Analysis of binary feature mapping rules for promoter recognition in imbalanced DNA sequence datasets using Support Vector Ma....

Conference Paper · October 2008

DOI: 10.1109/IS.2008.4670503 · Source: IEEE Xplore

CITATIONS
9

READS
129

1 author:

Robertas Damasevicius
Kaunas University of Technology
197 PUBLICATIONS **938** CITATIONS

SEE PROFILE

Analysis of Binary Feature Mapping Rules for Promoter Recognition in Imbalanced DNA Sequence Datasets using Support Vector Machine

Robertas Damaševičius, *Member, IEEE*

***Abstract*— Recognition of specific functionally-important DNA sequence fragments is considered one of the most important problems in bioinformatics. One type of such fragments are promoters, i.e., short regulatory DNA sequences located upstream of a gene. Detection of promoters in DNA sequences is important for successful gene prediction. In this paper, a machine learning method, called Support Vector Machine (SVM), is used for classification of DNA sequences and promoter recognition. For optimal classification, 11 rules for mapping of DNA sequences into binary SVM feature space are analyzed. Classification is performed using a power series kernel function. Kernel parameters are optimized using a modification of the Nelder-Mead (downhill simplex) optimization method. The results of classification for drosophila and human sequence datasets are presented.**

***Index Terms*— DNA mapping rules, Support Vector Machine, data mining, promoter recognition, bioinformatics**

I. Introduction

Biomolecular data mining is the activity of finding significant information in DNA, RNA and protein molecules. The significant information may refer to periodicities, motifs, clusters, genes, protein signatures, grammar rules and classification rules. In this paper, we consider the problem of promoter recognition. Promoters are short regulatory DNA sequences that precede the beginnings of genes. It is important to distinguish between promoter and non-promoter sequences, because this distinction allows identifying starting locations of genes in uncharacterized DNA sequences.

Patterns in the promoter sequences within a species are known to be conserved, but there are many exceptions to this rule which makes the promoter recognition a complex problem. Although many complex feature extraction schemes coupled with several classifiers have been proposed so far for promoter recognition, the problem is still open [1]. The inability to predict promoters has became a real obstacle to genome-wide analysis of gene regulation by bioinformatics [2].

To solve the problem of promoter recognition among the DNA sequences, i.e., to positively classify promoter sequences, we use a Support Vector Machine (SVM) [3,4]. The SVM is a machine learning method for creating binary classification functions from a set of labeled training data. The accuracy of a particular parametric classifier on a given dataset will depend on the relationship between the classifier and the available dataset [5]. The SVM requires that each data instance is represented as a vector of real numbers in feature space. Hence, if there are categorical attributes, we first have to convert them into numeric data. Good mapping of data into feature space often allows achieving better classification accuracy [5].

However, the SVM can be ineffective in dealing with the classification problem when the ratio between the number of the target (positive) and non-target (negative) training instances differs significantly from the 1:1 ratio, and the number of the negative data instances is much higher than the number of the positive data instances, i.e., a training dataset is imbalanced [6]. When the SVM classifier is trained using the original imbalanced data set, it favors higher accuracy rate on the negative data, because the instances of the negative data class dominate the training set. As a result, the false-negative rate can be excessively high in identifying important target objects. There are two groups of approaches for addressing the imbalanced training data problem [7]. The *algorithmic* approach focuses on improving or proposing new classification algorithms, whereas the *data processing* approach focuses on modifying data set to improve the accuracy of the classification. In this paper, we analyze different data encoding schemes, which can achieve optimal classification results for the promoter recognition problem.

Several authors considered different binary mapping schemes. For example, Podobnik *et al.* [8] analyze human chromosomes 1–22 and rice chromosomes 1–12 for seven binary mapping rules and discover that the correlation patterns are different for different rules, but almost identical for all of the chromosomes, despite their varying lengths and GC contents. Lin *et al.* [9] use a slightly different sequence encoding scheme to produce smaller input vectors and report an average precision of 90.9%. Gordon *et al.* [10] extract a global signal using a matching function between sequences and develop a kernel function for a SVM, thus achieving a precision of about 85%. Ranawana and Palade [11] use a neural network-based multi classifier system with different nucleotide frequency-based encodings of the promoter sequences and report 98% precision. Duplij and Duplij [12] map DNA sequence

R. Damaševičius is with the Software Engineering Department, Kaunas University of Technology, Studentų 50-415, LT-51368, Kaunas, Lithuania (email robertas.damasevicius@ktu.lt).

nucleotides to 2D real number space based on the assumption that nucleotides have an inner abstract characteristic, called the determinative degree.

