ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
"Antonio Ruberti"
CONSIGLIO NAZIONALE DELLE RICERCHE

P. Bertolazzi, G. Felici, M. Guarracino

LOGIC FORMULAS BASED KNOWLEDGE DISCOVERY AND ITS APPLICATION TO THE CLASSIFICATION OF BIOLOGICAL DATA

R. 08-06, 2008

Paola Bertolazzi – Istituto di Analisi dei Sistemi ed Informatica del CNR, viale Manzoni 30 - 00185 Roma, Italy. Email: bertola@iasi.rm.cnr.it.

Giovanni Felici – Istituto di Analisi dei Sistemi ed Informatica del CNR, viale Manzoni 30 - 00185 Roma, Italy. Email: felici@iasi.rm.cnr.it.

Mario Guarracino – High Performance Computing and Networking Institute National Research Council, Naples, Italy.E-mail: mario.guarracino@cnr.it.

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.cnr.it
URL: http://www.iasi.cnr.it

Abstract

Classifiers built through supervised learning techniques are widely used in computational biology. Examples are neural networks, decision trees and support vector machines. Recently, an extension of Regularized Generalized Eigenvalues Classifier (ReGEC) has been proposed, in which prior knowledge is included. When knowledge is formalized as a set of linear constraints to the ReGEC, the resulting non linear classifier has a lower complexity and halves the misclassification error with respect to the original method. In this work, we show how logic programming can extract knowledge from data to enhance classification models produced by ReGEC. The knowledge extraction method is based on two phases: a feature selection phase and a rules extraction phase. Feature selection is formulated as an integer programming problem that extends a set covering problem. The extraction phase is performed through the iterative solution of different instances of the same minimum cost satisfiability problem that models the logic separation rules used for classification. The overall method, that we call LF-ReGEC, guarantees that the number of points in the training set is not increased and the resulting model does not overfit the problem. Furthermore, the overall accuracy of the method is increased. Finally, the method is compared with other methods using genomic and proteomic data sets taken from the literature.

1. Introduction

The use of automatic classification methods based on supervised learning has become an important research topic for theoretical and applied mathematics. The capability of extracting information and knowledge from large amount of data is in fact a much demanded task for all those settings where complex phenomena are observed, and numerically measured, with the aim of understanding the underlying motivations that govern them.

The literature proposes different types of classifiers, each one characterized by its own specific features. Their aim is to fit a model on the observed data that is capable of connecting the observed measures of a data point with a relevant, non-observed characteristic of the data point itself, usually referred to as *class*. The standard approach of supervised learning is thus to infer the model's parameter from a set of observations of known class (the training sample) and then apply the model to predict or forecast the class value for observations for which the class is unknown. In the typical setting, one keeps aside a portion of the available data - referred to as test sample - to verify that the model, produced on the basis of the training data, obtains a sufficient level of precision in predicting the class for the testing data. More elaborated testing schemes have been developed to reinforce the model's evaluation, such as cross validation or leave-one-out testing.

Widely known examples of classifiers are neural networks[3], decision trees[18] and support vector machines (SVM)[5]. Classification methods have been successfully applied in a variety of fields. In particular, promising applications of these methods are in the field of biomedicine and bioinformatics. Here, the precision of the method is of particular relevance, and the data sets that have to be analyzed are typically very large; the available samples are composed of genetic expressions or DNA sequences that can reach dimensions of several tens of thousands.

The relevance of automatic classification for problems of this type pushes the efforts of research towards the identification of new ideas that are able to improve on the current performances of the available methods. One such idea is related with the integration of external or prior knowledge in a classification method. A natural approach is to plug a priori knowledge in a classifier adding directly more points to the data set. This results in higher computational complexity, and in a propension to overfitting; moreover, most times additional data is not available or expensive to obtain.

An interesting approach in this direction has been proposed by Mangasarian[16], showing that it is possible to analytically express knowledge as additional terms of the cost function of the optimization problem defining a standard SVM. This solution has the advantage not to increase the dimension of the training set, thus to avoid overfitting and poor generalization of the classification model[3].

