNE SYSTEM	0.92	0.89	0.91	484

Table.1 The performance for our best Naïve Bayes based model in each category.

Discussion & Future Work

The drawbacks of our models lie in multiple aspects. Firstly, we failed to address the impacts of amino acids separated by a large distance in the sequence, this may be a main defect to our model assuming that sequence information alone is sufficient to predict the protein function. One of the prospective methods to solve this issue is to use LSTM which could ‘remember’ the information long ago. Secondly, in fact, we cannot assume the sequence alone provides enough features for perfect classification. In real world, proteins generally have a 3-D structure and we cannot know its geometric information based on sequence alone. However, this is an inevitable problem and as a result of our limited biological knowledge, we simply assume that sequence alone is enough. What’s more, one may also wonder why we did not use the other features provided in the dataset as auxiliary information or build ensemble models together with them.

Other than analyzing the relationship between sequences and protein type, we also considered other features in the dataset, such as residueCount, resolution, etc.. In order to get a better understanding of the correlations between the physical attributes, we build a Pearson Correlation Matrix. An absolute value closer to 1 represents a stronger correlation between the two factors, further represents a weaker correlation. From the plot below we can tell that these attributes all have fairly small correlation with the classification.

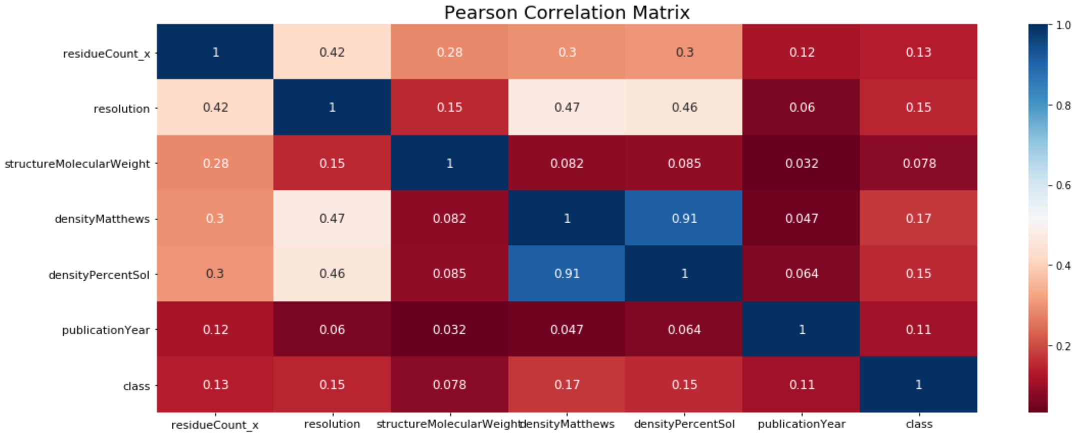


Fig.5 Pearson Correlation Matrix for other features in the dataset

Reference

1. Rose, Peter W., et al. "The RCSB Protein Data Bank: new resources for research and education." *Nucleic acids research* 41.D1 (2012): D475-D482.
2. Freund, Yoav, Robert Schapire, and Naoki Abe. "A short introduction to boosting." *Journal-Japanese Society For Artificial Intelligence* 14.771-780 (1999): 1612.
3. Rigort, A., & Plitzko, J. M. (2015). Cryo-focused-ion-beam applications in structural biology. *Archives of biochemistry and biophysics*, *581*, 122-130.
4. Lesh, Neal, Mohammed J. Zaki, and Mitsunori Ogihara. "Mining features for sequence classification." *KDD*. Vol. 99. 1999.
5. Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." *Emerging artificial intelligence applications in computer engineering* 160 (2007): 3-24.
6. Chawla, Nitesh V., Nathalie Japkowicz, and Aleksander Kotcz. "Special issue on learning from imbalanced data sets." *ACM Sigkdd Explorations Newsletter* 6.1 (2004): 1-6.