The aim of this paper is to explore the influence of various DNA sequence binary mapping rules on the quality of classification using a SVM classifier with kernel parameter optimization. The structure of the paper is as follows. Section 2 considers the SVMs, their kernel functions, and introduces a power series kernel. Section 3 discusses binary DNA sequence mapping rules. Section 4 presents a case study of promoter recognition in the drosophila and human sequence datasets using SVM classification with kernel parameter optimization. Finally, Section 5 presents conclusions and discusses future work.

II. Support Vector Machines and Their Kernels

Support Vector Machines (SVM) are one of the most popular tools in bioinformatics for supervised classification of genetic data (biosequences, protein structure data, microarray gene expression, etc.). The SVM is a binary classification algorithm based on structural risk minimization. First, the SVM implicitly maps the training data into a (usually higher-dimensional) *feature space*. A *hyperplane* (decision surface) is then constructed in this feature space that bisects the two categories and maximizes the margin of separation between itself and those points lying nearest to it (the *support vectors*). This decision surface can then be used as a basis for classifying vectors of unknown classification.

Consider an input space X with input vectors $x_i \in X$, a target space $Y = \{1,-1\}$ with $y_i \in Y$ and a training set $T = \{(x_1, y_1),...,(x_N, y_N)\}$. In the SVM classification, separation of the two classes $Y = \{1,-1\}$ is done by means of the *maximum margin* hyperplane, i.e. the hyperplane that maximizes the distance to the closest data points and guarantees the best generalization on new examples. In order to classify a new point x_j, the classification function $g(x_j)$ is used:

$$g(x_j) = \operatorname{sgn}\left(\sum_{x_i \in SV} \alpha_i y_i K(x_i, x_j) + b \right), \quad (1)$$

where SV are the support vectors, $K(x_i, x_j)$ is the kernel function, α_i are weights, and b is the offset parameter.

If $g(x_j) = +1$, x_j belongs to the *Positive class*, if $g(x_j) = -1$, x_j belongs to the *Negative class*, if $g(x_j) = 0$, x_j lies on the decision boundary and can not be classified.

The SVM is a powerful method for classification tasks. If trained optimally, it can produce excellent results. The quality of the training, however, depends on (1) the given *training data*, (2) the *mapping* of data into feature space, (3) the selection of an *optimal* kernel for the problem, which must conform with the learning target in order to obtain meaningful results, (4) the *kernel parameters* (if any), and (5) the SVM *learning* parameters, which are difficult to adjust.

The selection of the appropriate SVM kernel function is crucial for successful SVM training and accurate classification [13]. Basic SVM kernel functions are summarized in Table I. There are two common kernel optimization methods. (1) A method based on putting additional parameters in the kernel and optimizing those parameters to improve the classification precision. (2) A method based on designing new kernel functions using a combination of the available kernel functions.

TABLE I
Basic SVM Kernel Functions

Kernel	Formula
Linear	$K(x_i, x_j) = x_i^T x_j$
Polynomial	$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d$
Radial Basis Function (RBF)	$K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2)$
Sigmoid	$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

When a parameterized family of kernel functions is considered, the problem is to find an appropriate parameter vector for the given problem. The selection of the SVM parameters can be viewed as an optimization process [14, 15], and many of the standard techniques from the optimization domain may be applied directly to this problem. Optimization of the classification results requires to estimate the accuracy of the classification and to find its minimum over the parameter space.

Here we apply the second approach and propose to use a summation kernel based on the idea of power series, when the name *power series* kernel [16, 17]:

$$K_n^{pow}(x_i, x_j) = \sum_{k=1}^{n} a_k (x_i^T x_j - c)^k \quad (2)$$

It can be proven that $K_n^{pow}(x_i, x_j)$ is indeed a kernel function, because it is a summation of the linear and polynomial kernel functions. It has $n+1$ parameters available for optimization, i.e. more than basic kernel functions such as polynomial or RBF kernels have, which allows for more flexibility when optimizing the SVM classification results.

