Kernel Methods challenge

Group name: EpsilonGreedy

Joram Bakekolo
Lloyd Acquaye Thompson

May 30, 2020

Introduction

Classification algorithms have been developed in symbolic learning, statistics, and neural networks, usually using different data modelling and representation techniques. Classification is here defined to be the problem of correctly predicting the probability that an example has a predefined class from a set of attributes describing the example. There exist many different algorithms, but their relative merits and practical usefulness are unclear. Thus, the need arises to evaluate their relative performances, in particular on large-scale industrial problems.

0.1 Problem Description

DNA(Deoxyribonucleic Acid) sequence is the process of determining the nucleic acid sequence, the order of nucleotides in DNA. It includes any method or technology that is used to determine the order of the four bases: adenine, guanine, cytosine, and thymine. This information is useful for understanding the type of genetic information that is carried in the DNA, which may affect its function in the body.
In this report we used machine learning algorithms to our structural data of DNA. Indeed, we used machine learning algorithm on dataset constituted of sequence od DNA to classify whether a DNA sequence region is binding site to a specific transcription factor. Transcription factors (TFs) are regulatory proteins that bind specific sequence motifs in the genome to activate or repress transcription of target genes. Genome-wide protein-DNA binding maps can be profiled using some experimental techniques and thus all genomics can be classified into two classes for a TF of interest: bound or unbound.
In our study, we used three different TFs.

0.2 Data Description

This data challenge contains one dataset of 2000 training sequences. The main files available are the following ones.

* Xtr.csv - the training sequences.
* Xte.csv - the test sequences.
* Ytr.csv - the sequence labels of the training sequences indicating bound (1) or not (0).

Each row of Xtr.csv represents a sequence. Xte.csv contains 1000 test sequences, Ytr.csv contains the labels corresponding to the training data.

The visualization of data leaded us to the following figure decribing the distribution over classes.

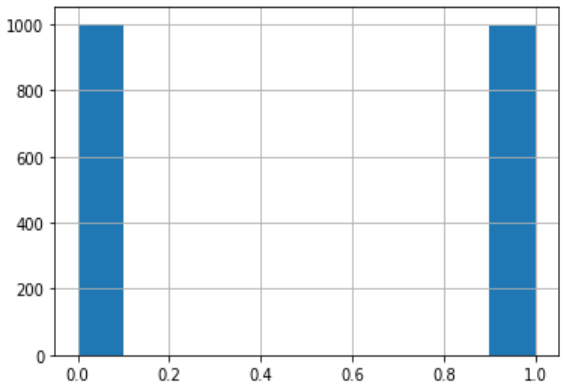

Figure 1: Data distribution

0.3 Computation

Differents algorithms have been used to solve this classification problem with differents significant results. In this part we will try to describe and explaining technique and preprocessing used to get our results.

0.3.1 Preprocessing using counting technique

1. **Features generation**
   We generated features by counting the number of subsequences occurated and defined the size of the subsequence from our DNA sequence and used the StandardScaler to scale the data.

2. **Data dimension reduction**
   We performed Principal component analysis (PCA) for reducing the dimensionality, increasing interpretability but at the same time minimizing information loss in the datasets.

3. **Trained model**
   We coded from scratch **Logistic regression** algorithm that we used to do the classification task.

4. **Training loop**
   We used gradient descent to minimize the loss. We trained the model with learning rate of 0.01, for 100 iterations which end with 0.73 accuracy in the private leadbord. then, Our prediction on kaggle gave an accuracy of 0.6160. We tryied to increase the accuracy by changing different values of the learning rate.

We decided to explore different models and differents technique of preprocessing our dataset in order to do a good classification.

0.3.2 Preprocessing using one hot encoding

1. **Features generation**
   We generated our feature in the same way(previously), used one hot encoding, MinMaxscalar. We splitted our dataset in two subsets, 90% for the training and 10% for validation.

2. **Kernels definition**
   We defined 3 types of kernel to be used for our classification problem, linear kernel, quadratic kernel and rbf kernel.

3. **Training Models**
   The trained with **kernel logistic regression** and **kernel support vector machine** by using with kernel defined above (linear, quadratic and rbf).

4. **Training kernel logistic regression**
   In the training , we used different kernels with different parameters. The model was trained with a learning rate of 0.01, with 100 iteration and tolerance of $1e-5$, we obtained we obtained 0.0.54,0.47 accuracy in the private board 0.59200, 0.51600.

5. **Training kernel support vector machine**
   In the training, we used different kernels with different parameters and svm dual soft for optimization in order to have good accuracy. With a linear kernel 0.1 for sigma and C, tolerence of 1e-5, we obtained 0.62 accuracy. With a rbf kernel 0.1 for sigma and tolerence of 1e-5, we obtained 0.50 accuracy.

   With sigma of 0.1 and tolerence of 1e-5, we obtained 0.645 accuracy in the private leading board and 0.686 accuracy in kaggle.

0.4 Conclusion

In conclusion, through this report we described how we did the preprocessing, built from scratch our classifier algorithms for the DNA sequence challenge using simple logistic regression, kernel logistic regression and kernel support vector machine with different kernels linear, quadratic and rbf. The results obtaimed in the private boad after training those models are follow: 0.73, 0.70 for the simple logistic regression, 0.0.54,0.47 for kernel logistic regression, 0.5600, 0.5640, 0.50 with rbf kernel. We obtained 0.63, 0.645 for kernel suport vector machine with quadratic kernel.

While the results obtained in the kaggle were as follow: 0.61600, 0.58199 for the simple logistic regression. 0.59200, 0.51600 for the kernel logistic regression, 0.59600, 0.60600 for the kernel support vector machine with rbf kernel and 0.66600, **0.68600** for the kernel support vector machine with quadratic kernel. Based on our results in kaggle, the kernel support vector machine performed better when using a quadartic kernel.

However one of the instruction of this challenge was not to use pre trained models to do the classification task. In trainedfuture it will important to try to do the classification with pre trained models and see how the performance is.