Guarracino et al.[15] have recently shown how to extend this approach to Generalized Proximal Support Vector Machines[14] (GEPSVM), halving the missclassification error of the original method. The idea of increasing the knowledge contained in the training set with additional knowledge is particularly appealing in the context of biomedical data, where it can provided by the experience of field experts or previous results.

In this paper, we propose an alternative method to incorporate additional knowledge in a SVM classifier, extending the work of Mangasarian et al.[17]. The proposed approach is based on the extraction of additional knowledge from the training data itself, but with a learning method that is consistently different from SVM. We adopt the logic mining method *Lsquare*[7] combined with a recently developed feature extractor based on integer programming, to extract logic DNF formulas from the training data. Then, we select the most meaningful portions of such formulas and plug them into the ReGEC algorithm as external non linear knowledge.

The results so obtained are indeed very interesting and exhibit quite consistently an increase in the recognition capability of the system, as measured by 10-fold cross validation. In some sense, we propose a combination of two very different learning methods: ReGEC, that operates in a multidimensional Euclidean space with highly nonlinear data transformation, and logic learning, that operates in a discretized space with models based on propositional logic. The former constitutes the master learning algorithm, while the latter provides the additional knowledge that is efficiently incorporated and dealt with by the ReGEC algorithm.

It is of interest to note that the logic rules adopted to represent the external knowledge are the most

appropriate form for synthesizing the additional knowledge that is possessed by field experts.

We briefly introduce the notation adopted throughout the paper below.

- Given a vector $x \in^n$, x_1 is the 1-norm, $(\sum_{i=1}^{n} |x_i|)$, and x is the 2-norm, $\left(\sum_{i=1}^{n} (x_i)^2\right)^{\frac{1}{2}}$.
- A^T is the transpose of the matrix $A \in^{m\times n}$, and A_i and $A_{.j}$ are the i-th row and the j-th column, respectively, of the matrix A.
- $e = (1, 1, \ldots, 1)^T$ and $0 = (0, 0, \ldots, 0)^T$.
- Given two matrices $A \in^{m\times n}$ and $B \in^{n\times k}$, a *kernel* $K(A, B)$ maps $^{m\times n}\times^{n\times k}$ into $^{m\times k}$. One of the most widely used kernels is the Gaussian kernel where the ij-th element of the matrix is defined as $(K(A, B))_{ij} = e^{-\mu {A_i^T - B_{.j}}^2}$, where A and B are matrices with the same number of columns, and μ is a positive constant.
- In classification, each point $x \in^n$ is assigned to one of the classes in $\{-1, 1\}$, and, for set Γ of m real points, which is represented by the m rows of the matrix in $^{m\times n}$, there is an associated a vector $c \in \{-1, 1\}^m$ of labels denoting their classes.

The paper is organized as follows. In Section 2, we give a general description of knowledge based SVM. Section 3 discusses the technique by means of which additional knowledge can be implicitly added to the ReGEC classifier with very limited additional computational cost. Section 4 describes the logic learning method adopted to extract separating logic rules from the data while Section 5 presents the experimental results obtained on 5 different biomedical data sets of large dimension. Section 6 provides some conclusions and directions for future research.

2. Knowledge based SVM

Support Vector Machine is a state of the art machine learning algorithm[5, 27]. The main idea of SVM is to separate the input space in two half spaces using the hyperplane $x^T w - \gamma = 0$ which maximizes the margin between the two classes. The hyperplane can be found by minimizing the norm of w, with constraints to correctly classified points of both classes.