II. DNA Mapping Rules

Before applying SVM classification to the available dataset, first, one must define a mapping of classified objects to the feature space. This mapping is called a *feature vector representation* of the subject area. A feature mapping rule can be described as a function $M : \hat{S} \to F$, where $\hat{S} = (s_1, s_2, ..., s_N), s_i \in \{A, C, G, T\}^k$ is a DNA sequence, N is the length of a DNA sequence, $k = \|s_i\|$ is the length of the mapped subsequence, and $F = (f_1, f_2, ..., f_M)$, where $f_j \in \{0,1\}^l$ in case of a binary feature space, M is the length of a feature vector, $l = \|f_i\|$ is the length (dimension) of a feature. Depending upon the values of k and l, the binary mapping rules can be

classified as the binary $1\to4$, $1\to2$, $1\to1$ and $2\to1$ rules [18].

A. $1\to4$ *rule*

An example of the $1\to4$ rule is *orthogonal encoding* (see Table II), where the nucleotides in a DNA sequence are viewed as unordered categorical values and are represented by C-dimensional orthogonal binary vectors, where C is the cardinality of the 4-letter DNA alphabet [19]. This feature mapping method allows achieving better classification results than a more compact mapping scheme, where each nucleotide is mapped into $\log_2 C$-dimensional binary vectors, due to the identical Hamming distances between the nucleotide encodings [20].

B. $1\to2$ *rules*

There are several methods to represent the DNA sequence as a binary 2-bit code. Jiménez-Montaño *et al.* [21] suggested a binary interpretation of the genetic code with these correspondences: $\mathrm{A}=00, \mathrm{G}=01, \mathrm{T}=10, \mathrm{C}=11$. Stambuk [22] defined the nucleotide base representation on the square with vertices $\mathrm{T}=00, \mathrm{C}=01, \mathrm{G}=10, \mathrm{A}=11$. Karasev and Stefanov [23] suggested using the correspondence $\mathrm{C}=00, \mathrm{T}=01, \mathrm{G}=10, \mathrm{A}=11$. He *et al.* [24] used the Gray code representation of the genetic code $\mathrm{C}=00, \mathrm{T}=10, \mathrm{G}=11, \mathrm{A}=01$ to generate a sequence of the genetic code-based matrices.

Actually, there are $4!=24$ such rules, however, only 3 rules are essentially different (see Table II, Binary 1, Binary 2 and Binary 3), while the remaining rules can be obtained from these by inversion of the binary code.

C. $1\to1$ *rules*

The $1\to1$ (or *single-nucleotide*) rules are unequal representation rules that map one nucleotide into 1 and the remaining nucleotides into 0. Actually, there are four such rules (see Table 1): A-rule, C-rule, G-rule and T-rule [25]. These rules reflect the distribution of a particular type of nucleotides along the DNA sequence.

D. $2\to1$ *rules*

The $2\to1$ rules are based on the grouping of the 4-letter DNA alphabet into two subsets of two nucleotides each. There are three different such partitions, therefore there are three different binary mapping rules that map a nucleotide to a binary number (Table II). Each rule represents a different aspect of the structure of the DNA molecules.

The SW mapping rule ($\{A,T\}$vs.$\{G,C\}$) reflects the difference in the number of hydrogen bonds in the DNA molecule [26]. Each strong (S) nucleotide (C or G) has 3 hydrogen bonds, and each weak (W) nucleotide (A or T) has only 2 hydrogen bonds [27]. The composition of base pairs, or GC level, is a strand-independent property of a DNA molecule and is related to important physical-chemical properties of the DNA molecule such as the transport of electrons [28, 29] or mechanical waves [30] along the DNA helix. This rule corresponds to the fundamental partitioning of the four bases into their natural pairs, and is particularly appropriate to analyze genome-wide correlations [31].

