Report of the data challenge

Prof Jean-Philippe Vert , Prof Julien Mairal, Romain Menegaux (TA)
AMMI 2020
Course: Kernel Methods
Group members: Armand Bandiang Massoua, Moteu Ngoli Tatiana

***Abstract*— For us to have hands-on experience in kernel methods and machine learning pipeline in general after the awesome lectures we had. Our lecturers decided to end the class with a data challenge. The challenge was on a binary classification task: predicting whether a DNA sequence region is binding site to a specific transcription factor.**

I. Introduction

The data challenge allowed us to implement machine learning algorithms, gain understanding about them and adapt them to structural data. The problem define here is a classification task.
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.
During the challenge, we used a dataset corresponding to three different TFs. The dataset is devide into 2000 training examples, 1000 test examples, for which we need to predict and the labels corresponding to the training data.

II. Tools

To achieve our goal in this challenge, we have used the following tools:

- 2 laptops with the following characteristics:
  - OS: Debian
  - RAM: 16GB
  - CPU: core i7
- Google Colab
- Jupyter Lab
- Python 3

III. Methodology

The main objective for this challenge was to learn how to implement things in practice, and gain practical experience with the machine learning techniques taught during the course. To address that, we did the following:

- Data prepossessing: Here we divide each of DNA sequence data into k-mers and for each k-mers we convert it into one-hot-encoding
- Data visualization: to see the distribution of the classes in the training set. We have 50.1% for class -1 and 49.9% for class 1. There's class balance.
- Models:
  - Logistic Ridge Regression
  - SVM soft margin
  - Kernel (SVM and Logistic Ridge Regression)

IV. Experiments and results

Since we've structure data constituted of strings. There are many techniques to deal with such a data. The approach we used is to take an example DNA sequence and dividing it into k-mers. Then we did a one-hot encoding of them to get numerical sparse vectors. For the training process we have divided the training data into training set and validation set.

The choice of models parameters and the hyperparameters are done through grid search couple with cross-validation. However, what we have observed during our cross validation was that it was actually difficult to choose a good C, lambda or sigma parameter, cause the outcome depend hugely on the split between training and validation data. So, After some experiments we have gotten the following results:

- SVM+soft margin with the rbf kernel, C= 66, sigma=5 gives us a public score of 67% and a private score of 65%
- Kernel Logistic ridge regression: with the Linear kernel, lambda = 0.159, gives us a public score of 69% and a private score of 63%
- Logistic ridge regression with lambda=0.001 gives us a public score of 63% and a private score of 62%
- SVM soft margin with C=138.48 gives us a public score of 62% and a private score of 61%

The kernel logistic ridge regression achieved the best performance on the public leaderboard with 69%, but poorly on the private leaderboard with 63%. We suspect this might be due to overfitting of the public data. It turns out that our best model the private leaderboard is the SVM with the rbf kernel, whose score was 65%.

V. CONCLUSIONS

During this challenge we have tried to implement and test various Kernels that could be applied to DNA sequences. We have also used a lot of regularization to avoid overfitting and achieve a good score. We were surprised and ended up being disappointed when the private leaderboard was disclosed, because we moved from the $7th$ place in the public leaderboard (with 69% accuracy) to the $34th$ place (63% accuracy). We dont think that is due to some kind of overfitting because from what we have seen our model was working fairly well during our validation. We also want to mention that our best model on the private learborad was left out for the final ranking. We can say we lack some experience in choosing two models for the final ranking. We dont think that this final outcome in the private leaderboard fairly represents the time that we spent on this project. However, the challenge was pretty exciting and interesting, it allowed us to learn a lot! And now we are equipped with tools, skills, and experience to tackle similar problems in the future.

REFERENCES

[1] https://github.com/rmenegaux/kernels-AMMI-2020

[2] https://stackabuse.com/one-hot-encoding-in-python-with-pandas-and-scikit-learn/

[3] https://www.kaggle.com/thomasnelson/working-with-dna-sequence-data-for-ml