For nonlinear classification, the SVM is used with kernel functions [17] and the basic solution technique is still through linear programming. With a kernel Radial Basis Function, the corresponding hyperlane, projected in the feature space [22], has the following form:

$$f(x) \equiv K(x^T, B^T)u - \gamma \ = 0, \tag{1}$$

where $B \in^{k\times n}$ and $K(x^T, B^T) :^{1\times n} \times^{n\times k} \rightarrow^{1\times k}$ is the Radial Basis Function that returns the vector y with components:

$$y_i = e^{-\frac{\|x - B_i\|^2}{\sigma}}.$$

Parameters $u \in^k$ and $\gamma \in$ are determined by solving the following linear programming problem[17]:

$$\begin{aligned} \min_{u,\gamma,y,s} \quad & \nu e^T y + e^T s \\ \text{s.t.} \quad & D(K(\Gamma, \Gamma^T)u - \gamma e) + y \geq e, \\ & -s \leq u \leq s, \\ & y \geq 0. \end{aligned} \tag{2}$$

Here, D is the diagonal matrix of ± 1 corresponding to elements of the training set or matrix Γ.

One can include the knowledge of an expert in the learning phase of the function (1) to improve the results obtained by a classifier from the training set. The expertise knowledge is represented by the

following implication which represents a knowledge region Δ in the input space in which points xs are known to belong to the class $+1$:

$$g(x) \leq 0 \Rightarrow K(x^T, B^T)u - \gamma \geq \alpha, \forall x \in \Delta, \tag{3}$$

where α is a nonnegative number. Therefore, based on the theorem of the alternatives[17] for a convex function, the implication (3) can be expressed as a linear inequality in terms of the parameters (u, γ), and we can add positive nonlinear knowledge to the problem (2) as follows:

$$\begin{aligned} \min_{u,\gamma,y} \quad & \nu e^T y + e^T s \\ \text{s.t.} \quad & D(K(A, B^T)u - \gamma e) + y \geq e, \\ & -s \leq u \leq s,\ y \geq 0, \\ & K(x_i^T, B^T)u - \gamma - \alpha + v^T g(x_i) + z_i \geq 0, \\ & v \geq 0,\ z_i \geq 0, \quad i = 1, \ldots, l. \end{aligned} \tag{4}$$

The same holds for negative knowledge regions. This leads to the linear program (2) with knowledge included in the cost function:

$$\begin{aligned} \min_{u,\gamma,y,s,v,p,z_1,\ldots,z_l,q_1,\ldots,q_t} \quad & \nu e^T y + e^T s + \sigma(\sum_{i=1}^{l} z_i + \sum_{j=1}^{t} q_j) \\ \text{s.t.} \quad & D(K(A, B^T)u - \gamma e) + y \geq e, \\ & -s \leq u \leq s,\ y \geq 0, \\ & K(x_i^T, B^T)u - \gamma - \alpha + v^T g(x_i) + z_i \geq 0, \\ & v \geq 0,\ z_i \geq 0,\ i = 1, \ldots, l, \\ & -K(x_j^T, B^T)u + \gamma - \alpha + p^T g(x_j) + q_j \geq 0, \\ & p \geq 0,\ q_j \geq 0,\ j = 1, \ldots, t. \end{aligned} \tag{5}$$

We note that the linear programming problem (5) minimizes the margin between the two classes constraining the classification problem to leave the two a priori knowledge sets in the corresponding halfspace.

3. Nonlinear Knowledge in GEPSVM

We recall that a SVM binary classifier finds a hyperplane which divides the space into two halfspaces. Points laying in one halfspace belong to class $+1$, the others to class -1. A different approach is also used in Proximal Support Vector Machines (PSVM) [10], where the class of a point is determined by the closeness to one of two hyperplanes.