The RY rule ($\{A,G\}$vs.$\{T,C\}$) describes how *purines* (R) and *pyrimidines* (Y) are distributed along the DNA sequence [32]. This rule corresponds to the chemical composition bias in the DNA strand.

The KM rule ($\{A,C\}$vs.$\{T,G\}$) describes how *amines* (M) and *ketones* (K) are distributed along the DNA sequence [32].

E. Summary

The list of DNA binary mapping rules presented here is by no means exhaustive. Similar rules also can be applied for dinucleotides such as CG or even longer nucleotide patterns [33]. In principle, the results obtained with each of these mapping rules are independent of each other, because they refer to the different aspects of the DNA molecule, all of them containing relevant information. The analyzed binary feature mapping rules are summarized in Table II.

TABLE II
SUMMARY OF BINARY FEATURE MAPPING RULES

Rule type	Rule name	Symbol: feature	Rule	Feature size
Binary	Orthogonal	1:4	$\langle A\to(0,0,0,1), C\to(0,0,1,0), G\to(0,1,0,0), T\to(1,0,0,0)\rangle$	$4N$
	Binary 1	1:2	$\langle A\to(0,0), C\to(0,1), G\to(1,0), T\to(1,1)\rangle$	
	Binary 2	1:2	$\langle A\to(0,0), C\to(0,1), G\to(1,1), T\to(1,0)\rangle$	$2N$
	Binary 3	1:2	$\langle A\to(0,0), C\to(1,1), G\to(0,1), T\to(1,0)\rangle$	
Single-letter	A	1:1	$\langle A\to1, B\to0\rangle, B=\{C,G,T\}$	
	C	1:1	$\langle C\to1, D\to0\rangle, D=\{A,G,T\}$	N
	G	1:1	$\langle G\to1, H\to0\rangle, H=\{A,C,T\}$	
	T	1:1	$\langle T\to1, V\to0\rangle, V=\{A,C,G\}$	
Grouping	SW	2:1	$\langle S\to1, W\to0\rangle, S=\{A,T\}, W=\{C,G\}$	
	KM	2:1	$\langle K\to1, M\to0\rangle, K=\{A,C\}, M=\{G,T\}$	N
	RY	2:1	$\langle R\to1, Y\to0\rangle, R=\{A,G\}, Y=\{C,T\}$	

IV. CASE STUDY: RECOGNITION OF PROMOTERS IN DROSOPHILA AND HUMAN SEQUENCE DATASETS

A. Datasets

For promoter classification, we use the following datasets:

1) The 2002 collection of data of drosophila (*D. melanogaster*) core promoter regions [34]. The dataset has three parts: promoter sequences, CDS (protein-coding) sequences, and non-coding (*intron*) sequences. The promoter dataset contains 1842 sequences from -250 bp to +50 bp with regards to the gene transcription site location. The intron dataset contains 1799 sequences, each 300 bp length. The CDS dataset contains 2859 sequences, each 300 bp length. The training file contains 1260 examples (372 promoters, 361 introns, 527 CDS). The test file contains 6500 examples (1842 promoters, 1799 introns, 2859 CDS).

2) The collection of data of human and additional eukaryotic (vertebrate) promoter regions. The promoters were extracted from the Eukaryotic Promoter Database rel. 50; the negative set contains coding and non-coding sequences from the 1998 GENIE data set [35]. The dataset also has three parts: promoter, CDS, and non-coding sequences. The promoter dataset contains 565 sequences from -250 bp to +50 bp with regards to the gene transcription site location. The intron dataset contains 4345 sequences, each 300 bp length. The CDS dataset contains 890 sequences, each 300 bp length. The training file contains 1386 examples (339 promoters, 869 introns, 178 CDS). The test file contains 5800 examples (565 promoters, 4345 introns, 890 CDS).

The difficulty with this dataset is that promoter sequences have both introns and CDS parts (from +1 bp to +50 bp), thus classification of promoters is much more difficult than pure introns or protein-coding sequences. The promoters are often assigned to introns or CDS. Furthermore, the datasets are imbalanced: there are 29.5% promoters against 70.5% non-promoters in the drosophila training dataset, and 28.3% promoters against 71.7% non-promoters in the drosophila test dataset. In the human dataset, there are 24.4% promoters against 75.6% non-promoters in the training dataset, and 9.7% promoters against 90.3% non-promoters in the test dataset. Therefore, the classification of the promoters is not a trivial task.

