Prediction of Signal Peptide Cleavage Site Using Supervised Learning

HUANG Yuxing, LE QUANG Dung*
École Polytechnique

Abstract
In this project, we investigate protein targeting, specifically focusing on the identification of cleavage sites in signal peptides. The problem is formulated as a binary classification task, aiming to predict the cleavage site based on the (p, q) neighborhood of amino acids in a protein sequence. We employ *Support Vector Machine* (SVM) with three different kernels: scalar-product kernel, substitution matrix-based kernel, and probabilistic kernel, comparing their performance on three datasets representing eukaryotes, Gram-positive prokaryotes, and Gram-negative prokaryotes. Our experimental results suggest that, with a large enough sample size, both the substitution matrix-based kernel and the probabilistic kernel can accurately predict cleavage sites. However, the substitution matrix-based kernel is faster and thus recommended. We conclude that using this kernel with parameters $p = 13$, $q = 2$, and at least 10,000 samples can achieve high prediction accuracy (F1-score of 0.991).

Contents

1 Introduction

Protein targeting is the process of guiding proteins to their proper location based on information contained within the protein, specifically a short sequence called the signal peptide near the N-terminal side. This process is essential for proper protein function, and understanding signal peptides and their cleavage site is important for drug development.

For example, the following sequence is the beginning of a protein, where the cleavage site is marked as the bond between the two underlined `AR` amino acids:

`MASKATLLLAFTLLFATCIARHQQ...`

It has been shown in [2] that the cleavage site may be characterized by amino acids in its close neighborhood, where the neighborhood is defined as the p amino acids before and the q after. Thus, values p and q define a neighborhood of $p+q$ amino acids in length (we call it a (p, q)-neigborhood of the j-th position), which indicate the cleavage site.

- **Problem statement:** From the analysis above, our problem is reduced to be a binary classification problem. That is, given a sequence of amino acids $a_{j-p}a_{j-p+1}\dots a_j\dots a_{j+q-1}$ which represent the (p, q) neighborhood of the j-th position of the protein whole sequence of amino acids is $(a_i)_{i=0,\dots,l-1}$, determine that the protein cleave at the position j or not. Also, we must find out which values of p and q will be best for our prediction (as suggested in the [2], $p = 13$, $q = 2$).

*Contact: dung.le-quang@polytechnique.edu

• **Method:** To solve this problem, we use the *Support Vector Machine* (SVM) with three different kernels: the kernel using scalar product, the kernel using substitution matrix, and the probabilistic kernel in [1]. We compare their performances in three difference data sets, representing the sequences of amino acids of three different organisms: eucaryotes, Gram-positive prokaryotes, Gram-negative prokaryotes.

2 Support Vector Machine

2.1 Theoretical background

Let us recall some basic notion of the SVM. SVM with kernel is an extension of the traditional SVM algorithm that can handle non-linearly separable data by mapping the input features to a higher-dimensional space using a non-linear function called the kernel function. The data points are then separated by finding a hyperplane that maximizes the margin between the two classes in this higher-dimensional space. Concretely, given a *reproducing kernel Hilbert function space* $\mathcal{H} \subset \mathbb{R}^{\mathbb{R}^d}$, a *feature function* $\Phi : \mathbb{R} \to \mathcal{H}$ inducing the *kernel* $k = \langle \Phi(\cdot), \Phi(\cdot) \rangle : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, and the data sets in $\mathbb{R}^d$ with their labels $(x_i, y_i)_{i=1}^n$ ($y_i \in \{-1, 1\}$), we need to find

$$\arg\min_{\beta,\beta_0} \|\beta\|^2 \text{ s.t } y_i(\Phi(x_i)^\top \beta - \beta_0) \geq 1,$$

$\forall i = 1, \ldots, n$. Using the representation theorem, i.e "kernel trick", we only need to find $\hat{\beta}$ of the form

$$\hat{\beta} = \sum_{i=1}^{n} \alpha_i y_i \Phi(x_i).$$

For a test data x, the predicted label is

$$\hat{y} = \text{sgn}\left(\beta_0 + \sum_{i=1}^{n} \alpha_i y_i K(x_i, x)\right).$$

Normally, the kernel k must be a positive semi-definite, i.e. $\forall n \in \mathbb{N}$, $\forall x_1, \ldots, x_n \in \mathbb{R}^d$, the Gram matrix $(k(x_i, x_j))_{i,j}$ is positive semi-definite (Mercer's conditions). In some case, k can be not semi-definite, we call in this case a *pseudo-kernel*.

2.2 Encoding

Noting that our data is the sequences of amino acids, which are the sequences of character. To using the kernel method in our problem, we must encode the data into the numerical vector. There are many ways to encoding these sequences, such as binary encoding, one-hot encoding, BLOSUM encoding, word encoding,... In our project, we choose the *one-hot encoding*, the simplest ones, that is, each amino acid is represented by a vector of zeros with a single one at the position corresponding to the amino acid in the alphabet. Because there are 20 characters used to represent the amino acids, each amino acid is encoded by a vector of length 20. Thus, for a sequence of amino acid of length n, the encoded vector has the length $20n$.

