The Second International Symposium on Optimization and Systems Biology (OSB'08)
Lijiang, China, October 31– November 3, 2008

Accurate Prediction of Translation Initiation Sites by Universum SVM

Tingting Gao[1] Yingjie Tian[2] Xiaojian Shao[1]
Naiyang Deng[1,*]

[1]College of Science, China Agricultural University, 100083, Beijing, China
[2]Research Center on Fictitious Economy and Data Science,Chinese Academy of Sciences, Beijing 100080, China

Abstract In order to extract protein sequences from nucleotide sequences, it is an important step to recognize points at which regions that start code for proteins. These points are called translation initiation sites (TIS). The task of recognizing TIS can be modeled as a classification problem. In this paper, we use a new pattern classification algorithm which has recently been proposed by Vapnik to deal with this problem. Numerical experiments proved the considerable improvement of this method compared with the leading existing approaches.

Keywords Support Vector Classification; Translation Initiation Site; Universum

1 Introduction

Translation, along with transcription and replication, are the major operations that related to biological sequences. The recognition of translation initiation sites (TISs) is essential for genome annotation and for better understanding of the process of translation. It has been recognized as one of the most critical problems in molecular biology, and this problem requires the generation of classification models to accurately and reliably distinguish the valid TISs from a set of false ones.

Machine learning techniques have been used successfully in TIS prediction using the mRNA or cDNA sequence. In Pedersen and Nielsen [1], an artificial neural network (ANN) was trained on a 203 nucleotide window centered on the AUG. They obtained results of 78% accuracy on start AUGs and 87% accuracy on non-start AUGs on their vertebrate dataset, giving an overall accuracy of 85%.

Zien et al.[2] obtain improved results on the same vertebrate dataset from Pedersen and Nielsen by using support vector machines (SVM). They show how to obtain improvements by appropriate engineering of the kernel function - using a locality-improved kernel with a small window on each position, a codon-improved kernel using codon structure in the downstream sequence and a Salzberg kernel using conditional positional probabilities. With the nucleotide-based kernels, they obtain an accuracy of 69.9% and 94.1% on start and non-start AUGs respectively, giving an overall accuracy of 88.1%.

*Corresponding author. E-mail: dengnaiyang@vip.163.com

Later, Wong et al.[3] shows that good performance can be obtained by simple feature generation and selection followed by a variety of standard machine learning methods. And in the follows, they repeat this approach, but use features generated from a translation of the mRNAs into the corresponding amino acids instead of the mRNAs directly, and they also use PCL(Prediction by Collective Likelihood of emerging patterns) [4] as the classification algorithm instead of traditional machine learning methods.

Although many approaches have been proposed to deal with this problem, there is still a great potential for the improvement of their accuracy. This is the motivation behind our research. In this paper, we aim to show that good performance comparable to the best results can be obtained by using simple feature generation and selection on the new pattern classification algorithm −Universum Support Vector Machine, which was first proposed by Vapnik [5]. The results from the TIS prediction are directly comparable with Li et al.[4]. The highest overall accuracy obtained is 96.51% which is better than previous results on this dataset.

This paper is structured as follows. In section 2, we describe the background related to our research, then in section 3 we briefly introduce the Universum SVM and derive the algorithm. In section 4 we present experiments and results. Finally in section 5 we make some concluding remarks .

2 Background and Problem Description

The main structural and functional molecules of an organismąs cell are proteins. Another important family of molecules is nucleic acids. The most common nucleic acids are deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). The term sequence is used to refer to the order of monomers that compose the polymer. A sequence can be represented as a string of different symbols, one for each monomer. There are twenty protein monomers called amino acids and five nucleic acid monomers called nucleotides. Every nucleotide is characterized by the nitrogenous base it contains: adenine (A), cytosine (C), guanine (G), thymine (T), or uracil (U). DNA may contain a combination of A, C, G, and T. In RNA U appears instead of T. A sequence of nucleotides has two ends called the $5'$ and the $3'$end. Moreover, it is directed from the $5'$ to the $3'$ end. Proteins are synthesized by the following process. DNA is transcribed into a messenger RNA (mRNA) molecule (transcription). Then mRNA is used as template for the synthesis of a protein molecule (translation). Translation, usually, initiates at the AUG codon nearest to the $5'$ end of the mRNA sequence. Figure 1 (cited from [6]) illustrates the process.