B. Classification metrics

Our aim is to classify DNA sequences as promoters and non-promoters. Therefore, here we have a binary classification problem in which the outcomes are labeled either as positive (*P*) or negative (*N*) class. There are four possible outcomes from a binary classifier. If the outcome from a prediction is *P* and the actual value is also *P*, then we have a *true positive* (*TP*); however if the actual value is *N* then we have a *false positive* (*FP*). Conversely, a *true negative* (*TN*) has occurred when both the prediction outcome and the actual value are *N*, and *false negative* (FN) is when the prediction outcome is *N* while the actual value is *P*. To evaluate the precision of classification the following metrics will be used:

1) *Specificity* (SPC) is a measure of how well a binary classification test correctly identifies the negative cases.

$$SPC = \frac{n_{TN}}{n_{FP} + n_{TN}} \cdot 100\% \quad (3)$$

here n_i is the number of i cases in the classification results.

2) *Sensitivity* (or *True Positive Rate*, TPR) is a measure of how well a binary classification test correctly identifies the positive cases.

$$TPR = \frac{n_{TP}}{n_{TP} + n_{FN}} \cdot 100\% \quad (4)$$

3) To evaluate the accuracy of classification, the *F-measure* (F) is used, which is the harmonic mean of specificity and sensitivity:

$$F = \frac{2 \cdot SPC \cdot TPR}{SPC + TPR} \quad (5)$$

The best possible classification method would yield 100% sensitivity (all true positives are found), 100% specificity (no false positives are found), and 100% F-measure. All three metrics can be used as objective functions for optimization. However, since we are particularly interested in the correct recognition of the positive cases, in this paper we use the sensitivity metric as the objective function for the optimization of the kernel function parameters.

C. SVM and its parameter optimization method

As a classifier, we use SVM^{light}, which is an implementation of the SVM in C [36, 37].

For optimization of the SVM kernel parameters, we selected the Nelder-Mead (*downhill simplex*) method [38]. The method constructs a simplex of $M+1$ vertices in M-dimensional optimization problem. The method approximately finds a locally optimal solution to a problem with M variables, when the objective function varies smoothly. A new simplex point is generated by extrapolating the behavior of the objective function measured at each test point arranged as a simplex. The algorithm then replaces the worst point with a point reflected through the centroid of the remaining M points. If this trial point is best, the new simplex is expanded further out. If the function value is worse, than the second worst point of the simplex is contracted. If there is no improvement, the simplex is shrunken towards the best point. This procedure terminates if the differences in the function values between the best and worst points are negligible. The initial simplex is also important, because a too small initial simplex can lead to a local search.

D. Evaluation of results

The classification results for the different mapping rules are presented in Table III (for the drosophila dataset) and Table IV (for the human dataset).

The best classification results for the drosophila dataset are obtained using Binary 1, KM, A and T rules, and for the human dataset – using Binary 1, Binary 2, Binary 3, SW, C and G rules (the specificity and sensitivity values are close to 100% indicating very good classification precision). This can be explained by the fact that drosophila promoter sequences are characterized by the repeating occurrences of the TATA box (TATAA or TATAAA subsequences) or

the Pribnow box (TATAAT subsequence), thus the best results can be achieved using the rules, where *A* and *T* nucleotides are coded using different binary values (as, e.g., in the KM rule).

Human promoter sequences are also characterized by the occurrence of the DPE (*downstream promoter element*), which is a distinct 7-nucleotide sequence (A/G)G(A/T)CGTG [39], thus the best classification results are obtained, when *C* and *G* nucleotides are coded using different binary values (as, e.g., in the SW rule). Interestingly, the Binary 1 mapping rule seems to work best for both datasets, because it also separates A/T and C/G nucleotides explicitly by inverted binary code values. Note that the orthogonal mapping rule has failed for the human dataset, perhaps, due to the feature space explosion, which caused the failure of the SVM classifier.

