DNA Splice Site Detection Using Naive Bayes and K-NN

Zhongyue Zhang
June 10, 2014
Statistics 391 Final Project
zhangz6@cs.washington.edu

Abstract

Naive Bayes and K-Nearest Neighbors (K-NN) are two types of classifier. Here we trained both classifier with different feature vector to identify whether a 60 bases long DNA sequence is an intron/exon or exon/intron splicing site. Our experiments show that K-NN performs better than Naive Bayes. In particular, K-NN achieve better result than the baseline, which is to classify everything not being a splicing site, while Naive Bayes does worse than that. However, neither classifier is suitable for this task, since both of them have a low accuracy.

1. Introduction

Splice site are points on a DNA sequence where part of the DNA, called exon, is removed from the sequence during the process of protein creation. The part that is kept is called intron. The training dataset contains 3000 DNA sequences that are 60 bases long. Each base can be either be A, G, T, C, D, N, S or R. D, N, S and R represent ambiguous bases (see fig. 1). The class labels have three category, intron/exon splicing site represents by 1, exon/intron splicing site represents by 2, and 0 means it is not a splice site. The test set contains 150 examples with no ambiguous bases. We trained and validated our classifier using the training set and then test on the test set. We also use the validation to tune our hyper parameters, if there are for the specific classifier. The experiments results indicated that neither K-NN nor Naive Bayes do well on this task.

base	meaning
D	A or G or T
N	A or G or C or T
S	C or G
R	A or G

Figure 1

2. Naive Bayes

For Naive Bayes classifier, we used the frequency of AGTC at each 60 position as feature vector. We experimented both with and without removing ambiguous DNA sequences, as well as Laplace smoothing. When ambiguous DNA are removed, the Naive Bayes simply take the frequency vector as features. When ambiguous DNA are not removed, each base that can be represented by the ambiguous base are added by 1/#of possible base represented by the ambiguous base. For example, if we detect a D, 1/3 are added to A, G and T.

3. K-NN

For K-NN classifier, we tried three different feature vector: base frequency at each position (BFP), codon (groups of three bases, since one codon can be translated to an amino acid) counts (CC) and amino acid counts (AAC). There are also two ways of counting codon and amino acid frequency. One being splitting the 60 base long DNA in to 20 groups, another being counting each 58 position with consecutive three bases. The rationale of implementing the second method is that the DNA may not align to start from exactly the first base. We also experimented with several different distance measurement. They are gram edit distance for BFP, Manhattan and Euclidean distance and cosine similarity for CC and AAC. For cosine similarity, we choose the largest k instances instead of the smallest in contrast to three other distance measurement.

4. Experiments

For training and validation, we used holdout method, where we randomly generated a holdout set from the training set with a ratio of 4 to 1, 4 being the training set and 1 being the validation set. This result in a training set of size 2400 and validation set of size 600. During training and validation we also tune our K-NN to maximize accuracy with optimal k. After validation, we train on both training and validation set, and then test on the test set.

Noted that we will use the frequency of not splice as the baseline accuracy, which for validation set is 51.0204%, and test set is 60%

4.1 Naive Bayes

Table 1 Validation result: Naive Bayes without Ambiguous DNA without smoothing

			Predicted Class			
			Not Splice Site	*IE Site*	*EI Site*	*Recall*
Actual	*Class*	*Not Splice Site*	*255*	*22*	*23*	*85.000%*
		IE Site	*122*	*14*	*16*	*9.211%*
		EI Site	*122*	*5*	*9*	*6.618%*
	Precision		*51.102%*	*34.146%*	*18.750%*	*47.279%*

Table 2 Validation result: Naive Bayes without Ambiguous DNA with smoothing

			Predicted Class			
			Not Splice Site	*IE Site*	*EI Site*	*Recall*
Actual	*Class*	*Not Splice Site*	*256*	*22*	*22*	*85.333%*
		IE Site	*122*	*14*	*16*	*9.211%*
		EI Site	*122*	*5*	*9*	*6.618%*
	Precision		*51.200%*	*34.146%*	*19.149%*	*47.449%*

Laplace smoothing made the classifier predict one less EI Site and one more Not Splice, which shows no significant improvement. The main reason here is that the dataset is quite large, and there is no zero probability.

Table 3 Validation result: Naive Bayes with Ambiguous DNA without smoothing

			Predicted Class			
			Not Splice Site	*IE Site*	*EI Site*	*Recall*
Actual	*Class*	*Not Splice Site*	*253*	*22*	*25*	*84.333%*
		IE Site	*121*	*12*	*19*	*7.895%*
		EI Site	*120*	*6*	*10*	*7.353%*
	Precision		*51.215%*	*30.000%*	*18.519%*	*46.769%*

Taking account of ambiguous DNA balances the recall of IE Site and EI Site, otherwise, everything is worse. However, the difference is not significant, which may be accounted by the variance of the validation set.

Since ignoring ambiguous DNA with smoothing has the best result in validation, I decided to run it on test set.

Table 4 Test result: Naive Bayes with Ambiguous DNA with smoothing

			Predicted Class			
			Not Splice Site	*IE Site*	*EI Site*	*Recall*
Actual	*Class*	*Not Splice Site*	*78*	*4*	*8*	*86.667%*
		IE Site	*28*	*3*	*3*	*8.824%*
		EI Site	*21*	*3*	*3*	*1.111%*
	Precision		*61.417%*	*30.000%*	*21.429%*	*55.629%*

The result is worse than baseline's 60%. Above experiments show Naive Bayes cannot perform better than baseline.