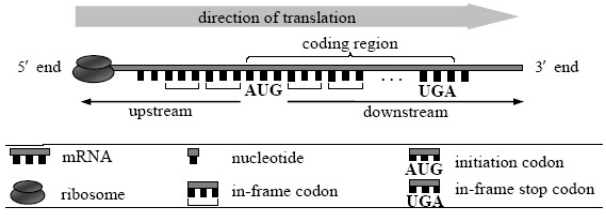

Figure 1

The position of the first nucleotide base pair (bp) in the start codon is denoted by translation initiation site (TIS). The aim of this work was to use a new method to correct

distinguish the valid TISs from a set of upstream and downstream false ones.

3 Methods

In this section, we will introduce a new pattern classification algorithm which has recently been proposed by Vapnik [5] and his coworkers [7]. First we discussed this idea of inference through Universum, then we present how it was realized as an algorithm.

3.1 Inductive Inference with Universum

Learning algorithms need to make assumptions about the problem domain in order to generalize well. These assumptions are usually encoded in the regularisation or the prior. A generic learning algorithm usually makes rather weak assumptions about the regularities underlying the data. An example of this is smoothness. More elaborate prior knowledge, often needed for a good performance, can be hard to encode in a regularisation or a prior that is computationally efficient too.

A prominent example of data-dependent regularisation is semisupervised learning , where an additional set of unlabelled data, assumed to follow the same distribution as the training data. A novel form of data-dependent regularisation was recently proposed by Vapnik [5]. The additional dataset for this approach is explicitly not from the same distribution as the labelled data, but represents a third class. This kind of dataset was first proposed by Vapnik under the name Universum, owing its name to the intuition that the Universum captures a general backdrop against which a problem at hand is solved. And the Universum plays the role of prior information in Bayesian inference. It describes our knowledge of the problem we are solving. According to Vapnik, a suitable set for this purpose can be thought of as a set of examples that belong to the same problem framework.

Although initially proposed for transductive inference, the authors of [5] proposed an inductive classifier where the decision surface is chosen such that the Universum examples are located close to it. Implementing this idea into SVM, they get the Universum Algorithms.

3.2 SVMs in the Universum Environment

The Universum Algorithm can be implemented using SVM techniques as follows.

First, mapping the training set

$$T = \{(x_1, y_1), \cdots, (x_l, y_l)\} \in (\mathscr{X} \times \mathscr{Y})^l \,, \tag{1}$$

where $x_i \in \mathscr{X} \subset \mathbf{R}^n, y \in \mathscr{Y} = \{-1, 1\}, i = 1, \cdots, l.$
and the Universum Set

$$U = \{x_1^*, \cdots, x_u^*\} \in R^n \,, \tag{2}$$

into Hilbert space

$$\{\Phi(x_1), y_1), \cdots, (\Phi(x_l), y_l\}, \tag{3}$$

and

$$\{\Phi(x_1^*), \cdots, \Phi(x_u^*)\}, \tag{4}$$

therefore in the quadratic optimization framework for SVM, the optimization problem is

$$\min_{w,b,\xi,\psi^{(*)}} \quad \frac{1}{2}(w\cdot w)+C_1\sum_{i=1}^{l}\xi_i+C_2\sum_{s=1}^{u}(\psi_s+\psi_s^*) \tag{5}$$

$$\text{s.t.} \quad y_i((w\cdot\Phi(x_i))+b)\geq 1-\xi_i,\xi_i\geq 0, i=1,\cdots,l \tag{6}$$

$$-\varepsilon-\psi_s^*,\leq (w\cdot\Phi(x_s^*))+b\leq\varepsilon+\psi_s, s=1,\cdots,u \tag{7}$$

where $\psi^{(*)}=(\psi_1,\psi_1^*,\cdots,\psi_u,\psi_u^*)^T$. This optimization problem is convex, and just like SVM the solution can also be computed through the corresponding dual optimization problem

$$\max_{\alpha,\mu,\nu} \quad \sum_{i=1}^{l}\alpha_i-\varepsilon\sum_{s=1}^{u}(\mu_s+\nu_s)-\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j K(x_i,x_j) \tag{8}$$