TABLE III
DROSOPHILA SEQUENCE CLASSIFICATION RESULTS

Mapping rule	Classification metric		
	Specificity (SPC)	Sensitivity (TPR)	F-measure (F)
Orthogonal	95.12	90.83	92.93
Binary 1	98.39	99.67	99.03
Binary 2	90.43	77.04	83.20
Binary 3	94.58	89.14	91.78
SW	90.61	78.56	84.16
KM	96.44	99.51	97.95
RY	88.26	81.60	84.80
A	99.63	99.78	99.70
C	88.94	78.45	83.37
G	88.41	72.80	79.85
T	99.75	99.89	99.82

TABLE IV
HUMAN SEQUENCE CLASSIFICATION RESULTS

Mapping rule	Classification metric		
	Specificity (SPC)	Sensitivity (TPR)	F-measure (F)
Orthogonal	90.26	0.00	0.00
Binary 1	99.75	99.89	99.82
Binary 2	99.61	99.29	99.45
Binary 3	99.52	99.78	99.65
SW	99.52	99.78	99.65
KM	96.81	75.40	84.77
RY	98.26	90.80	94.38
A	98.60	91.50	94.92
C	99.52	99.89	99.70
G	99.73	99.29	99.51
T	96.74	73.98	83.84

To represent the accuracy of classification graphically, the *Receiver Operating Characteristic* (ROC) [40] is used. ROC is a graphical plot of sensitivity *vs.* (100 - specificity). The ROC plot allows to select possibly optimal models and to discard suboptimal ones independently from the class distribution. The perfect classification corresponds to the (0,100) point in the ROC plot.

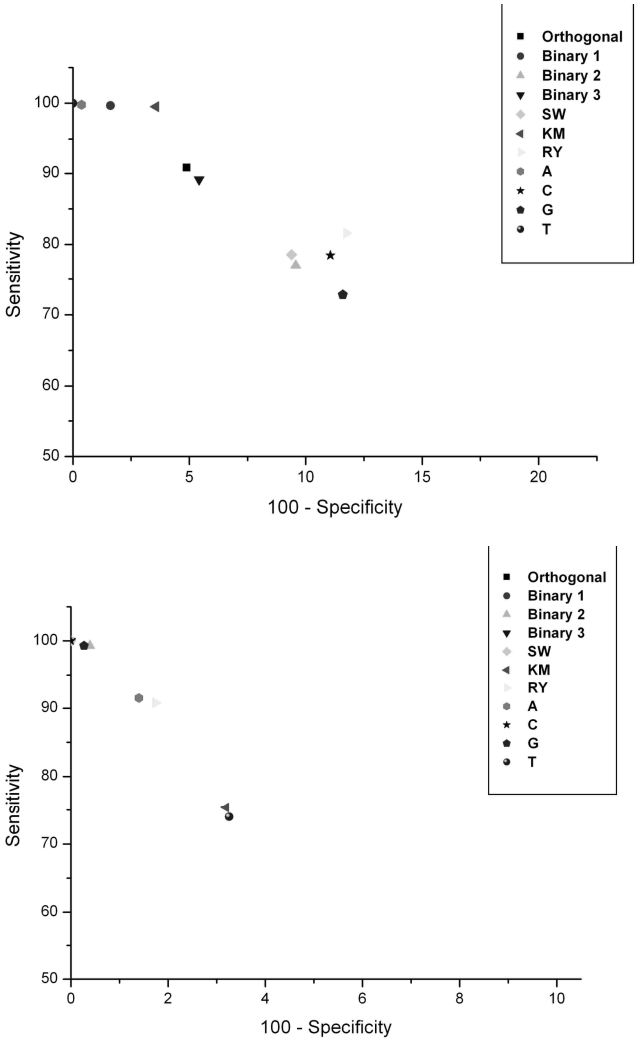

Fig. 1. Comparison of SVM classification results for different sequence mapping methods: drosophila dataset (top), and human dataset (bottom).

V. CONCLUSIONS AND FUTURE WORK

The quality of the classification depends on many different choices such as (1) the mapping of data into a feature space, (2) the selection of an optimal kernel for the problem, (3) SVM kernel parameters (if any), and (4) SVM learning parameters. These classification parameters are usually selected by the SVM user *ad hoc*.