Given matrices $A \in^{m \times n}$ and $B \in^{k \times n}$, respectively representing points of class $+1$ and -1, $\Gamma = A \cup B$, we can find the hyperplane approximating the class $+1$ solving the following minimization task:

$$\min_{u,\gamma \neq 0} \frac{K(A, \Gamma)u - e\gamma^2}{K(B, \Gamma)u - e\gamma^2}, \tag{6}$$

which finds the hyperplane minimizing the distance from the points of class $+1$ and at the same time maximizing the distance from the points of class -1. Conversely, the hyperplane for points in class -1, can be found solving the reciprocal of (6):

$$\min_{u,\gamma \neq 0} \frac{K(B, \Gamma)u - e\gamma^2}{K(A, \Gamma)u - e\gamma^2}. \tag{7}$$

Equation (7) finds the hyperplane that minimizes the distance from points in class -1 and maximizes the distance from points in class -1. Each of these problems can be solved using regularization as

proposed by Guarracino et. al.[14]:

$$\min_{u,\gamma\neq 0} \frac{K(A,\Gamma)u - e\gamma^2 + \delta\widetilde{K}_B u - e\gamma^2}{K(B,\Gamma)u - e\gamma^2}, \tag{8}$$

$$\min_{u,\gamma\neq 0} \frac{K(B,\Gamma)u - e\gamma^2 + \delta\widetilde{K}_A u - e\gamma^2}{K(A,\Gamma)u - e\gamma^2}, \tag{9}$$

where $\widetilde{K}_A$ and $\widetilde{K}_B$ are diagonal matrices, with diagonal elements taken respectively from matrices $K(A,\Gamma)$ and $K(B,\Gamma)$.

Given

$$\begin{aligned} G &= [K(A,\Gamma) \ -e]^T [K(A,\Gamma) \ -e], \\ H &= [K(B,\Gamma) \ -e]^T [K(B,\Gamma) \ -e], \\ z &= \begin{bmatrix} u^T & \gamma \end{bmatrix}^T, \end{aligned} \tag{10}$$

the equation (6) becomes:

$$\min_{z\in^m} \frac{z^T G z}{z^T H z}. \tag{11}$$

Similarly for B, we obtain the reciprocal problem:

$$\min_{z\in^m} \frac{z^T H z}{z^T G z}. \tag{12}$$

Equations (11) and (12) represent Rayleigh quotients of the eigenvalue problems $Gz = \lambda Hz$ and their reciprocal.

The minimum eigenvectors obtained as solution to (8)-(9) give the proximity planes $P_i, 1 = 1, 2$. A given point x will thus be classified according to the following formula:

$$class(x) = \arg\min_{i=1,2} dist(x, P_i) \tag{13}$$

using the distance

$$dist(x, P_i) = \frac{K(x,\Gamma)u - \gamma}{u}. \tag{14}$$

GEPSVM algorithm has several advantages with respect to SVM. First of all, in its linear formulation, it can be used to classify problems that are not linearly separable. Furthermore, its computational complexity is dominated by the number of training samples. Finally, its implementatino is reduced to eigenpairs computation, which can be expressed in a single line code in many problem solving environments such as R and Matlab.

It is possible to add nonlinear prior knowledge formulating the model in terms of a constrained generalized eigenvalue problems. The latter has been extensively studied and a procedure for its solution has been proposed by Golub[12].

If G and H, as defined in (11), are symmetric matrices of order n, constraints can be expressed by the equation:

$$C^T z = 0, \tag{15}$$

where C is an $n \times p$ matrix of rank r, with $r < p < n$. The constrained eigenvalue problem for the classification surface for points in class +1 is:

$$\begin{aligned} \min_{z\in^m} \quad & \frac{z^T G z}{z^T H z} \\ \text{s.t.} \quad & C^T z = 0. \end{aligned} \tag{16}$$

Let Δ be the set of class +1 points describing nonlinear knowledge, constraint matrix C must represent knowledge imposed on class +1 points, hence it will be:

$$C = \begin{bmatrix} K(\Delta, \Gamma) & -e \end{bmatrix}^T \tag{17}$$

Matrix C needs to be rank deficient in order to have non-trivial solution. The set of constraints 15 requires all points in Δ to have null distance from the hyperplane, and thus to belong to class +1.