$$-\sum_{i=1}^{l}\sum_{s=1}^{u}\alpha_i y_i(\mu_s-\nu_s)K(x_i,x_s^*)-\frac{1}{2}\sum_{s,t=1}^{u}(\mu_s-\nu_s)(\mu_t-\nu_t)K(x_s^*,x_t^*)$$

$$\text{s.t.} \quad \sum_{i=1}^{l}y_i\alpha_i+\sum_{s=1}^{u}(\mu_s-\nu_s)=0 \tag{9}$$

$$0\leq\alpha_i\leq C_1,\ \ i=1,\cdots,l\,. \tag{10}$$

$$0\leq\mu_s,\nu_s\leq C_2,\ \ s=1,\cdots,u\,. \tag{11}$$

The decision function is then formulated as

$$f(x)=\text{sgn}(\sum_{i=1}^{l}\alpha_i^0 y_i K(x_i,x)+\sum_{s=1}^{u}(\mu_s^0-\nu_s^0)K(x_s^*,x)+b_0) \tag{12}$$

where $\varepsilon\geq 0$ is constant number, and C_1, $C_2\succ 0$ are the penalty parameter of the error term. Furthermore, $K(x_i,x_j)$ is called the kernel function. Though new kernels are being proposed by researchers, the following four basic kernels are often been used:

- linear: $K(x_i,x_j)=x_i^T x_j$.
- polynomial: $K(x_i,x_j)=((x_i\cdot x_j)+c)^d, c\geq 0$.
- radial basis function (RBF): $K(x_i,x_j)=exp(-\frac{\|x_i-x_j\|^2}{2\sigma^2}), \sigma>0$.
- sigmoid: $K(x_i,x_j)=tanh(\kappa(x_i^T x_j)+\upsilon), \kappa>0, \upsilon>0$

4 Implementation

4.1 Dataset of TIS

We use the vertebrate dataset provided by Pedersen and Nielsen [1]. Since the dataset is processed DNA, the TIS is ATG. In total, there are 13375 ATG sites. Of these possible translation initiation sites, 3312 (24.76%) are the true start TIS, while the other 10063 (75.24%) are non-TIS.

4.2 Feature Generation

When building feature space for classification, we use the approach provided by Li et al. [4]: a window centered at each ATG, with both upstream and downstream are 100

bases long, is generated from each ATG. So there are 203 bases indicated by A, T, C and G in each window. For each ATG site, we matched 3 nucleotides to 1 amino acid and count the frequency of each amino acid. We distinguish these amino acid as upstream or downstream regarding to that it appears before or after the centered ATG. Besides the single amino acid, we also considered the frequency of a pair of amino acid. Thus, the following types of features are generated :

(1) up-X(or down-X): which counts the number of times the amino acid letter X appears in the up-stream(or down-stream) part, for X ranging over the standard 20 amino acid letters and the special stop symbols.

(2) up-XY(or down-XY): which counts the number of times the two amino acid letters XY appear as a substring in the up-stream(or down-stream) part, for X and Y ranging over the standard 20 amino acid letters and the special stop symbols.

Here, we also use these three features: down4-G, up3-AorG, up-ATG. Finally, we got a feature space containing 927 features.

4.3 Feature Selection

As can be seen, the techniques described above can generate a large number of features for each sequence segment. Clearly, not every feature is important. Using only the more important features has the advantage of avoiding noise and speeding up subsequent construction of the recognition model. It is thus often desirable to discard weaker features. A number of general techniques can be used for this purpose [8, 9].

Here, we use the entropy method[8]. The basic idea of this method is to filter out those features whose value distributions are relatively random. For the remaining features, this method can automatically find some cut points in these features's value ranges such that the resulting value intervals of every feature can be maximally distinguished. If each value interval induced by the cut points of a feature contains only the same class of samples, then this partitioning by the cut points of this feature has an entropy value of zero. The smaller a feature's entropy is, the more discriminatory it is.

Applying the entropy method to the 927 features, the 10 features of lowest entropy are selected: up-ATG, down-STOP, up3-AorG, down-A, down-V, up-A, down-E, down-L, down-D, and up-G. Some of these 10 features also make good biological sense. The up-ATG feature makes sense because it is uncommon for an in-frame up-stream ATG to be near a translation initiation site, as this runs counter to the scanning model of eukaryotic protein translation [10]. The down-STOP feature makes sense because it is uncommon for an in- frame stop codon to be near a translation initiation site, as this implies an improbably short protein product. The up3-AorG feature makes sense because it is consistent with the well-known Kozak consensus signature observed at translation initiation sites [10].

