Classification Process Analysis of Bioinformatics Data With A Support Vector Fuzzy Inference System

STERGIOS PAPADIMITRIOY KONSTANTINOS TERZIDIS

Technological Educational Institute of Kavala,
Department of Information Management,
Kavala, 65404, Greece

Abstract: Recent complex bioinformatics data sets, such as Microarray and Proteomics data sets, which are characterized by sparsity and high dimensionality, require an analysis, which on the one hand offers a high degree of accuracy, but on the other hand simultaneously provides transparency in the analysis process. Recent Machine learning techniques, like e.g. the Support Vector Machines, own a remarkable generalization ability and are among the first choices to confront such complex data. However, the black-box structure of most machine learning algorithms constitutes a significant drawback. On the other hand, Fuzzy rule based systems form an attractive alternative since they result in linguistically, interpretable rules, but suffer from the problem of overfitting and are sensitive to the curse of dimensionality. In order to merge the advantages of both approaches Support Vector algorithms have been adapted for the identification of a Support Vector Fuzzy Inference (SVFI) system. However, although the high generalization performance of the SVM approach is retained, the SVFI rules usually lack understandability. The paper proposes the derivation of a simpler fuzzy system that approximates the accurate set of rules keeping only the more important aspects of the data. The approximation algorithms either receive an a priori description of a set of fuzzy sets or, especially for the case when interpretable fuzzy sets cannot be prespecified by the experts, an algorithm is presented for building them automatically. After the construction of the interpretable fuzzy partitions, the developed algorithms extract from the SVFI rules a small and consice set of interpretable rules. Finally, the Pseudo-Outer Product (POP) fuzzy rule selection orders the interpretable rules by using a Hebbian like evaluation in order to present the designer with the most capable rules.

Key-Words: Interpretable Rules, Fuzzy Rules, Rule Mining, Support Vector Machines, Kernel Classifiers

1 Introduction

The need to analyse and understand the scientific content of many bioinformatics data sets, especially of microarray and proteomic data sets, requires the use of sophisticated tools. Machine learning approaches, like the Support Vector Machines(SVM), which recently received a lot of attention, own a remarkable generalization ability and are among the first choises to confront such complex data. Fuzzy logic approaches in contrast result in predictions, which are easily interpretable and can be extra-polated in predictable ways. Fuzzy modeling however is limited because in its traditional formulation, the number of rules in a fuzzy model grows exponentially with the number of variables and resolution. The paper presents a simple but effective set of algorithms for merging the advantages of the above approaches by the construction of an approximate interpretable fuzzy system. We initially utilize the algorithms of [3] for the construction of a Support Vector Fuzzy Inference (SVFI) system, where the number of SVFI rules is controlled by the extracted support vectors. After that, an approximate system is constructed using interpretable application specific fuzzy sets.

The paper proceeds as follows: Section 2 deals with the automatic construction of interpretable fuzzy sets along dimensions for which the human experts cannot predefine them easily. Section 3 reviews the Support Vector Fuzzy Inference system and the corresponding algorithms for fuzzy rule construction from the trained SVMs. Section 4 concerns the derivation of the interpretable fuzzy rules from the Support Vector Fuzzy Inference rules. The results section (i.e. Section 6) presents applications of the techniques of the extracted rules and finally, section 7 concludes the work of the paper.

2 Features with unspecified a priori interpretable fuzzy sets

The presented approach requires the a priori specification of interpretable fuzzy partitions for every feature. However, frequently, interpretable fuzzy sets for some features cannot be prespecified by the human experts. In this case, we use an algorithm, which can be used to derive automatically a fuzzy partition that owns interpretability properties. The interpretable fuzzy set partition construction algorithm for the feature f proceeds by hierarchically merging candidate fuzzy partitions.

The key point for the effective generation of interpretable fuzzy partitioning is the design of the proper distance metric D_m that will best separate the m fuzzy sets. A clever design for D_m based on the concepts of external and internal distances is proposed in [10].

3 Support Vector Fuzzy Inference (SVFI) learning

This section reviews the framework for Support Vector Fuzzy Inference (SVFI), a method with high generalization and overfitting prevention ability. We implemented an approach similar to one presented in [3] for the extraction of Support Vector Fuzzy Inference rules, which we formulate belowin the algorithmic format.

Algorithm for SVFI fuzzy classifier identification

1. Construct a classification SVM from the training data to get a decision boundary in the feature space $\mathcal{F}$ of the form $f(\mathbf{x}) = sgn(\sum_{i \in S} y_i \cdot \alpha_i \cdot K(\mathbf{x}, \mathbf{x}_i) + b_0)$ where S is the set of obtained support vectors. Also K is the Gaussian Mercer kernel. This kernel defines implicitly a nonlinear mapping Φ from the input space X to a kernel induced feature space $\mathcal{F}$.