The QR decomposition of C gives two matrices Q and R such that $C = QR$. Q is an orthonormal matrix where $Q^TQ = I$. R is an order r upper triangular matrix. Reordering the rows of C, we can write:

$$Q^TC = \begin{bmatrix} R & S \\ 0 & 0 \end{bmatrix},$$

where S is an $r \times (p - r)$ matrix. Let

$$z = Qw = Q \begin{bmatrix} y \\ v \end{bmatrix},$$

where y is a vector of the first r components of w and v of the last $(n - r)$ components of w, thus having a representation of z in the space generated by Q. We have:

$$C^Tz = \begin{bmatrix} R^T & 0 \\ S^T & 0 \end{bmatrix} \begin{bmatrix} y \\ v \end{bmatrix} = 0$$

and hence $y = 0$. Defining $z = Qw$ it is possible to reformulate the equation (11) as:

$$\min_{w \neq 0} \frac{w^T\ Q^T\ G\ Q\ w}{w^T\ Q^T\ H\ Q\ w}.$$

To simplify, we let $L = Q^TGQ$ and $M = Q^THQ$, with

$$L = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix}, M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},$$

where L_{11}, M_{11} are $r \times r$ matrices and L_{22}, M_{22} are $(n - r) \times (n - r)$ matrices. Both L and M are symmetric matrices. Moreover, being matrix M positive definite, we have

$$0 < \lambda_{min}(M) \leq \lambda_{min}(M_{22}) \leq \lambda_{max}(H_{22}) \leq \lambda_{max}(H), \tag{18}$$

where λ_{min} and λ_{max} represent minimum and maximum eigenvalues. Leading back to a minimization problem, we have to find w such that:

$$\min_{w \neq 0} \frac{w^TLw}{w^TMw}. \tag{19}$$

Minimization problem (19) contains positive nonlinear knowledge represented by C. This expression is Rayleigh quotient of the generalized eigenvalue problem $Lw = \lambda Mw$. Stationary points are those and only those corresponding to the eigenvectors of (19). Moreover, being M positive definite, Rayleigh quotient is limited and varies in the interval determined by minimum and maximum eigenvalues [19]. Considering (18), we just need to search stationary values of the equation:

$$L_{22}\ v = \lambda M_{22} v. \tag{20}$$

Since L and M are symmetric and M is positive definite, L_{22} and M_{22} will be symmetric, and M_{22} positive definite.

So far, having found the $n - r$ eigenvalues and eigenvectors of $L_{22}v_i = \lambda_i M_{22}v_i$, $i = 1 \ldots n - r$, we calculate the components of the vector z, original solution of the problem (16):

$$w_i = Q \begin{bmatrix} 0 \\ \cdots \\ I_{n-r} \end{bmatrix} v_i. \tag{21}$$

The constrained method just introduced has a lower complexity, compared to the original method, in the solution of the eigenvalue problem (20), which involves matrices of order $(n - r)$, although an initial QR factorization is needed for the matrix C.

4. Additional Knowledge in the form of Logic Formulas

The additional knowledge for the ReGEC classifier is extracted from training data with a logic mining technique capable of dealing efficiently with large data sets. Such choice is motivated by two main considerations: first, the nature of the method is intrinsically different from the SVM adopted as primary classifier; second, the logic formulas are, semantically, the form of "knowledge" closest to human reasoning and therefore resemble at best contextual information.

The logic mining system consists of two main components, each characterized by the use of integer programming models and state-of-the-art solution techniques. Below, a brief description of the system is given, pointing, when needed, to extended descriptions made available in the related literature.

Discretization. When the data is in numeric form (e.g., as in the case of the experiments described in the following of this paper) the system adopts a discretization method that builds, from each original variable, one or more logic variables based on some thresholds on the range of variation of the variable itself. This is accomplished by an iterative procedure that first sections the variation interval of the variable in a large number of intervals, and the joins these intervals based on class entropy. A detailed description of the methods can be found in [2].

