Data Challenge Report for Machine Learning with Kernel Methods

Team: TL, Member: Le Tran Ngoc Tran

Universite de Paris

tran.le-tran-ngoc@etu.u-paris.fr,

Abstract -

This report is associated to the data challenge for Kernel Methods course in MVA program 2020-2021. The public score is **0.63533** *ranking* **58/74** *and the private score is* **0.62533** *ranking* **62/74**. *The goal of this challenge is to predict whether a DNA sequence region is binding site to a specific transcription factor. We present the methods and how to implement it to get the predicted results. The source code is available here:* https://github.com/ltngoctran/Kernel_Methods_MVA

1 Introduction

To solve the challenge, we try to apply some methods that we have seen in class, we use both sequences and feature matrices for training. For DNA sequences, we have used the two kernels which are spectrum and mismatch string kernels. Otherwise, radius basis function (rbf), polynomial and linear kernels are used. To perform binary classification task, we have built two kernel classifiers which are logistic regression and support vector machine. In this report, we will give a brief description of kernels are used and the table illustrates the accuracy of each model we used are given.

2 Building Kernels for DNA sequences

We would like to divide this section into two main part: the first one for the simple kernels like the linear, polynomial and Rbf for training feature matrices. The other one for string kernels to train DNA sequences which is motivated from the articles [1] and [2].

2.1 Kernels for numeric data

- The *linear kernel*: $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$
- The *polynomial kernel*: $K(\mathbf{x}, \mathbf{x}') = \left(\mathbf{x}^T \mathbf{x}' + a\right)^d$
- The *RBF kernel*: $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right)$

2.2 Kernels for string

2.2.1 Spectrum kernel

On the input space $\mathcal{X}$ of all finite length sequences of characters from an alphabet $\mathcal{A}$, $|\mathcal{A}| = l$. Given a number $k \geqslant 1$, the $k-$spectrum of a biological sequence is the set of all the $k-$length subsequences that it contains; we refer to such a $k-$length subsequences as a $k-$mer. We can assign to a sequence $\mathbf{x}$ a vector given as follows: for each $k-$mer α, the coordinate indexed by α will be the number of times α occurs in $\mathbf{x}$. This gives the $k-$spectrum feature map:

$$\Phi_k(x) = (\phi_a(x))_{a \in \mathcal{A}^k}$$

where $\phi_a(x) =$ number of times a occurs in $\mathbf{x}$. Thus, the $k-$spectrum kernel $K(\mathbf{x}, \mathbf{x}')$ is given by:

$$K_k(\mathbf{x}, \mathbf{x}') = \langle \Phi_k(\mathbf{x}), \Phi_k(\mathbf{x}') \rangle$$

2.2.2 Mismatch kernel

For a more sensitive and biologically realistic kernel, we want to allow some degree of mismatching in our feature map. For a fixed $k-$mer $\alpha = \alpha_1, \alpha_2, \cdots, \alpha_k$, with each a_i a character in $\mathcal{A}$, the $(k, m)-$pattern generated by α is the set of all $k-$length sequences β from $\mathcal{A}$ that differ from α by at most m mistmatches. We denote this set by $N_{(k,m)}(\alpha)$, the 'mismatch neighborhood' around α.

We can now define our feature map into the l^k-dimensional feature space: if α is a fixed $k-$mer, we first define the $\Phi_{(k,m)}$ on α by:

$$\Phi_{(k,m)}(\alpha) = (\phi_\beta(\alpha))_{\beta \in \mathcal{A}^k}$$

, where $\phi_\beta(\alpha) = \begin{cases} 1 & \text{if } \beta \text{ belongs to } N(k,m)(\alpha) \\ & \text{or } \alpha \text{ belongs to } N(k,m)(\beta) \\ 0 & \text{otherwise} \end{cases}$

