Kernel Methods for Machine Learning: Data Challenge Report - AMMI 2020

Team: SO, Members: Volviane Saphir Mfogo and Sewade Olaolu Ogun

May 31, 2020

Abstract

The data challenge was made to enable groups learn how to implement machine learning algorithms, gain understanding about them and adapt them to structural data. The task was to predict 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. Our team implemented the Kernel Ridge Regression(KRR), Kernel Logistic Regression(KLR) and Kernel SVM.

I. Introduction

The Kaggle competition enabled us to apply the theoretical principles of kernel methods seen in class to a practical dataset. The dataset contains 2000 training DNA sequence and 1000 DNA sequence to be predicted. As a start, an already transformed data was provided which is based on bag of words representation. For the transformed data, all the subsequences of length l (here l=10) are extracted from the sequences and are represented as a vector of 4xl dimensions using one-hot encoding (with A=(1, 0, 0, 0), C=(0, 1, 0, 0), G=(0, 0, 1, 0), T=(0, 0, 0, 1)). However, this was not an optimal representation for the challenge so other methods were considered.

II. Methods

The idea is to map each string $x \in X$ to a vector $\phi(x) \in F$, then train a classifier for vectors on the images $\phi(x_1), ..., \phi(x_n)$ of the training set (KRR, KLR, K-SVM...). For the challenge, due to limited time available, we implement 2 mappings which we believed were good for DNA sequence classification.

- Spectrum Kernel - Base kernel
- Weight Degree Kernel

i. Spectrum Kernel

The spectrum kernel was our base kernel. It involved indexing the feature space by fixed-length strings, i.e.,

$$\phi(x) = (\phi_u(x))_{u \in \mathcal{A}^k}$$

We applied different spectrum lengths and performed cross validation to get the best length. Length 6 showed to perform better than others. It finds k-mers, which are all $L - k + 1$ possible substrings of length k that are contained in the sequence x of length L.

ii. Weight Degree Kernel

The weight degree kernel compares two sequences x and x' by summing all contributions of k-mer matches of lengths $k \in \{1, ..., d\}$ weighted by coefficients

$$\beta_k = 2(d - k + 1)/(d(d + 1))$$

where d is the degree. For this challenge, we experimented with weight degree kernel of $d \in \{4, 5, 6, 7, 8, 9, 10\}$.

Table 1: *Table showing result of experiments*

Name			
Model	Parameters	Val LB	Pub LB
KRR	$\lambda = 1e-4$	0.6713	0.672
KLR	$\lambda = 0.1$	0.6733	0.688
K-SVM	$C = 0.1$	0.6767	0.6740

III. Models - Kernel Methods

The different methods which were implemented for this challenge include;

- Kernel Ridge Regression (KRR)
- Kernel Logistic Regression (KLR)
- Kernel Support Vector Machines (K-SVM)

The classical l2-regularized models can be extended to make use of kernels due to the kernel trick. The kernel trick enable us to perform inner-products in the feature space of data points without directly computing the features.

The kernels were first computed ahead and saved as a python pickle file before training and predictions are done. This helps to enable reuse of these features and reduce the cost and time of computation.

IV. Results

Here, we highlight the results of the experiments we performed. Table 1 shows the results of the final experiments performed for the challenge. Weight degree kernel of $d = 10$ gave the most informative features for the challenge. All models above made use of this kernel for their final predictions. Although, we tried other degrees, none gave better performance even after cross validation.

V. Discussion

The challenge helped us to learn about how machine learning tools can be applied to unstructured data, most especially DNA sequences. The challenge also showed that classifying DNA sequence is a difficult task and the feature space of the data is highly non-linear. It is therefore necessary to map the data to a space where the data can be linearly separable. Also, we tried not to overfit in this new space, because with the very high dimension of the data and few examples given, it was easy to start overfitting. We, therefore, performed cross-validation on the data and made use of regularization parameters to find the fit that generalizes to the test set. In clonclusion, due to the limited time, it was difficult to consider many other kernels and maybe combinations of them, but, all-in-all we learnt alot applying theory to practice.

References

[Nojoomi, 2017] Nojoomi, Saghi, and Patrice Koehl (2017). A weighted string kernel for protein fold recognition. *BMC bioinformatics*, vol. 18,1 378

[J.P Vert, 2020] Julien Mairal and Jean-Philippe Vert(2020). Kernel Methods in Machine Learning *http://members.cbio.mines-paristech.fr/ jvert/svn/kernelcourse/slides/-ammi2020/ammi2020.pdf*, pg 193.