4.4 SVM for TIS recognition

In this part, we use standard SVM [11] for TIS recognition. And the SVM learning was implemented using LIBSVM, available from: http://www.csie.ntu.edu.tw/ cjlin/libsvm.

In order to compare with the method proposed in the work of [4], we use the same approach of feature generation and feature selection, and three-fold cross-validation with RBF kernels is carried out in the whole dataset (13375) to evaluate the performance. Finally, an overall accuracy of 88.00 % is obtained as shown in Table 2 which is comparable

Table 1: Confusion Matrix

	predicted positive	predicted negative
actual positive	TP	FN
actual negative	FP	TN

Table 2: Prediction results of PCL and SVM

Classifier	Sensitivity	Specificity	Precision	Accuracy
PCL	84.72%	88.66%	71.09%	87.69%
SVM	87.17%	88.22%	71.69%	88.00%

to the 87.69 % in [4] , especially because when we take sensitivity into consideration, the sensitivity of 87.17 % is better than their 84.72%. The non-start ATGs outnumber the start ATGs in even greater ratios, therefore, the superior sensitivity of our methods may be more desirable. Sensitivity, Specificity, Precision, Accuracy and MCC of a classifier are defined as:
$Sensitivity = \frac{TP}{TP+FN}$,
$Specificity = \frac{TN}{TN+FP}$,
$Precision = \frac{TP}{TP+FP}$,
$Accuracy = \frac{TP+TN}{TP+FN+TN+FP}$,
and $MCC = \frac{TP*TN-FP*FN}{\sqrt{((TP+FN)*(TP+FP)*(TN+FN)*(TN+FP))}}$, where TP, TN, FP, FN are given in Table 1:

4.5 Universum-SVM for TIS recognition

In this part, we show how to boost the process of learning by choosing the appropriate Universum. Two numerical experiments are performed in this part.

First, to deal with the common imbalance problem in the prediction of TIS, we use the Under-Sampling Algorithm, that is, in each split, each partition contains the same number of positive and negative datapoints. Then the standard SVM and an Universum SVM are performed in three-fold cross-validation experiments. Furthermore, in order to explore what kind of Universum is useful, we construct two kinds of Universum:

(i)U_{noise}: whose features are generated following uniformly distribution;

(ii)U_{mean}: create an artificial sample by first selecting a random positive and negative example from the training set, and then constructing the mean of this two examples.

The results are shown in Table 3:

From Table 3, we can see that U_{noise} was included to show that not just any Universum helps, it has to be related to the problem of our domain. But U_{mean} can significantly improve the performance of SVM. And the overall accuracy of 96.51 % is better than previous results on this dataset.

Second, in order to explore how U_{noise} and U_{mean} influence the results, we sample the

Table 3: Prediction results of SVM and Universum SVM (U-SVM)

Classifier	Sensitivity	Specificity	Precision	Accuracy	Mcc
SVM	89.31%	95.11%	94.81%	92.21%	0.846
U_{noise}-SVM	89.32%	95.10%	94.81%	92.22%	0.845
U_{mean}-SVM	95.83%	97.19%	97.15%	96.51%	0.93

labelled sets of size 50, 100, 500, 1000 and 2000 from this dataset, and we use a test set of 1000 data randomly sampled from the remainder. The results for different training subset sizes and different Universum sizes are reported in Figure 2.