   Assign a suitable value to the regularization parameter C, and solve the corresponding quadratic program to obtain the Lagrange multipliers α_i and a suitable value for the constant bias term b_0. Effective algorithms for training SVMs [7] can be readily utilized at this stage.

2. Extraction of fuzzy rules from the SVM decision rule:
   $r \longleftarrow 0$ // r indexes the rule under construction
   for $i = 1$ **to** l // all training samples l
   if $\alpha_i > 0$ **then** // training samples i with Lagrange multipliers $\alpha_i > 0$ are support vectors
   $r \longleftarrow r+1$ // one more rule corresponding to the current SV will be constructed
   $\mathbf{z}_r \longleftarrow \mathbf{x}_i$ // the location parameter $\mathbf{z}_r$ for rule's r membership functions is the support vector x_i
   $c_r \longleftarrow y_i \alpha_i$ // the value of the singleton type output fuzzy set for rule r
   We denote by $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ the feature values of an input vector $\mathbf{x}$ and by $\mathbf{z}_r = [z_{r_1}, z_{r_2}, \ldots, z_{r_N}]$ the corresponding feature values of the support vector r. The constructed fuzzy rule takes the form:
   if *CloseToSV*(x_1, z_{r_1}) **and** *CloseToSV*(x_2, z_{r_2}) **and** ... **and** *CloseToSV*(x_N, z_{r_N}) **then** y **is** c_r
   Compute the weight of rule r as: $w_r \longleftarrow \alpha_i$ // the magnitude of the Lagrange multiplier signifies the importance of the corresponding rule
   end if
   end for

4 Interpretable rules

The SVFI approach has the following basic drawbacks:

- The SVFI rules are formulated with fuzzy sets defined in terms of the feature coordinates of the support vectors (i.e. the *CloseToSV()* fuzzy sets).

These later sets usually *do not have a particular meaning* to the human expert.

- For problems with large input feature space dimensionality N the obtained rules involve N conjunctive clauses and it is very difficult to comprehend them intuitively.
- When the number of support vectors becomes large the corresponding large SVFI rule base imposes additional interpretability problems.

Therefore, the derivation of interpretable and comprehensible to the human expert fuzzy rules from the SVFI rules is a very important task since it offers the potentiality for a readable and intuitive knowledge representation. The presented framework constructs rules that are expressed in terms of concepts that the human expert can understand easily. We develop a completely novel and effective framework for the extraction of interpretable rules from the SVFI when the interpretable fuzzy sets for a feature can be prespecified by the human expert.

Below we present the interpretable fuzzy system construction algorithm in pseudocode format. We recall that the main idea is to replace each of the SVFI clauses *CloseToSV*(x_f, z_{r_f}) by *FuzzyLinguisticVariable*(x_f, z_{r_f}) if the feature dimension f of the support vector $\mathbf{z}_r$ (i.e. z_{r_f}) attains a sufficiently high maximum membership $\mu_{F_{f,max}}(z_{r_f})$ at the *FuzzyLinguisticVariable* fuzzy set $F_{f,i}$.

Algorithm: Extraction of interpretable rules from the SVFI rules
// **Notation:**
// $\mathbf{z}_r$, z_{r_f} : the location parameter of the rth support vector
// and the corresponding feature coordinate f of $\mathbf{z}_r$
// x_f: the input value for the f feature
interpretableClauses = {};
ruleSupport = 1.0;
for all the features f of the support vector $\mathbf{z}_r$ **do**
// replace the clause *CloseToSV*(x_f, z_{r_f}) with a possible interpretable clause
for the interpretable fuzzy set $F_{f,max}$ of the fth feature variable for which z_{r_f} obtains the maximum membership
(e.g. for the interpretable fuzzy sets *HighExpression*, *LowExpression* a value 0.9 will attain maximum membership at the *HighExpression* set)
if $\mu_{F_{f,max}}(z_{r_f}) > \beta$ **then**
// β is the formely described threshold parameter
/* the support vector feature value z_{r_f} attains enough membership to the interpretable fuzzy set $F_{f,max}$, thus concatenate the new clause */
if $F_{f,max}$ is not the default fuzzy set **then**
interpretableClauses = *interpretableClauses* **and** (x_f **is** $F_{f,i}$)
(e.g. x_f can be a gene named BRC (i.e. $V_k \equiv BRC$) and the newly added clause can be: *BRC* **is** *HighExpression*)
// compute a measure of how much the new interpretable rule is supported by the SVM inference rule
ruleSupport = *ruleSupport**$\mu_{F_{f,max}}(z_{r_f})$
endif;
else
/* if even one conjunctive clause cannot have a satisfactory approximation with an interpretable fuzzy set (the default set included) the whole Support Vector rule cannot derive an interpretable rule */
interpretableClauses = {};
return *null*
end else;
end for;
if interpretableClauses != null)
/* interpretable clauses exist, construct the "then" part of the potential interpretable rule that will correspond to the support vector. This construction proceeds by first deciding if the possible rule is sufficiently significant by using the relative magnitude of the Lagrange multiplier. For the positive case we derive the "then" part as *Class* = "*Positive*" if the corresponding $b_i = \alpha_i \cdot y_i$ is ≥ 0 and *Class* = "*Negative*" at the opposite case. */