Feature Selection. The logic variables obtained in the discretization step are selected with a feature selection method inspired to the one often referred to as *Combinatorial Feature Selection* or *Minimal Test Collection* (see [11]). We proposed a modification of such method based on the infimum-norm that amounts to the following integer linear program:

$$\begin{aligned} \max \quad & \alpha \\ \text{s.t.} \quad & \sum_{h=1}^{m} x_h \leq k \\ & \sum_{h=1}^{m} a_{ij}^h x_h \geq \alpha, \quad i = 1 \dots n,\ j = 1 \dots n,\ c(i) \neq c(j) \\ & x_h \in \{0,1\}, \quad h = 1 \dots m, \end{aligned} \tag{22}$$

where the binary variables $x_h = \{0, 1\}$ are associated with each feature ($h = 1, ..., m$) and have value of 1 only if the feature is chosen; the coefficients a_{ij}^h equal to 1 when individuals i and j differ on feature h, and 0 otherwise, and $c(i)$ indicates the class of individual i. For details, refer to [2], [6]. Such combinatorial problem is hard in nature, and, for interesting cases, of large dimensions. For this reason we adopt a randomized heuristic solution technique of the GRASP[1, 9] family, which stand for Greedy Randomized Adaptive Search Procedures, that has proven to reach solutions of very good quality in limited time.

Logic Separation. The extraction of logic formulas that holds true for individuals in one class and false for those in the other one is accomplished by the logic miner *Lsquare*. *Lsquare* is a learning method that operates on data represented by logic variables and extracts separating logic formulas in Disjunctive Normal From (DNF). The classification rules are determined using a particular problem formulation that amounts to be a well know and hard combinatorial optimization problem, the *minimum cost satisfiability problem*, or MINSAT, that is solved using a very sophisticated solver based on decomposition and learning techniques [26]. The DNF formulas are formed by few clauses with large coverage (the interpretation of the trends present in the data) and, if needed, additional clauses with smaller coverage (the interpretation of the outliers in the training set). The system and its additional components have been presented and described in related papers ([7, 8]) and their detailed description is out of the scope of this paper.

5. Experimental results

LF-ReGEC has been implemented with Matlab 7.3.0. The computational kernels of the logic formulas are implemented in C. Results are evaluated using an Intel Xeon CPU 3.20GHz, 6GB RAM running Red Hat Enterprise Linux WS release 3. Matlab function *eig* for the solution of the generalized eigenvalue

problem was used as computational kernel of ReGEC. Tests have been performed for ReGEC, LF and LF-ReGEC algorithms. Accuracy results for SVM and TSP are taken from literature[25].

5.1. A case study

We have chosen acute leukemias microarray dataset as a test case. The aim is to classify acute leukemias into those arising from lymphoid precursors (acute lymphoblastic leukemia, ALL) or from myeloid precursors (acute myeloid leukemia, AML). Distinguishing ALL from AML is critical for successful treatment. Indeed, chemotherapy regimens for ALL are generally different from those for AML and, although remissions can be achieved using ALL therapy for AML (and vice versa), cure rates are markedly diminished, and toxicities are encountered. The dataset is obtained from Golub et. al., 1999[13]. It consists of 25 AML and 47 ALL samples. The gene expression data for 7129 probes has been acquired with an Affymetrix microarray. The dataset has been divided in 10 folds, each containing approximately 10% of the complete dataset. Each fold has been extracted from the original data and used for testing. The remaining 90% in each fold has been used for training. Firstly, the dataset has been discretized and the logic formulas have been evaluated. Those formulas are in the form:

$$\begin{aligned}
&\text{IF} \quad p(4196) > 3.435 \quad \text{AND} \quad p(6041) > 3.004 \quad \text{THEN} \quad class1,\\
&\text{IF} \quad p(6573) < 2.059 \quad \text{AND} \quad p(6685) > 2.794 \quad \text{THEN} \quad class1,\\
&\text{IF} \quad p(1144) > 2.385 \quad \text{AND} \quad p(4373) < 3.190 \quad \text{THEN} \quad class-1,\\
&\text{IF} \quad p(4847) < 3.006 \quad \text{AND} \quad p(6376) < 2.492 \quad \text{THEN} \quad class-1,
\end{aligned}$$

where $p(i)$ represents the i-th probe. Each of the previous formulas is true for some samples of one class and it is false for all samples of the other class. We chose only these formulas for which is maximum the number of points satisfying the condition. The knowledge region for each class, are those given by the intersection of all chosen formulas.

