DNA Sequence Analysis

using Multi-class classification

Author: Yingying Wang

Stat 391-Final Project

1. Abstract

In this paper, we perform data analysis of a real dataset of DNA sequence. And the goal is to estimate the middle of the DNA sequence is a intron/exon boundary, exon/intron boundary or not a splice site. We use two classification methods including Naive Bayes classifier, and the Nearest-Neighbor(NN) classifier. For Naive Bayes method, we also use Laplace smooth to optimize the estimation result, but there is no improvement on the result, then we found Naive Bayes is not a good classifier for this dataset. For the Nearest-Neighbor method, the estimation result is slightly better than the result from Naive Bayes method. Although the estimation results from Nearest-Neighbor(NN) classifier is not good enough, but it's a better classifier compared to Naive Bayes.

2.1 Introduction

DNA(Deoxyribonucleic acid) can encode the genetic instructions used in the development and functioning of all known living organisms and many viruses. Most DNA molecules consist of two biopolymer strands coiled around each other to form a double helix. Each nucleotide is composed of a nitrogen-containing nucleobase including guanine(G), adenine (A), thymine(T), 0r cytosine(C)
Splice junctions are points on a DNA sequence at which 'superfluous' DNA is removed during the process of protein creation in higher organism. For this dataset is to recognize the boundaries between eons and introns, and then giving the correct label to the DNA sequence.

2.2 DataSet

The dataset contains two files including X-train file and Y-train file. For the X-train file, there is a large set of DNA sequences. Each DNA sequence is placed in one line. Each DNA sequence with length of 60 contains a specific combination of four kind of nucleotide, which represented as ACTG. The dataset also contains a small number of characters DNSR. In particular, D represents A or G or T, N represents A or G or C or T, S represents C or G, R represents A or G. For the Y-train file, there are same number of lines as in X-train file and each line has only one single number 0, 1 or 2. In particular, 0 represents not a splice site, 1 represents intron/exon site, 2 represents exon/intron site. So the single number in each line represents the label of DNA sequence for the corresponding line of DNA sequence in X-train file.

3. Preprocessing:

Before training the classifier, I read the data in two different ways. The first way is to read all the provided data and then train the classifier based on all data. The second way is to remove the data points containing D,N,R,S value in X-train file and the corresponding label value in Y-train file, and then train the classifier based on the selected data.

4.1.1 Naive Bayes classification

The given dataset D = {(x(1), c(1)),…(x(n), c(n))}, x(1)…x(n) represents a DNA sequence with length of 60, and c(1)…c(n) represents the label for the corresponding DNA sequence. For this dataset, the labels c are 0, 1 or 2.

The first step of Naive Bayes classification is to count the number of appearance of each label and get the probability of each label appears.

$$P_C(c) = \frac{\#c^{(i)} = c}{n} \equiv \frac{n_c}{n}, \text{ for } c \in \{1, \ldots M\}$$

All data	
#appearance of 0 & Probability of 0	1546 & 0.51533
#appearance of 1 & Probability of 1	721 & 0.240333
#appearance of 2 & Probability of 2	733 & 0.244333

Selected Data(without DNRS)	
#appearance of 0 & Probability of 0	1519 & 0.51619
#appearance of 1 & Probability of 1	704 & 0.239211
#appearance of 2 & Probability of 2	720 & 0.24464

Then compute P_x0|c…P_x59|c, that is for all DNA sequences labeled in same number, count the number of ACTG respectively in same position. ex. P_x1|c=0 is for all DNA sequences labeled in 0, count the number of ACTG respectively in position 1, and compute the probability of each count.

$$P_{X_j|C}(x_j = l|c) = \frac{\#x_i^{(i)} = l,\ c^{(i)} = c}{n} \equiv \frac{n_{jc}(l)}{n},\ j = 1 : m,\ l \in S_{X_j}$$

j is positions in range[0, 60), and x_j = l, where l is in {A, T, C, G}

c is the label in range{0, 1, 2}

I store it into a matrix like:

Px0IC	A	C	T	G
C = 0	0.25674786	0.23897301	0.24687294	0.25740619
C = 1	0.25	0.234375	0.27982955	0.23579545
C = 2	0.25416667	0.25	0.2625	0.23333333

So I have 60 matrix in a list showed above

By this model, we can estimate the label of a given DNA sequence using

C = maxarg(Pc(c) ∏(i=0~59)P(xj|c(xj|c))

Result:

There are two results: one for all data, another for selected data:

All Data				
Class	reall#	estimate#	#estimate success	percentage
0	84	127	73	92.40
1	37	8	3	3.80
2	29	15	3	3.80
all	150	150	79	52.67

Selected Data				
Class	reall#	estimate#	#estimate success	percentage
0	84	123	71	91.03
1	37	12	4	5.13
2	29	15	3	3.85
all	150	150	78	52

After the experiment, the result shows that estimation accuracy of using classifier trained by all data is slightly higher than the estimation accuracy of using classifier trained by selected data. However, the result is still not good enough. So I tried to optimize the result using Laplace smoothing.

4.1.2 Naive Bayes classification with Lap'Lace smoothing

Since P_xj|c could be zero, that is the count of label c in position j could be zero. Under this situation, it will make all probability to zero. In order to avoid 0 in probability, using Lap'Lace smoothing, we add 1 for each count so that the probabilities will not become zero.

Results:

All Data				
Class	reall#	estimate#	#estimate success	percentage
0	84	109	63	86.30
1	37	18	6	8.22
2	29	23	4	5.48
all	150	150	73	48.67

Selected Data				
Class	reall#	estimate#	#estimate success	percentage
0	84	107	63	86.30
1	37	20	6	8.22
2	29	23	4	5.48
all	150	150	73	48.67

After experiment, we could see the result is even worse than before. Accuracy is only 48.67%. So Lap'Lace is not a good optimization method because it make the accuracy decrease.