4.2 K-NN

For K-NN we will show the tuning process and the optimal confusion matrix

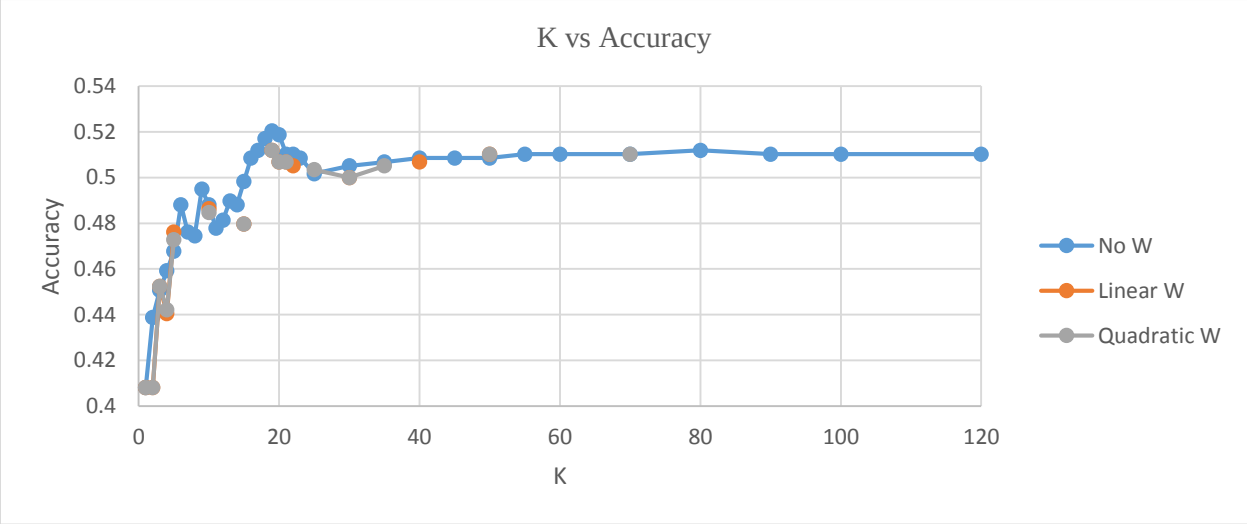

Figure 2

Accuracy peeks at K = 19 with no weight

Table 5 Validation result: K-NN BFP feature with no weight

			Predicted Class			
			Not Splice Site	*IE Site*	*EI Site*	*Recall*
Actual	*Class*	*Not Splice Site*	*293*	*3*	*4*	*97.667%*
		IE Site	*139*	*9*	*4*	*5.921%*
		EI Site	*131*	*1*	*4*	*2.941%*
	Precision		*61.417%*	*30.000%*	*21.429%*	*52.041%*

Table 6 Test result: K-NN BFP feature with no weight

			Predicted Class			
			Not Splice Site	*IE Site*	*EI Site*	*Recall*
Actual	*Class*	*Not Splice Site*	*90*	*0*	*0*	*100.000%*
		IE Site	*32*	*2*	*0*	*5.882%*
		EI Site	*23*	*3*	*0*	*0.000%*
	Precision		*62.069%*	*40.000%*	*0.000%*	*61.333%*

K-NN BFP is able to outperform baseline and Naive Bayes consistently in terms of accuracy.

Using the same technique, I found that K-NN CC cannot outperform baseline, no matter which configuration I use. Here is a local optimal K for K-NN CC as an illustration.

Table 7 Test result: K-NN CC feature with Manhattan distance and no weight

			Predicted Class			
			Not Splice Site	*IE Site*	*EI Site*	*Recall*
Actual	*Class*	*Not Splice Site*	*87*	*2*	*1*	*96.667%*
		IE Site	*33*	*1*	*0*	*2.941%*
		EI Site	*26*	*0*	*0*	*0.000%*
	Precision		*59.589%*	*33.333%*	*0.000%*	*58.667%*

Next is K-NN AAC. It is not able to achieve higher accuracy than baseline on validation set. However, it surprisingly reached an accuracy of 61.333% on test set with K = 21 or 22.

Table 7 Test result: K-NN AAC feature with Manhattan distance and no weight

			Predicted Class			
			Not Splice Site	*IE Site*	*EI Site*	*Recall*
Actual	*Class*	*Not Splice Site*	*90*	*0*	*0*	*100.000%*
		IE Site	*32*	*2*	*0*	*5.882%*
		EI Site	*26*	*0*	*0*	*0.000%*
	Precision		*60.811%*	*100.000%*	*0.000%*	*61.333%*

Other configuration like cosine similarity, Euclidean distance and assuming that the codons are align to the first base in the DNA sequence all have worse performance than K-NN BFP, with accuracy below baseline.

5. Conclusion

Experiments results show Naive Bayes and K-NN do not offer satisfactory result on DNA splicing site detection. Among these classifiers, K-NN BFP has the highest accuracy. It perform better than the baseline on both validation and test. The primary cause of Naive Bayes' result might be that the bases is highly dependent, as these bases form amino acid and then further form protein. As for K-NN, since it does not rely on the assumption of each feature being dependent, it works better than Naive Bayes. However, the features I choose still cannot make the classifier to differentiate different class well. One of the surprise result is that K-NN AAC is worse than K-NN BFP. One possible reason might be that the exon in DNA do not form amino acid and protein, which means grouping exon's bases together is a incorrect feature for the classifier and it does not provide any advantages. Another possible feature that we did not try that may work better is entropy of the DNA sequence. If exon and intron have significant entropy difference, it may works better than the methods tested here. Finally, both K-NN and Naive Bayes are not very good at capturing the hierarchical structure of DNA, since DNA is translated in multiple steps to protein. In this regard, other classifiers like deep neural networks may work significantly better.