Accuracy results of ten fold (1) and leave one out (2) cross validation

Dataset	ReGEC (1)	LF (1)	LF-ReGEC (1)	SVM(2)	TSP(2)
Leukemia	98.33%	86.36%	100%	98.61%	93.80%

Accuracy results reported in Table 5.1 show ten fold cross validation results for ReGEC, LP algorithm and SVM, leave one out cross validation for the last two. The LF-ReGEC method is fully accurate on the dataset.

5.2. Numerical experiments

LF-IReGEC was tested on publicly available benchmark data sets. The data has been obtained from k-TSP Program Download Page [25]. Dataset characteristics and references are reported in Table 5.2. Results regarding its performance in terms of classification accuracy are also presented. In Table 5.2, accuracy results are reported for the datasets of Table 5.2 for various methods. Null classification results have been computed on the complete datasets, supposing that all samples would have been classified in the class containing the larger number of samples. We note that the LP method is more accurate than TSP in three cases out of five. In all cases, the use of LF in conjunction with ReGEC, produces equal or higher accuracy results. We note that in LOO cross validation results are usually more accurate, because the training is done on all training set, except a sample. Nevertheless, LF-ReGEC well compares with SVM.

Datasets characteristics

Dataset	Platform	genes (P)	samples (N)		Reference
Leukemia	Affy	7129	25 (AML)	47 (ALL)	(Golub et al. [13])
Prostate1	Affy	12 600	52 (T)	50 (N)	(Singh et al. [23])
Prostate2	Affy	12 625	38 (T)	50 (N)	(Stuart et al. [24])
CNS	Affy	7129	25 (C)	9 (D)	(Pomeroy et al. [20])
GCM	Affy	16 063	190 (C)	90 (N)	(Ramaswamy et al. [21])

10.

Ten fold (1) and leave one out (2) cross validation accuracy results

Dataset	NULL	ReGEC (1)	LF (1)	LF-ReGEC (1)	SVM(2)	TSP(2)
Leukemia	65.27%	98.33%	86.36%	100%	98.61%	93.80%
Prostate1	50.98%	84.62%	77.80%	84.62%	91.18%	95.10%
Prostate2	56.81%	65.78%	73.50%	75.25%	76.14%	67.60%
CNS	73.52%	65.78%	79.20%	82.58%	82.35%	77.90%
GCM	67.85%	70.45%	79.60%	71.43%	93.21%	75.40%

6. Conclusions and future work

In the present work, we have proposed a new method to incorporate nonlinear knowledge provided by Logic Formulas in ReGEC, in a fashion similar to what has been proposed in Mangasarian and Wild, 2006[17]. Results show that accuracy of the new algorithm well compares with those of the single algorithms. In future, we will test and compare the LF-ReGEC method against other standard datasets. Finally, we believe further investigation needs to be devoted to the identification of knowledge regions, using only a subset of the training set with an incremental technique[4].

References

[1] P. Bertolazzi and G. Felici. Learning to classify species with barcodes. Technical report, Institute for System Analysis and Computer Science, Tech. Rep. 665, 2007.

[2] P. Bertolazzi, G. Felici, P. Festa, and G. Lancia. Logic classification and feature selection for biomedical data. *Computer and Mathematics with Applications*, 55(5):889–899, 2008.

[3] C. M. Bishop. *Neural networks for pattern recognition*. Oxford Press, 1995.

[4] C. Cifarelli, M. R. Guarracino, O. Seref, S. Cuciniello, and P. M. Pardalos. Incremental classification with generalized eigenvalues. *Journal of Classification*, 22(1):73–81, 2007.

[5] C. Cortes and V. Vapnik. Support vector machines. *Machine Learning*, 20:273–279, 1995.

