Kaggle report for data challenge

Kernel Method

Amath Sow
Djibril Dassebe

1 Introduction

The goal of the data challenge is to learn how to implement machine learning algorithms, gain understanding about them and adapt them to structural data. For this reason, we have chosen a sequence classification task: predicting 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**.

We have used several steps to process and organize our data as described in the following lines.

2 Data processing

Working with DNA sequence data is commonplace. The file can contain one or many DNA sequences. Here we'll use one-hot encode the sequence letters and use the resulting array. Let's first create some utility functions such as for creating a numpy array object from a sequence string and for that, we have implemented the function *stringtoarray* and the and the function to encode a DNA sequence string as an ordinal vector. And we later define the function *onehotencoder* which as input the array. One challenge that remains is that none of these above methods results in vectors of uniform length, and that is a requirement for feeding data to a classification. We finally implement the function that can be used to convert any sequence (string) to overlapping k-mer words and now ready to be used with our models.

We have used several models which we will describe in detail in the following lines

3 Models and Results

the different methods used in this challenge are:

3.1 Gaussian kernel with numerical data

The Gaussian kernel or radial basis function kernel, or RBF kernel, is a popular kernel function used in various kernelized learning algorithms. In particular, it is commonly used in support vector machine classification. Here we are given numerical data for $Xtrain$ and $Xtest$. We have the implemented the SVM with Gaussian Kernel as well as the optimization problem related. When evaluating the performance of this model, we obtained an accuracy of 60.000%

3.2 Gaussian kernel using word embedding

Word Embedding is a language modeling technique used for mapping words to vectors of real numbers. It represents words or phrases in vector space with several dimensions which was very helpful for this DNA sequence data. But the performance was lower with an accuracy 59%

3.3 Kernel Logistic Regression

We have implemented a Logistic Regression without the help of built-in libraries (except numpy) where the implementation class is descried in our notebook as well as all the steps. And again, we got another lower accuracy $\sim 53.8\%$

3.4 Kernel Ridge Regression

This model is the one that has given us a better accuracy so far. The implementation is well detailed in the notebook as well as the kernel functions used with the help of representer theorem seen in class. We have used cross validation to identify how well our model performed and to test the accuracy of our model to verify that, it's well trained with data without any overfitting and underfitting. We displayed our prediction after fitting the model in order to compute the accuracy and it turns out that, the accuracy for this model is 68.200%

4 Discussion

In this competition, we used several models of DNA sequence data of 2000 training sequences but one of the most important question is *What are the performance outcomes of a machine learning process?* To what extent is the algorithm able to extrapolate from the training sample to a large citation dataset? But we know in Machine learning, the high the accuracy the better the model is. with our amount of data, we were able to try a variety of models using *kernel Methods*. Since we are using almost all our models just after the data processing, the big challenge will be to choose the Best model among the models we have used which turned to be kernel Ridge regression with 0.68200 accuracy.

Conclusions

Our aim was to learn how to implement machine learning algorithms, gain understanding about them and adapt them to structural data. After our study, we realize that there exist many algorithms that can do that and our goal was to used kernel methods. We implemented about four of them and it turns out that hyperparameter optimization plays a great role because it highly depends on the performance of the model. This could be justify by the fact that Kernel Ridge Regression for instance, the accuracy was initially 0.6600% but after tunning well our hyperparameter, we managed to improve our accuracy up to 0.68200.

Kernel methods have not only enriched the machine learning research by offering the opportunity to dealing with different tasks and different input structures, but have also provided new perspectives for solving typical problems with a methodology supported by strong intuition and well founded mathematical theory. One of the important thing that we have learned is that one can straightforwardly perform nonlinear transformations of the original patterns into a (generally) higher dimensional feature space which is consequently nonlinearly related to the input space. Many algorithms in the scientific literature have been kernelized

References

[1] Andrew Ng, notes on SVM and kernel method, Stanford University

[2] Lectures notes on Kernel Methods in Machine Learning by Julien Mairal, Jean-Philippe Vert & Romain Ménégaux (T.A.)

[3] http://members.cbio.mines-paristech.fr/ jvert/svn/kernelcourse/course/2019ammi/index.html