ADVANCED LEARNING MODELS DATA CHALLENGE

Kritika Mehta
Grenoble INP
kritika.mehta@grenoble-inp.org

Lyubomyr Polyuha
Grenoble INP
lyubomyr.polyuha@grenoble-inp.org

Abstract

In this report,we present our work on the Advanced Learning Models Data Challenge. The goal of the challenge was a classification task - to predict whether a DNA sequence region is binding site to a specific transcription factor. No machine learning libraries were allowed and the main goal was to implement machine learning algorithms and gain understanding about them.

1 Introduction

The data challenge consisted of 3 datasets with 2000 observations each where each row represented a sequence and the labels were to be predicted separately for each of the dataset. 3 test datasets were provided with 1000 observations each, for which the predictions were to be performed and the results concatenated in a single file. For each dataset, 2 different versions were provided, the first one being the raw dataset and the second one was a numeric representation of the data based on bag of words representation. The numeric data was provided for those who prefer to work direcly on numeric data but the representation was not gauranteed to be optimal. Our approach was to work with second version which was fairly simple to work with and then move to the raw sequences which required some feature creation, to obtain better results.

2 Our Approaches

2.1 Bag-Of-Words Representation

Our first approach was to use algorithms from the scikit learn python library on the simpler dataset, to get an idea about which algorithm works better. We used cross-validation to perform such tests and found out that logisitic regression and Kernel SVM gave decent results. We trained the first model with a simple l_2 regularized logistic regression algorithm and submitted our predictions on kaggle which gave a score of 0.63. After tuning the hyperparameters the accuracy increased to 0.64 but different values of hyperparameters also gave worse accuracy of 0.54. The different values of hyperparameters tried is listed in Table 1. Different combinations of hyperparameters gave results ranging from 0.54 to 0.64.

Hyperparameters		
Learning Rate	Regularization Parameter	Iterations
0.01	0.1	500
0.1	0.001	800
0.5	0.5	1000
0.0001	1	1200

Table 1: Hyperparameters in Logistic Regression

2.2 Raw Dataset

As a part of the second approach, we worked with the raw dataset and created k-mers of length 5 giving us 1024 substrings (Number of substrings = 4^5). To find such substrings we used the library 'itertools' and its combinator generator 'combinations_with_replacement'. For each observation/ sequence, we counted the occurrence of each subsequence and stored in a feature array. Similarly, the features are formed for the test data. In the second approach, we implemented the kernel SVM algorithm with linear, polynomial and rbf kernels with different values of the penalty coefficient. For the kernel SVM model, we create a Gram matrix and find the components for the quadratic programming problem and solve it using the optimization library 'cvxopt'. We use the solution to find the support vectors by cutting it off at 1e-6 (Support vectors have non zero lagrange multipliers). The support vectors are then fitted with the intercept. The prediction for each test observation/sequence is then performed by checking the sign of :

$$f(x) = \sum_{i}^{N} \alpha_i y_i (x_i^T x_i) + b$$

The predictions added to one file along with the Id's. We tried the algorithm with linear kernel first and tested it with cross validation. The accuracy with linear kernel was 59%. Next, we tested the algorithm with rbf and polynomial kernels and obtained the highest score with the polynomial kernal of degree 3. The score obtained on Kaggle public leaderboard for this was 0.67 for penalty coefficient C = 1. The same algorithm was tested with different values of the penalty coefficient but the score did not deviate a lot from 0.67 with different values of C ranging from 0.1 to 10.

We tried other approaches that performed worse on Kaggle leaderboard:

- Using logistic regression on BOW dataset
- Using linear kernel with penalty coefficient c = 10
- Using rbf kernel with penalty coefficient with a range of penalty coefficients
- Using Gaussian kernel with a range of penalty coefficients
- Using polynomial kernel with degree 2 and 4 or more
- Creating features with k-mers of length 4 or 6

The different approaches with the scores obtained(Public leaderboard) is listed in the Table 2.

Kernel	Penalty Coefficient	Score
Linear	10	0.59
RBF	0.1-10	0.50 - 0.62
Gaussian	0.1-10	0.62
Polynomial with degree 2	1	0.60
Polynomial with degree 3	1	0.67
Polynomial with degree 4	1	0.59

Table 2: Different Approaches and scores

3 Conclusion

After having tested different approaches with the Bag-Of-Words dataset, it was clear that the representation was not very optimal and it was better to work with the sequences directly for improved performance. Therefore, we decided to work with k-mers with a length of 5 which proved to be the most optimal in our case. Since, the size of the training data was not too large, simple algorithms like logistic regression also gave decent results with some hyperparameter tuning but kernel SVM gave better results. After trying different approaches with different kernels and penalty coefficients, polynomial kernel with a degree of 3 helped us achieve a score of 0.69 on the private leaderboard securing the 9^{th} place among 29 teams.