[6] G. Felici, V. de Angelis, and G. Mancinelli. *Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques*, chapter Feature Selection for Data Mining, pages 227–252. Felici G. and Triantaphyllou E. eds., Springer, 2006.

[7] G. Felici and K. Truemper. A minsat approach for learning in logic domains. *INFORMS Journal on Computing*, 13(3):1–17, 2001.

[8] G. Felici and K. Truemper. *Encyclopedia of Data Warehousing and Mining*, volume 2, chapter The Lsquare System for Mining Logic Data, pages 693–697. J. Wang ed., Idea Group Inc., 2006.

[9] P. Festa and M.G.C. Resende. *Essays and Surveys on Metaheuristics*, chapter Grasp: An annotated bibliography, pages 325–367. Ribeiro C.C. and Hansen P. eds., Kluwer Academic Publishers, 2002.

[10] G. Fung and O. L. Mangasarian. Proximal support vector machine classifiers. In *Knowledge Discovery and Data Mining*, pages 77–86, 2001.

[11] M.R. Garey and D.S. Johnson. *Computer and Intractability: a guide to the theory of NP-completeness*. Freeman, San Francisco, 1979.

[12] G. H. Golub and R. Underwood. Stationary values of the ratio of quadratic forms subject to linear constraints. *Zeitschrift fr Angewandte Mathematik und Physik (ZAMP)*, 21(3):318–326, 1970.

[13] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[14] M. R. Guarracino, C. Cifarelli, O. Seref, and P. M. Pardalos. A classification algorithm based on generalized eigenvalue problems. *Optimization Methods and Software*, 22(1):73–81, 2007.

[15] M.R. Guarracino, D. Abbate, and R. Prevete. Nonlinear knowledge in learning models. In *Workshop on Prior Conceptual Knowledge in Machine Learning and Knowledge Discovery, European Conference on Machine Learning*, pages 29–40, 2007.

[16] Y. Lee and O. L. Mangasarian. Ssvm: A smooth support vector machine for classification, 1999.

[17] O. L. Mangasarian and E. W. Wild. Nonlinear knowledge-based classification. Technical report, Data Mining Institute Technical Report 06-04, Computer Science Department, University of Wisconsin, Madison, Wisconsin, November 2006.

[18] T.M. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[19] B. N. Parlett. *The Symmetric Eigenvalue Problem*. SIAM, Philadelphia, PA, 1998.

[20] S.L. Pomeroy, P. Tamayo, M. Gaasenbeek, L.M. Sturla, M. Angelo, M.E. McLaughlin, J.Y. Kim, L.C. Goumnerova, P.M. Black, C. Lau, J.C. Allen, D. Zagzag, J.M. Olson, T. Curran, C. Wetmore, J.A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D.N. Louis, J.P. Mesirov, E.S. Lander, and T.R. Golub. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415:436–442, 2002.

[21] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J.P. Mesirov, T. Poggio, W. Gerald, M. Loda, E.S. Lander, and T.R. Golub. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences of the United States of America*, 98:15149–15154, 2001.

[22] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[23] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203–209, 2002.

[24] R.O. Stuart, W. Wachsman, C.C. Berry, J. Wang-Rodriguez, L. Wasserman, I. Klacansky, D. Masys, K. Arden, S. Goodison, M. McClelland, Y. Wang, A. Sawyers, I. Kalcheva, D. Tarin, and D. Mercola. In silico dissection of cell-type-associated patterns of gene expression in prostate cancer. *Proceedings of the National Academy of Sciences of the United States of America*, 101:615–620, 2004.

[25] A.C. Tan, D.Q. Naiman, L. Xu, R.L. Winslow, and D. Geman. Simple decision rules for classifying human cancers from gene expression profiles. *Bioinformatics*, 21:3896–3904, 2005.

[26] K. Truemper. *Design of Logic-Based Intelligent Systems*. Wiley-Interscience, New York, 2004.

[27] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.