In this paper, we have examined the influence of the selection of binary feature mapping rules on the quality of classification in the promoter recognition problem. The experimental results show that the selection of the appropriate rule can greatly influence the classification results. The mapping rule should be selected based on the properties of the available data for a specific classification problem. The obtained classification results confirm this conclusion: the mapping rule(s) with the best classification results correspond to the characteristics of the repeating subsequences ("boxes") of the promoter sequences.

Future work will focus on the analysis of the real number feature mapping rules such as frequency or entropy-based rules for promoter recognition as well as for other classification problems in the bioinformatics domain such as splice-site junction recognition.

REFERENCES

[1] T. Sobha Rani, S. Durga Bhavani and R.S. Bapi, "Analysis of E.coli promoter recognition problem in dinucleotide feature space," *Bioinformatics,* vol. 23(5), pp. 582-588, 2007.

[2] T. Werner, "The state of the art of mammalian promoter recognition," *Briefings in Bioinformatics*, vol. 4(1), pp. 22-30, 2003.

[3] V. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.

[4] N. Cristianini, and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[5] C. M. van der Walt and E. Barnard, "Data characteristics that determine classifier performance," *Proc. of the 17th Annual Symp. of the Pattern Recognition Association of South Africa*, pp. 166-171, 2006.

[6] Y. Zhao, and Q. He, "An unbalanced dataset classification approach based on v-Support Vector Machine," *Proc. of the Sixth World Congress on Intelligent Control and Automation, WCICA 2006,* pp. 10496-10501, 2006.

[7] G. Wu, and E. Y. Chang, "Class-boundary alignment for imbalanced dataset learning," *ICML'2003 Workshop on Learning from Imbalanced Data Sets* (II), August 21, 2003, Washington, DC.

[8] B. Podobnik, J. Shao, N. V. Dokholyan, V. Zlatic, H. E. Stanley, and I. Grosse, "Similarity and dissimilarity in correlations of genomic DNA," *Physica A*, 373:497-502, 2006.

[9] C. J. Lin, "Prediction of RNA polymerase binding sites using purine-pyrimidine encoding and hybrid learning methods," *Int. J. Appl. Sci. Eng.,* Vol. 2, 177–188, 2004.

[10] L. Gordon, A. Chervonenkis, A.J. Gammerman, I.A. Shahmuradov, and V. V. Solovyev, "Sequence alignment kernel for recognition of promoter regions", *Bioinformatics,* vol. 19, pp. 1964–1971, 2003.

[11] R. Ranawana, and V. Palade, "A neural network based multiclassifier system for gene identification in DNA sequences," *J. Neural Comput. Appl.,* Vol. 14, pp. 122–131, 2005

[12] D. Duplij, and S. Duplij, "DNA sequence representation by trianders and determinative degree of nucleotides," *J. Zhejiang Univ. Sci. B.,* vol. 6(8), pp. 743–755, 2005.

[13] V. Cherkassky, and F. Mulier, *Learning from Data: Concepts, Theory, and Methods*. John Wiley & Sons, 1998.

[14] L. Zhuang, and H. Dai, "Parameter optimization of kernel-based one-class classifier on imbalance learning," *Journal of Computers,* vol. 1(7), pp. 32-40, 2006.

[15] T. Eitrich, and B. Lang, "Efficient optimization of Support Vector Machine learning parameters for unbalanced data sets," *Journal of Computational and Applied Mathematics*, vol. 196(2), pp. 425-436, 2006.

[16] R. Damaševičius, "Optimization of SVM parameters for promoter recognition in DNA sequences," *Proc. of Int. Conf on Continuous Optimization and Knowledge-Based Technologies (EurOPT-2008),* Neringa, Lithuania, May 20-23, 2008.

[17] R. Damaševičius, "Splice site recognition in DNA sequences using k-mer frequency based mapping for Support Vector Machine with power series kernel," *Proc. of Int. Conf. on Complex Software Intensive Systems (CISIS-2008),* Barcelona, Spain, March 4-7, 2008, pp. 687-692.