4.1.2 Special component replacement(D,N,R,S)

In above experiments, we use all data and selected data to train the classifier. The result shows that there is no big difference between the result from classifier trained by all data and selected data. When we use all data to train the classifier, since D represents A or G or T, N represents A or G or C or T, S represents C or G, R represents A or G and we don't know the exact probability of each component occurs, so I make them to appear with same probability. However when I do the experiment of changing the appearance probability of each component, the final result is slightly different, but they are almost same.

#appearance of each component				
	D	N	R	S
Count	2	54	0	1

We could see that the number of appearance of N is much larger than other these components. So we decide to do experiment on changing the occurrence probability of A, G, C, T that used to replace N.

	A	C	G	T
D	0.3		0.4	0.3
N	0.25	0.25	0.25	0.25
S		0.3	0.3	0.4
R	0.5		0.5	
result				52.67%

	A	C	G	T
D	0.3		0.4	0.3
N	0.9997	0.0001	0.0001	0.0001
S		0.3	0.3	0.4
R	0.5		0.5	
result				52%

	A	C	G	T
D	0.3		0.4	0.3
N	0.0001	0.9997	0.0001	0.0001
S		0.3	0.3	0.4
R	0.5		0.5	
result				52.67%

	A	C	G	T
D	0.3		0.4	0.3
N	0.0001	0.0001	0.9997	0.0001
S		0.3	0.3	0.4
R	0.5		0.5	
result				52.67%

	A	C	G	T
D	0.3		0.4	0.3
N	0.0001	0.0001	0.0001	0.9997
S		0.3	0.3	0.4
R	0.5		0.5	
result				51.33%

We can conclude from the experiment statistics, the accuracy is slightly different but they are almost same. When we increase a lot of the probability of using T to replace N, the accuracy is the lowest which is 51.33%, and when we increase a lot of probability of using A to replace N, the accuracy is the second lowest which is 52%, and when we increase the probability of using C or G to replace N, the result is 52.67%. Those results did't differ to much, so the probability of special component replacement didn't influence the accuracy of classifier to much.

4.2 The K-th Nearest-Neighbor(NN) classification

The classification rule is to find k points in the data set that are closest to x. Choose the class of x to be the class of the majority of the k neighbors. For DNA dataset, we use a DNA sequence that we are estimating to compare with all DNA sequence in dataset and find K-th closest value, and then choose the class of estimating DNA sequence to be the class of the majority of k neighbors. And I did experiment on different k value and to find the most appropriate k value.

k	19	20	21	22	23	24	25	26	80	100
result	54%	54.67%	55.33%	54.67%	52.67%	54%	52.67%	55.33%	56%	56%

As we can see from the result, the k value do influence the accuracy of classifier but it doesn't influence too much. I use k value from 19 to 26, 80 and 100. The highest accuracy occurs when k = 80 and 100 which the accuracy become 56%. But when we use k value as 80 or 100, the estimation labels for all DNA sequence in test file will all be 0. So the accuracy become 56%. From the theory of K-th Nearest-Neighbor(NN) classification, we know we need to choose a some k instead of 80 or 100 such large numbers. So when k is from 19 to 26, the highest accuracy is 55.33% when k is equal to 21 and 26. So I choose 21 as k value in KNN classifier.

Result:

All Data				
Class	reall#	estimate#	#estimate success	percentage
0	84	140	79	95.18
1	37	6	3	3.61
2	29	4	1	1.20
all	150	150	83	55.33

Compare result from KNN classifier with result from Naive Bayes classifier, the accuracy of KNN classifier is higher than the accuracy of Naive Bayes classifier, even with optimization. But both of them are still within the range of [50%, 60%], which is still not good enough. Then I tried to combine 5 estimations together to get the better result.

4.3 Combine KNN result and NB result

There are four model trained by NB method. 1) All data & Lap'Lace; 2) All data & without Lap'Lace; 3) Selected data & Lap'Lace; 4) Selected data & without Lap'Lace and one mode fron KNN method. Totally we get 5 models. And I put the class estimations of DNA sequence from these five models together in one data file. Then I choose the class of a DNA sequence to be the class of the majority of estimations from 5 models. For example, four models estimate the first DNA sequence to be class 0 and

one model estimate the first DNA sequence to be class 1. We combine result together and class first DNA sequence to be 0.

Result:

All Data				
Class	reall#	estimate#	#estimate success	percentage
0	84	125	64	92.75
1	37	16	2	2.90
2	29	22	3	4.35
all	150	150	69	46

Surprisingly, the accuracy of this method is even much lower than accuracy of estimation from 5 models above. Therefore, this is not a good optimization method to increase the accuracy of classifier.

5. Conclusion

After all experiments, it is clear that the accuracy of each model does have difference, but they didn't differ to much and there is even no improvement after using some optimization methods on it. First we use Lap'Lace method but the result is even worse after this optimization. So Lap'Lace is not a appropriate optimization method for this dataset. For the model trained by KNN method, the accuracy is better than other method but is still under 60%. Additionally, the k-value that used in KNN classifier is also a aspect that can influence the accuracy. Finally, I combine the estimations from five models together to get a new estimation. That is, choose the class of a DNA sequence to be the class of the majority of estimations from 5 models. But the result is even worse, so this is not a good optimization method to increase accuracy. Compare these five models together, the accuracy of KNN method is highest, but it's still good enough to estimate the class of DNA sequence.