2.3 Implemented kernels

In the project, we consider the data sets as the points $\{a^i\}_{i=1,\ldots,n}$, where $a^i = a_0^i \ldots a_{p+q-1}^i$ be a sequence of amino acids. We consider the three types of kernels. Given two sequences $a = a_0 \ldots a_{p+q-1}$, $b = b_0 \ldots b_{p+q-1}$, after encoding them, we can define the three kernels as the following:

• **Scalar-product kernel**: this kernel is defined as the scalar product between the two encoding vectors using the one-hot encoding method. Obviously, this kernel simply counts the number of common letters between the two corresponding words.

• **Kernel based on substitution matrix**: this kernel bases on the matrix of score of similarity M, where $M(x, y)$ denotes the similarity between any pair (x, y) of amino acids. From this, we have the similarity score

between two words a and b

$$S(a,b) = \sum_{i=0}^{p+q-1} M(a_i, b_i).$$

Note that such a score is not related to a distance and that it does not satisfy Mercer's conditions. But it is safe to use it as a pseudo dot product to define a RBF kernel, for instance.

For the choose of substitution matrix, we have many choice, such as `BLOSUM62`, `BLOSUM80`, `GONNET`, `PAM250`, `PAM30`,... They are integrated in the module `Bio.Align.substitution_matrices` of the package `Bio`. In our project, we use the `BLOSUM62` ones.

• **Probabilistic kernel**: In his paper [1], J.P.Vert propose a new probabilistic kernel that the value of the kernel with the two inputs a and b:

$$K(a,b) = \frac{1}{2^{p+q}} \prod_{i=0}^{p+q-1} \phi(a_i, b_i),$$

where

$$\phi(x,y) = \begin{cases} p_i(x)p_i(y) \text{ if } x \neq y \\ p_i^2(x) + p_i(x) \text{ if } x = y \end{cases}$$

$p_i(x)$ is the probability that the amino acid x appears at the position i. For more details, please refer to the original paper [1].

3 Experimental implementation

We first fix the value of p and q, and we increase the number of the training samples gradually to see how the performance of the three model (quantified by *F1-score*) changes as sample size changes.

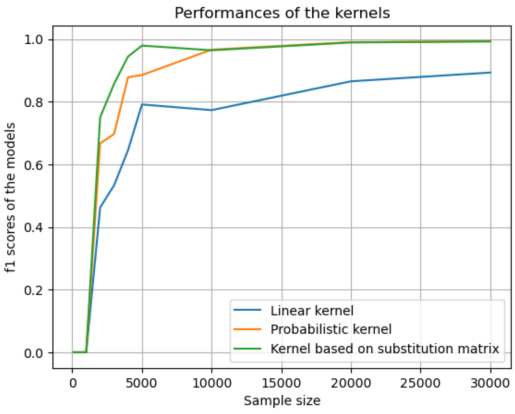

Figure 1: Performances of different kernels

In the figure 1 we see that their F1-score remains when the sample size is less than 1000, and it increases rapidly when the sample size is in the interval $[1000, 5000]$. After that, their growth rate slow down and the F1-score of second and the third kernel converges to 1.0.

And in table 1 we see that the linear kernel is the fastest, while the probabilistic kernel is the slowest.

Finally we fix the sample size at 10000 and change the value of p and q to find the best p, q who provide with the highest F1-score (table 2). We find that the model has the highest F1-score when $p = 13$ and $q = 2$, which coinsides with the statement in [2].

4 Conclusion

Among the three kernels, the linear kernel has the fastest speed, and it can predict the cleavage site to a certain extent, but the result is not accurate enough. While the kernel based on substitution matrix and the probabilistic kernel can perfectly predict the cleavage site, provided with a large enough sample size. Their performance is very close when the sample size is bigger than 10,000, but the kernel based on substitution matrix is much faster. As a result, we suggest that we use the kernel based on substitution matrix with the parameter $p = 13$, $q = 2$, and at least 10,000 samples to achieve an accurate prediction (whose F1-score equals to 0.991).

Kernel	**F1-score** (test set)	**F1-score** (training set)	**Running time** (seconds)
Linear	0.773	0.838	5.73
Substitution matrix	0.964	1.000	16.24
Probabilistic	0.966	1.000	22.31

Table 1: Performance with different kernels when $p = q = 3$, $N = 10000$

(p, q)	**F1-score** (test set)	**F1-score** (training set)	**Running time** (seconds)
(3,3)	0.964	1.000	16.24
(5,5)	0.987	1.000	24.59
(13,2)	0.991	1.000	37.49
(13,3)	1.000	1.000	39.99
(13,5)	0.990	1.000	53.55

Table 2: Performance with different (p, q)

References

[1] VERT Jean-Philippe. Support vector machine prediction of signal peptide cleavage site using a new class of kernels for strings. 2001.

[2] Gunnar Von Heijne. A new method for predicting signal sequence cleavage sites. *Nucleic acids research*, 14(11):4683–4690, 1986.