Data Challenge: Classification of DNA sequence regions for transcription factor binding sites

Quentin Chan-Wai-Nam Imke Mayer

quentin.chan-wai-nam@ens-paris-saclay.fr imke.mayer@ens-paris-saclay.fr

1 Introduction

The goal of the data challenge was to predict whether a DNA sequence region is a binding site for a specific transcription factor or not. The prediction is simply based on DNA sequences of 101 nucleotides (A,T,C,G), i.e. each data point is an element of the set $\{A,T,C,G\}^{101}$

Throughout the project we designed different kernels to apply on the data points and we then trained our kernel SVM classifier with these different kernels on the three training sets (each one containing 2000 DNA sequences of length 101).

2 Method

2.1 Spectrum kernel and its variants

Spectrum After having implemented and tested several generic kernels on the bag-of-words representation (giving scores up to 67% on the public test set), we went on testing the Spectrum kernel applied directly on the sets of DNA sequences. We implemented two versions of this kernel, one using tries (prefix trees) and one using pre-indexing. Because of the small size of the alphabet under consideration, $|\{A,T,C,G\}| = 4$ we used the pre-indexed version which allows for faster computations in our case. The spectrum kernel having one hyperparameter to tune, the length k of the considered substrings in a given string, we manually searched for the best $k \leq 101$ for each dataset: for Xte0 and Xte1 we found $k = 5$ and $k = 7$ respectively. The tuning of k for the set Xte2 was trickier and we were not able to find a choice of k yielding a better accuracy (without overfitting) than a well-tuned Gaussian kernel on the BoW representation of Xte2. We will discuss the difficulty of predicting the labels for this dataset further on. Note that we center the data in the feature space for all kernels.

Multiple spectrum Intuitively, having information about substrings of different lengths for a given sequence is more helpful than having information about substrings of a sole given length in the sense that we can find a trade-off between overfitting and underfitting by well combining several k-spectrum kernels. Following this intuition, we designed a new kernel based on this one by using the property that any positively weighted sum of p.d. kernels is still p.d. More precisely, given k_1, k_2,... the possible lengths of consecutive substrings ($k_j \leq 101$), let $\mathbf{K}_{k_1}, \mathbf{K}_{k_2}$,... be the Gram matrices on the dataset of the respective spectrum kernels. Then the multiple spectrum kernel is defined via the Gram matrix $\mathbf{K}$ as the sum of all these $\mathbf{K}_{k_j}$.

We enhanced the multiple spectrum kernel by noting that the data representation Φ is the count of every sequence of characters of lengths k_1, k_2, $\cdots$, even those that were not observed during the training phase. In order to avoid overfitting, we chose to only keep the dimensions in Φ that were observed during training. We call this kernel the mult. spectrum kernel w/ dim. reduction.

Stacked kernels Finally, we also tried to apply a Gaussian kernel on the representations Φ given by the mult. spectrum kernel w/ dim. reduction. This kernel, and others, turned out to overfit, due to the large increase in dimensionality but the overall accuracy on the test set was still the best when using a Gaussian kernel on the Φ given by the multiple spectrum kernel (at least for set 0 and 1). We report the accuracies achieved on the training and test sets for the previously mentioned kernels in Table 1.

Kernel	Set 0		Set 1		Set 2	
Spectrum	k= 5	$\lambda = 0.028$	k=7	$\lambda = 0.025$	k=4	$\lambda = 0.245$
	0.861 (*0.800*)		0.999 (*0.888*)		0.688 (*0.650*)	
	Test set accuracy: 0.75666 (public) / 0.75266 (private)					
Multiple Spectrum	$k_{(s)}$=[1:8]	$\lambda = 0.145$	$k_{(s)}$=[1:8]	$\lambda = 0.085$	$k_{(s)}$=[1:8]	$\lambda = 0.083$
	0.950 (*0.748*)		0.995 (*0.880*)		0.980 (*0.643*)	
	Test set accuracy: 0.78880 (public) / 0.78133 (private)					
Multiple Spectrum with dim. reduction	$k_{(s)}$=[4:8]	$\lambda = 0.166$	$k_{(s)}$=[5:7]	$\lambda = 0.117$	$k_{(s)}$=[1:8]	$\lambda = 0.254$
	0.970 (*0.754*)		0.986 (*0.888*)		0.880 (*0.647*)	
	Test set accuracy: 0.79600 (public) / 0.77933 (private)					
Multiple Gaussian Spectrum	$k_{(s)}$=[1:8]	$\lambda = 6.52 \times 10^{-6}$ $\gamma = 3.3$	$k_{(s)}$=[1:8]	$\lambda = 2.96 \times 10^{-5}$ $\gamma = 4.6$	Used MS w/ dim. red.	
	1.00 (*0.759*)		0.982 (*0.889*)		0.950 (*0.667*)	
	Test set accuracy: 0.78733 (public) / 0.78800 (private)					

Table 1: Training and test accuracies estimated by 5-fold cross-validation with kernel SVM classifier. The latter are given in brackets. All the kernels are centered in the feature space.

2.2 Classifier and parameter tuning

In order to choose the kernel parameters k, γ and the SVM parameters λ, we used a systematic search approach for λ based on 5-fold cross-validation on the training sets. When the model is coherent with the data, this approach gives reasonably reproducible bounds on λ, that enables us to find values for the hyper-parameters of the model which allow a generalization to new data. We used Python and optimized our code so that the computation time is < 5 min by complete submission for these kernels.

2.3 Quick note on other possible kernels

For our multiple Spectrum kernel we treated each sub-kernel equally but it would also have been possible to weight them differently in the sum. The problem with this supplementary parameter of sub-kernel weights is the difficulty of tuning them correctly (unless using strong priors on the importance of each sub-kernel).

We concentrated our work on variants of the Spectrum kernel and did not attempt to use the Substring kernel because of its computational complexity which we did not achieve to reduce sufficiently in order to be able to train our classifier with this kernel. Further testing of other kernels such as the previously mentioned examples could have improved our results since it appears that none of the string kernels we tested was well-suited for the set Xtr2, as shown by the estimated test accuracies in Table 1. The string kernels we used tended to strongly overfit the data, especially the combinations of a string kernel and a Gaussian kernel or a Polynomial kernel.

3 Conclusion

The construction and empirical study of several kernels, either generic or specifically designed for strings, allowed us to set apart differences between the three datasets in the sense that the prediction of transcription factor binding sites was more or less difficult for the three sets. A possibility to partially overcome this issue was the combination of several kernels which allowed us an overall increase of 4 basis points on the public dataset accuracy despite the through moderate predictive performance on Xtr2/Xte2.