[18] R. Damaševičius, "Feature representation of DNA sequences for machine learning tasks," *Proc. of Fifth Int. Workshop on Computational Systems Biology (WCSB 2008),* Leipzig, Germany, June 11-13, 2008.

[19] S. Brunak, J. Engelbrecht, and S. Knudsen, "Prediction of human mRNA donor and acceptor sites from the DNA sequence," *J. Mol. Biol*., vol. 220, pp. 49–65, 1991.

[20] B. Demeler, and G. W. Zhou, "Neural network optimization for E. coli promoter prediction", *Nucleic Acids Res*, vol. 19, pp. 1593–1599, 1991.

[21] M.A. Jimenez-Montano, C.R. de la Mora-Basanez, and T. Poschel, "The hypercube structure of the genetic code explains conservative and non-conservative amino acid substitutions in vivo and in vitro," *Biosystems*, vol. 39, pp. 117-125, 1996.

[22] N. Stambuk, "Universal metric properties of the genetic code", *Croatica Chemica Acta,* vol. 73, pp. 1123-1139, 2000.

[23] V.A. Karasev and V. E. Stefanov, "Topological nature of the genetic code," *J. Theor. Biol.,* vol. 209, pp. 303-317, 2001.

[24] M. He, S. Petoukhov, and P. E. Ricci, "Genetic code, Hamming distance and stochastic matrices," *Bull. Math. Biol.*, 00, pp. 1–17, 2004.

[25] P. Bernaola-Galvan, R. Roman-Roldan, and J. L. Oliver, "Compositional segmentation and long-range fractal correlations in DNA sequences", *Phys. Rev. E,* vol. 53, pp. 5181–5189, 1996.

[26] M. Y. Azbel, Y. Kantor, L. Verkh, and A. Vilenkin, "Statistical analysis of DNA sequences," *Biopolymers*, vol. 21, pp. 1687-1690, 1982.

[27] B. Lewin, *Genes*. New York: Wiley and Sons, 1983.

[28] P. J. Dandliker, R. E. Holmlin, and J. K. Barton, "Oxidative thymine dimmer repair in the DNA helix," *Science*, vol. 275, pp. 1465–1468, 1997.

[29] P. Carpena, P. Bernaola-Galvan, P. Ivanov, and H. E. Stanley, "Metal–insulator transition in one-dimensional solids with correlated disorder," *Nature*, vol. 418, pp. 955–959, 2002.

[30] M. Rief, H. Clausen-Schaumann, and H. E. Gaub, "Sequence-dependent mechanics of single DNA molecules," *Nat. Struct. Biol.*, vol. 6, pp. 346–349, 1999.

[31] A. Arneodo, E. Bacry, P. V. Graves, and J. F. Muzy, "Characterizing long-range correlations in DNA sequences from wavelets analysis," *Phys. Rev. Lett.*, vol. 74, pp. 3293–3296, 1995.

[32] H.E. Stanley, S.V. Buldyrev, A.L. Goldberger, S. Havlin, C.-K. Peng, and M. Simons, "Scaling features of noncoding DNA," *Physica A*, vol. 273, pp. 1-18, 1999.

[33] S. Karlin, and V. Brendel, "Patchiness and correlations in DNA sequences," *Science,* vol. 259, pp. 677-680, 1993.

[34] Berkeley Drosophila Genome Project. Drosophila promoter dataset. http://www.fruitfly.org/seq_tools/datasets/Drosophila/promoter/

[35] Berkeley Drosophila Genome Project. Human promoter dataset. http://www.fruitfly.org/seq_tools/datasets/Human/promoter/

[36] T. Joachims, *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers, 2003.

[37] SVMlight. http://svmlight.joachims.org/

[38] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7 (4), pp. 308-313, 1965.

[39] C. Yang, E. Bolotin, T. Jiang, F. M. Sladek, and E. Martinez, "Prevalence of the initiator over the TATA box in human and yeast genes and identification of DNA motifs enriched in human TATA-less core promoters," *Gene*, vol. 389(1), pp. 52-65, 2007.

[40] M. S. Pepe, *The Statistical Evaluation of Medical Tests for Classification and Prediction*. New York: Oxford, 2003.