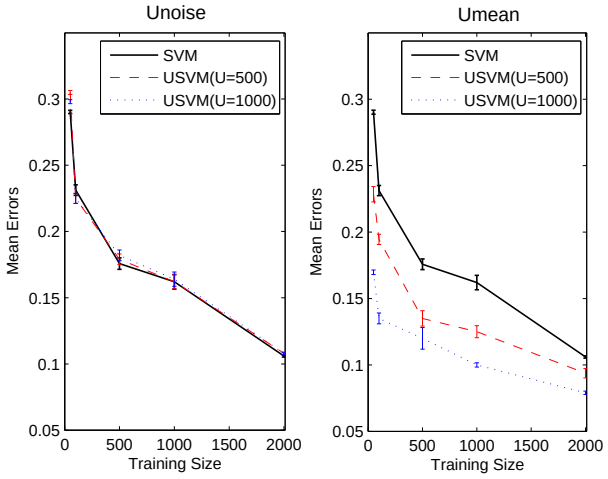

Figure 2

From Figure 2, we can see that, for U_{noise}, as expected, its performance converges to the performance of SVM (if choose C_u =0, the impact of the loss on the Universum points on the optimisation problem is zero. Therefore, a Universum-SVM with $C_u = 0$ is equivalent to a stand SVM). However, for U_{mean}, it has a positive effect on the accuracy. The role of the Universum becomes more important with decreasing training size. However, even when the training size is large, the Universum still has a significant effect on performance. The theoretical result that U_{mean} is an appropriate Universum set has been proved by Fabian Sinz in [12].

5 Discussion

In this paper, we considered the utilization of a set of a third class of data, termed the Universum [5], in order to achieve better accuracy for the prediction of translation initiation sites in genomic sequences. We applied this algorithm on a real-world dataset that contains processed DNA sequences from vertebrates, and we achieved satisfactory results.

We conclude by providing some directions for future work. There is a great variety of features that can be generated and describe a genomic sequence. Only a portion of them has been so far studied. Our future plans involve the experimentation with novel features, such as the regulatory signals relevant to this process. Additionally, we aim to use more datasets such as the prokaryotic genomes in order to verify the results of our method. Finally, how to choose the appropriate Universum is still the subject of research and under our consideration, and we expect to creat a meaningful Universum for other existing biomolecular problems to furtherly boost the performance.

Acknowledgments

This work is supported by the Key Project of the National Natural Science Foundation of China(No.10631070) and the National Natural Science Foundation of China (No.10601064).

References

[1] Pedersen,A.G., and Nielsen,H., Neural Network Prediction of Translation Initiation Sites in Eukaryotes: Perspectives for EST and genome analysis. Intelligent Systems for Molecular Biology 5,226-233 (1997)

[2] Zien, A.,Ratsch,G.,Mika,S.,Engineering Support Vector Machine Kernels that Recognize Translation Initiation Sites,Bioinformatics, 799-807,9 (2000)

[3] Zeng,F., Yap,H., Wong,L.,Using Feature Generation and Feature Selection for Accurate Prediction of Translation Initiation Sites, Genome Inform,13:192-200 (2002)

[4] Li,J.,Ng,S.-K.,and Wong,L.,Bioinformatics Adventures in Database Research, Proc.9th Int.Conf.on Database Theory, 31-46 (2003)

[5] Vapnik, V. N.,Estimation of Dependences based on Empirical Data, Berlin: Springer Verlag. 2nd edition (2006)

[6] Tzanis,G., Berberidis,C.,Alexandridou, A. and Vlahavas,I., Improving the Accuracy of Classifiers for the Prediction of Translation Initiation Sites in Genomic Sequences,Advances in Informatics, 426-436 (2005)

[7] Weston,J.,Collobert,R. ,Sinz,F.,Bottou, L. and Vapnik, V. N., Inference with the Universum. Proc.23 rd Int.Conf.on Machine Learning (2006)

[8] Fayyad,U., and Irani,K.,Multi-interval Discretization of Continuous-Valued Attributes for Classification Learning. International Joint Conferences on Artificial Intelligence 93, 1022-1029.

[9] Liu,H. and Chi,R. Sentiono., Feature Selection and Discretization of Numeric Attributes. In Proc. IEEE 7th Intl. Conf. on Tools with Artificial Intelligence,338-391 (1995)

[10] Kozak,M., An analysis of 5'-noncoding sequences from 699 vertebrate messenger RNAs.NAR,15:8125-8148 (1987)

[11] Deng,N.Y.,Tian,Y.J.,A New Method in Data Mining: Support Vector Machine. Science Press, Beijing (2004)

[12] Sinz,F., Chapelle,O., Agarwal,A., Schölkopf,B., An Analysis of Inference with the Universum (2007).Paper available at
http://books.nips.cc/papers/files/nips20/NIPS2007_ 0780.pdf