Kernel Method in Machine Learning Data Challenge

ADEMOLA ABIODUN SAHEED, ABDULAI HAMID FADILA

July 5, 2022

1 Introduction

In this data challenge our goal is to implement machine learning algorithms to structural data, we have chosen a sequence classification task:short DNA fragments (100 to 300bp long), that come from sequencing experiments, or that were simulated from full genomes. The goal is to discriminate the Covid-19 fragments, hence the task is a binary classification task: the labels are either 1 if the fragment is identified as Covid-19, and 0 otherwise. The performance is evaluated by the classification accuracy on the test data.A leader board will be available during the challenge, which shows the best results per team, as measured on a subset of the test set. A different part of the test set will be used after the challenge to evaluate the results. The goal of this data challenge is to learn how to implement machine learning algorithms from scratch. For this reason, external machine learning libraries, as well as computer vision libraries are forbidden.

2 Data Preprocesing

The label data y were given as either 1 if the fragment is identified as Covid-19, and 0 otherwise. But with the kernel algorithm we used, it is pertinent to work with a binary class of label -1 and 1. So the y label was first converted to label -1 and 1 keeping the initial label 0 and 1 for Logistic regression.

2.1 Split Data

The Sklearn library was used for splitting of the data into train and evaluation set for easy cross validation and hyper-parameter optimization.

2.2 K-mer counting/Spectral Embedding

The DNA fragment sequence were break up into subsequences using k-mer size of 3. Then assign index to the subsequences. Convert the index to one

hot vector called spectral embeddings.
For large k-mer size k, the vector will be very large which can only be resolved using sparse vectors. But in our case, we limit our spectral embedding process to k =3.

3 Models

3.1 Logistic regression

Logistic regression was written from scratch and cross validation was used to evaluate the performance. Different Word embeddings were used to to evaluate the logistic model.
K-mer === 0.916
Count-Vectorizer === 0.925
Tf-idf === 0.549
The result above is a simple logistic regression without any hyper-parameters search, making a baseline model. Count-Vectorizer gives the best representation compare to the other two.

3.2 Kernel

Different kernels were implemented such as; guassian kernel, linear kernel, polynomial kernel and rbf kernel. All the kernels algorithms were developed in class. The parameter sigma was obtained using median heuristic approach.

3.3 Kernel Ridge regression and Support Vector Machine(SVM)

In the case of kernel Ridge regression and kernel SVM, we replace all data case with their features vectors.
i.e

$$X_i \implies \Phi_i = \Phi(X_i)$$

4 Results and Conclusion

SVM models and ridge regression with special kernels were implemented with cross validation. Optuna optimization framework was used for optimization of the parameters. Optimizing parameters for kernels is very difficult task. we took a lot of time doing parameter search. The following hyper-parameter were obtained as the best hyper-parameter after optimization. model_name='KernelSVM', kernel = 'rbf', k=4, sigma = 26.495, C = 4.724, lambda =0.01. The final accuracy obtained with the hyper-parameters obtained is 0.9615. This earn 0.992 on the public leaderboard on kaggle competition page.