We define the fearure map on an input sequence $\mathbf{x} \in \mathcal{X}$ is as the sum of the feature vectors for the $k-$mers in $\mathbf{x}$:

$$\Phi_{(k,m)}(\mathbf{x}) = \sum_{k-\text{mers } \alpha \text{ in } \mathbf{x}} \Phi_{(k,m)}(\alpha)$$

Then, the $(k,m)-$mismatch kernel is given by

$$K_{(k,m)}(\mathbf{x},\mathbf{x}') = \langle \Phi_{(k,m)}(\mathbf{x}), \Phi_{(k,m)}(\mathbf{x}') \rangle$$

2.3 Normalization

We also apply normalize the kernels via

$$K(\mathbf{x},\mathbf{x}') \longleftarrow \frac{K(\mathbf{x},\mathbf{x}')}{\sqrt{K(\mathbf{x},\mathbf{x})}\sqrt{K(\mathbf{x}',\mathbf{x}')}}$$

which results in a p.d kernel. In the training process, we realize that applying normalization does not improve the result.

3 Building classifers for our task

3.1 Logistic regression

We implement iteratively reweighted least-square (IRLS) method to solve the standard linear logistic regression. The detail of this method can be found in the lecture.

3.2 Support vector machine (SVM)

For SVM, we consider the dual form as the minimization of a quadratic function under box constraints

$$\min_{\alpha\in\mathbb{R}^n} \left\{ q(\alpha) = \frac{1}{2}\alpha^T K\alpha - \alpha^T y \right\} \text{ s.t } \forall i, 0 \leqslant y_i\alpha_i \leqslant C$$

There are two libaries which can be used for this optimizing problem: cvxopt and quadprog (quadratic programming solver). In fact, the libary cvxopt is faster than quadprog from our implementation.

4 Experiments

4.1 Numeric data

First, we implement directly the numeric data which is feature matrices Xtrk-mat100.csv and Xtek-mat100. The result given by the kernels in subsection 2.1 is not good and the accuracy on the validation set is around ≈ 0.59.

4.2 Sequence data

The rest of experiments, we focus on training with string kernel. For both the spectrum and mismatch kernels, we run many cases with different k, anologously, with different k and m. The result is given by spectrum and mismatch kernel is not really different but the time consuming of mismatch is much more than the one of spectrum kernel. Therefore, we refer to use the spectrum kernel.

Id set	Validation accuracy
0	0.61
1	0.62
2	0.72

Table 1. Implementation: spectrum kernel $k = 7$ with SVM $C = 10$.

4.3 Final results

Our best submission on the public leaderboard is the spectrum kernel $k = 7$ along with SVM $C = 10$ as in Table 1. In fact, we try to get the optimal value for the parameter C but the result is not really different on the value of C from 0.1 to 10 and we realize that the result is very sensitive.

We also perform the summing of spectrum kernel for k from 5 to 8, the validation accuracy is better (0.64, 0.68, 0.73 for 3 set respectively); the public score is lower 0.63066 (compared to the best public score) and the private score is 0.63466. Our best private score is 0.64533 for spectrum kernel with $k = 8$ and SVM with $C = 1$.

5 Conclusion and Discussion

We perform the classification for DNA with both numeric and string data and the accuracy is better for DNA sequences. The spectrum kernel and mismatch kernel give similar result in our implementation, but the mismatch kernel takes more time for the computation of the Gram matrix.

We believe that if we use the spectrum and mismatch kernel with parameter k from 10 to 16, the results would be better. And of course the time consuming will be much. Because of the limitation of computer, we only perform for up to $k = 8$.

References

[1] Christina Leslie, Eleazar Eskin, and William Stafford Noble. The spectrum kernel: A string kernel for svm protein classification. In *Biocomputing 2002*, pages 564–575. World Scientific, 2001.

[2] Christina S Leslie, Eleazar Eskin, Adiel Cohen, Jason Weston, and William Stafford Noble. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467–476, 2004.