5 Pseudo-Outer Product Evaluation of the interpretable rules

Another approach for data-driven construction of fuzzy rules is based on Hebbian like learning [1] and is referred as the Pseudo Outer Product (POP) rule [11]. This approach evaluates the compatibility of all the possible rules with the training data and keeps the most promising ones.

The objective of the Pseudo-Outer Product (POP) rule evaluation phase is to compute the degree with which each derived interpretable rule is supported by

the training set. The POP evaluation phase computes for each training pattern the degree *antecedentRuleFiring* with which the antecedent part of an interpretable rule fires.

Denote by *antecedentRuleFiring* this degree. Subsequently it checks whether the predicted class aggress with the actual class of the training pattern. If so, it adds the computed *antecedentRuleFiring* to the total score over the training set, otherwise it subtracts it.

6 Results

At this section we demonstrate the potentiality of the presented interpretable rule extraction algorithms from the SVFI systems with two examples:

1. The exact discovery of the XOR boolean function from synthetic data derived from a continuous domain XOR like functional.

2. The approximate uncovering of fuzzy rules that implement a simple gene regulation network, from data generated by sampling the original fuzzy rules.

3. The discovery of useful and simple rules from a real public domain gene expression dataset concerning breast carcer tissue classification onto different subtypes.

Since the SVFI system implements accurately the RBF-SVM classification decision function, all the results concerning the generalization potential of the RBF-SVM are valid, and thus we do not elaborate on them. Instead we focus on the results obtained from the interpretable fuzzy rule extraction subsystem.

6.1 XOR-Data: Exact discovery of the XOR boolean function

For the XOR learning problem we use numerical data obtained from the function $y = -x_1 \cdot x_2$. Clearly this function evaluates precisely the logical XOR for the numbers -1 and 1 considering them as *false* and *true* respectively. The other values of the output are classified as false or true depending on the sign of y, i.e. *false* for the negative sign and *true* for the positive one. We generated 50 examples by producing uniform random values for x_1 and x_2 at the range $[-1, 1]$, computing the corresponding $y = x_1 \cdot x_2$ and outputing as class label the sign of y (i.e. $sgn\{y\}$).

On the contrary, the derivation of interpretable rules from the SVFI rules .

6.2 Blind discovery of fuzzy rule systems

In order to test the efficiency of the interpretable SV-based rule extraction approach we conducted experiments with synthetic data generated by randomly sampling the operation of known fuzzy rule systems. The objective is to test the efficiency of the algorithms at uncovering the fuzzy rules from which the data were generated, by using only the data samples without any *a-priori* knowledge about the data generating rules.

We use a simple gene regulation network as the generator of controlled synthetic data for training. The small gene regulation example consists of three genes treated as variables that receive continuous values at the range -1 to 1, with -1 meaning totally underexpressed ("Low"), 0 totally unaffected by the experiment and 1 totally overexpressed ("High"). The continuous range of values between these extremes fuzzifies the concept of gene expression, as usually, e.g. a value of 0.8 for one gene signifies larger relative expression level at the particular experiment from a value of 0.7. The *GEnchancer* is an enhancer gene i.e. one that its expression enhances the level of expression of the control gene *GControl*. Similarly, the *GRepressor* gene is a suppressor gene for GControl, i.e. its expression tends to block the expression of the control gene *GControl*.

We generated training data sets by randomly sampling the input variables *GEnhancer* and *GRepressor* (taking about 50 samples). Consequently we conclude at the value of *GControl* variable by evaluating the fuzzy rule system. In order to treat the learning task as an SVM classification problem we discretize the positive cases of the outcome *GControl* to 1 (i.e. overexpression) and the negative one to -1 (i.e. underexpression). In particular the first and third rule reveal qualitatively the operation of the *Enhancer* gene to up-regulate the control gene, while the second and fourth rule discover the role of the *Repressor* gene. We should note that by varying the specification of the interpretable fuzzy sets, the system derives slightly different rule sets. These fuzzy rule sets describe the properties of the data in terms of the altered interpretable fuzzy sets.

