Report: Data challenge
Kernel methods for machine learning

Zineb ZIANI Mohammad KHOSHNAZAR

M2 SIAM / MOSIG
Kernel methods

Supervisor
Julien MAIRAL

Contents

1 Introduction

In this challenge, we have a sequence classification task which is predicting whether a DNA sequence region is binding site to a specific transcription factor. For that we will work with three data-sets corresponding to three different transcription factors which are regulatory proteins that bind specific sequence motifs in the genome to activate or repress transcription of target genes. In the rest of the paper, we present the support vector machine method, describe the kernels used to present and discuss our results and finish by a conclusion.

Rank and code:

Public rank 5 with accuracy 0.71800.

Private rank 6 with accuracy 0.70600.

The link to our code is as follows: https://github.com/zianizineb/data-challenge.git

2 support vector machine

Support vector machines are a set of supervised learning methods used for classification. It has a lot of advantages and the one why we decided to is use it is that a different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels reason why we use it in our challenge. This algorithms use a set of mathematical functions that are defined as the kernel. The function of kernel is to take data as input and transform it into the required form. Different SVM algorithms use different types of kernel functions. These functions can be different types, we will therefore present the kernels that we used in the following pargaph. In this challenge we used C-SVM algorithm. The C parameter tells the SVM optimization how much you want to avoid misclassifying each training example.

3 Kernel and Kernel methods

To proceed with our kernel method, we used several and different kernel functions on our three data-sets corresponding to the three different transcription factors.

•The spectrum kernel, for use with support vector machines in a discriminative approach to the protein classification problem, this kernel is simple and efficient to compute and in our case, spectrum kernel(k) related to spectrum kernel and term k shows all the possible sub-strings of length k.

•The Weighted Degree Kernel efficiently computes similarities between sequences while taking positional information of k-mers into account. In our case, we noted it as Weighted degree(d) where d parameter present the order.

•The Weighted Degree kernel with shifts, shifts the two sequences against each other in order to tolerate a small positional variations of sequence motifs. Conceptually, it is a mixture between

the Weighted Degree and the spectrum kernel. We noted it as weighted degree shift(d,s) in our code, with a shift size of s and degree of d.

•Mismatch Kernel measures the sequence similarity based on shared occurrences of k-length subsequences, counted with up to m-mismatches. In our code, we noted mismatch kernel(k,m) which compute the m mismatches for k-mers.

•String kernel consists of using of simple, yet expressive, sub-string features to compute a similarity function between sequences. String kernel(lbda, k) is our notation in the code where the k parameter shows length of k-mers and lbda related to a float number.

•The Gappy Kernel is Generally proposed after using the spectrum kernel where some similarities are lost between sequences. Instead of only recognising all k-mers in a sequence for a given k, it also counts how many k-mers with a certain number of gaps(g) appear in the sequence. We noted it by gappy kernel(g,k) where the parameters k shows the k-mers and g is defined for the gaps.

•The normalized Kernel is used to balance the weight of the different kernels.

Learning algorithms based on kernels have been used with much success in a variety of tasks as classification algorithms such as support vector machines reason why we decide to use Learning Non-Linear Combinations of Kernels as a kernel method, which combine all the kernels seen above. A standard regression problem is considered in this algorithm where the learner receives a training sample of size m. The learner must select a hypothesis among a family of the hypothesis that reproducing kernel Hilbert space(RKHS). In the paper, they considered a derivative of Kernel Ridge Regression as a learning algorithm. For learning non-linear kernels and solving the min-max problem they directly solved the inner maximization problem.

4 Result and conclusion

After some implementation, we got the best result from the Non-Linear Combination of the kernel that is proposed by Google in 2009. We implemented it with a different number of kernels and our best results came out with 9 kernels with 5-fold cross-validation of hyperparameter. Our accuracy on the validation sets 1,2 and 3 are: 0.6673, 0.6974 and 0.7430. We had not a huge change in the accuracy and rank in private and public leaderboard so can say that we did our job well. However, a huge problem with our work was the time of running. It took more than 19 hours for the first time to create the kernels. Even with the rechange function of the data that we defined in the general part of the code to speed up our algorithm. And this led to an obstacle to did a huge change in the codes. In this challenge, we wanted to do some classification on DNA sequences. During this work, we became familiar with how to apply the method of machine learning and how to combine techniques to get better results.

References

Learning Non-Linear Combinations of Kernels, Corinna Cortes, Mehryar Mohri, Afshin Rostamizadeh

https://scikit-learn.org/stable/modules/svm.html

https://data-flair.training/blogs/svm-kernel-functions