6.3 Gene Expression Dataset

Below we describe an application for a gene expression analysis task, and in particular to the discovery of rules for the classification of breast cancer subtypes from gene expression data. The data set analyzed were obtained from the CD-ROM of the book of Sorin Draghici [12] and concern the work of Hedenfalk [13]. Hedenfalk and coworkers [13] studied gene expression, using spotted cDNA arrays containing 6512 sequences representing 5361 unique genes, in in a total of 22 breast tumors obtained from individuals of three categories:

1. with BRCA1 mutations (BRCA1 class,7 samples).
2. with BRCA2 mutations (BRCA2 class ,8 samples).
3. individuals which were wild type at these loci (WildType class, 7 samples).

Histopathological and molecular evidence relating to estrogen and progesterone receptors suggests that tumors originating in individuals with BRCA1 mutations are molecularly distinct from those with BRCA2 mutations. One of the main goals of the gene expression analysis is to further elaborate such hypotheses.

Applying the inference system we extracted discriminating rules which involve genes that tend to follow the significant gene list detected by Hedenfalk and coworkers [13], quantitatively there is about 80% overlap. Since we use the SVM learning in 2-class classification we perform three classification experiments (i.e. BRCA1 vs BRCA2 and Wild Type, BRCA2 vs BRCA1 and Wild Type and Wild Type vs. BRCA1 and BRCA2). We observed a tendency of the three classifiers to use the same genes, a fact that can have biological significance in terms of the operation of these genes at the corresponding biological processes. By redefining the concepts of Low and High gene expression with more "strict" membership functions we extracted a smaller number of rules, but we expect these rules to provide more concise descriptions of the essential aspects of the data.

7 Conclusions

The paper has presented a dual approach to the problem of fuzzy system identification from training examples that extends the work of [3] by building interpretable fuzzy-rule systems on top of the SVFI algorithms proposed in [3]. Future work continues with the elaboration of the interpretable fuzzy system construction with algorithms that adapt the interpretable membership functions in the spirit of [5, 6]. The LibSVM based extensions of Support Vector Learning for fuzzy identification and the fuzzy inference engine are available upon request from the authors.

References

[1] Simon Haykin, Neural Networks, MacMillan College Publishing Company, Second Edition, 1999

[2] Bart Kosko, *Fuzzy Engineering*, Prentice Hall, 1997

[3] Yixin Chen, James Z. Wang, "Support Vector Learning for Fuzzy Rule-Based Classification Systems", *IEEE Transactions on Fuzzy Systems*, Vol. 11, No. 6, December 2003, p. 716-728

[4] Jung-Hsien Chiang, Pei-Yi Hao, "Support Vector Learning Mechanism for Fuzzy Rule-Based Modeling: A New Approach", Vol. 12, No. 1, February 2004, pp. 1-12

[5] D. Chakraborty and N. R. Pal, "A Neuro-Fuzzy Scheme for Simultaneous Feature Selection and Fuzzy Rule-Based Classification", *IEEE Transactions on Neural Networks*, Vo. 15, No. 1, January 2004, p. 110-123

[6] M. J. del Jesus, F. Holfmann, L. Jun Navascues, L. Sanchez, "Induction of Fuzzy-Rule-Based Classifiers With Evolutionary Boosting Algorithms", *IEEE Trans. on Fuzzy Systems*, Vo. 12, No. 3, June 2004, pp. 296-308

[7] Chang, C.-C., Lin, C.J, "LIBSVM: A library for support vector machines",2001, Available on-line:
http://www.csie.ntu.edu.tw/ cjlin/libsvm

[8] S. Papadimitriou, K. Terzidis, Symbolic Adaptive Neuro-Fuzzy Inference for Data Mining of Heterogenous Data, *Intelligent Data Analysis*

(IDA) journal, Volume 7 (4), IOS Press, 2003, 327-346

[9] S. Papadimitriou, S.D. Likothanassis, Kernel-Based Self-Organized Maps trained with Supervised Bias for Gene Expression Data Analysis, *Journal of Bioinformatics and Computational Biology (JBCB)*, Imperial College Press, Vol. 1, No. 4 (2004) 647-680

[10] S. Guillaume, B. Charnomordic, "Generating an Interpretable Family of Fuzzy Partitions From Data", *IEEE Trans. Fuzzy Systems*, Vo 12, No 3, June 2004, p. 324-335

[11] C. Quek, R. W. Zhou, "The POP learning algorithms: reducing work in identifying fuzzy rules", *Neural Networks*, Vo 14, 1431-1445, 2001

[12] Sorin Draghici, "Data Analysis Tools for DNA Microarrays", Chapman & Hall/CRC, 2003

[13] Ingrid A. Hedenfalk, "Gene Expression Profiling of Hereditary and Sporadic Ovarian Cancers Reveals Unique BRCA1 and BRCA2 Signatures", *Journal of the National Cancer Institute*, Vol. 94, No. 13, July 3, 2002