MINING EFFICIENT AND INTERPRETABLE FUZZY CLASSIFIERS FROM DATA WITH SUPPORT VECTOR LEARNING

STERGIOS PAPADIMITRIOU KONSTANTINOS TERZIDIS

Dept of Information Management, Technological Educational Institute of Kavala Kavala, 65404, GREECE

Abtract: The construction of fuzzy rule-based classification systems with both good generalization ability and interpretability is a chalenging issue. The paper aims to present a novel framework for the realization of these important (and many times conflicting) goals simultaneously. The generalization performance is obtained with the adaptation of Support Vector algorithms for the identification of a Support Vector Fuzzy Inference (SVFI) system. The SVFI is a fuzzy inference system that implements the Support Vector network inference and inherits from it the robust learning potential. The construction of the SVFI is based on the algorithms presented in [6]. The contribution of the paper is the development of algorithms for the construction of interpretable rule systems on top of the SVFI system. However, the SVFI rules usually lack interpretability. For this reason, the accurate set of rules is approximated with a simpler interpretable fuzzy system that can present insight to the more important aspects of the data.

Key-Words: Support Vector Machines, Fuzzy Identification, Data Mining, Interpretable Rules, Generalization Performance

1 Introduction

Recently, another approach to the fuzzy identification problem based on Support Vector Learning has been developed [6, 7]. This approach aims to offer a robust framework for fuzzy systems that are able to generalize effectively [3]. The technique of Support Vector Machine (SVM) grounded on the Statistical Learning Theory of Vapnik [1] has received recently a lot of attention by researchers since it demonstrated success in many difficult application domains [2, 12]. The SVM is an approximate implementation of the structural risk min*imization* inductive principle that aims at minimizing an upper bound on the generalization error of a model, rather the usual minimization of the mean-square error over the training set accomplished with most training algorithms. We utilize the algorithms of [6] for the construction of a Support Vector Fuzzy Inference (SVFI) system.

The resulting fuzzy system implements accurately the SVM inference and thus it is generally quite effective. But, the interpretation of its rules by the human expert is difficult, since the rules are defined in terms of the proximity to support vectors, a concept that usually does not provide an intuitive information to the human expert.

The major contribution of the paper is the presentation of a simple but effective set of algorithms for the construction of an approximate interpretable fuzzy system from the accurate SVM-based one. The advantages of the presented approach can be summarized to:

- 1. The rules are expressed with domain specific fuzzy sets and thus they can directly provide intuition to the human expert.
- 2. It has generally much smaller number of rules from its "base" SVFI system and each such rule involves usually a *small subset* of the input features. To the

contrary at the SVFI rules every rule consists of the conjunction of clauses, with one clause of the form *CloseTo* for *every* feature dimension.

3. The approximation accuracy and the complexity of the interpretable set of rules can be traded off dynamically by adjusting a set of thresholds.

We should note however that the implementation of the system requires an a priori characterization of the interpretable fuzzy sets for the input variables of interest. Thus, we confront application domains for which interpretable partitions for the input variables can be defined a priori as for example most gene expression analysis tasks. Since for most features we can define approximately interpretable fuzzy sets, our method can be applied in order to extract possibly interesting, simple and interpretable rules.

The paper proceeds as follows: Section 2 reviews the Support Vector Fuzzy Inference system and the corresponding algorithms for fuzzy rule construction from the trained SVMs. This presentation is based on the work of [6]. Section 3 concerns the main contribution of the paper, i.e. the derivation of the interpretable fuzzy rules from the Support Vector Fuzzy Inference rules. The results section (i.e. Section 4) presents applications of the techniques, which deal with both synthetically generated data and real data sets. Finally, section 5 concludes the work of the paper.

2 Support Vector Fuzzy Inference (SVFI) learning

This section reviews the framework for Support Vector Fuzzy Inference (SVFI) proposed in [6]. This method provides a solid foundation for obtaining generalization and over-fitting prevention ability. The main contribution of the current work is the construction of a reduced set of interpretable rules from the SVFI that pinpoint many interesting aspects of the data set in easily conceivable representation for the human expert. The corresponding interpretable rule set does not claim to maximize the classification accuracy. Its purpose is to discover important aspects of the dataset and perhaps to help in elaborating domain specific knowledge. The rules discussed at the current section are expressed in terms of the Support Vectors and implement accurately the SVM inference. Thus, the presented methodology by utilizing concurrently two sets of rules offers both interpretability and accuracy.

We proceed by describing the rules that correspond directly to the SV inference, referred to as the SV-Inference rules. The SV-Inference rules are extracted by utilizing the algorithms of Chen & Wang [6]. Since the presented interpretable rule is constructed "on-top" of the SVFI system we present a self-contained description of these algorithms in order to preserve the continuity of the presentation and to clarify the close coupling of the two systems (i.e. the SVFI and the interpretable rule systems).

The general form of the Support Vector Fuzzy Inference (SVFI) rules is:

Rule k: if P_1^k and P_2^k and $\ldots P_N^k$ then c_k

where $P_i^k, i = 1, \ldots, N$ are fuzzy clauses, of the form

 x_i is CloseToSV(k, i),

that test the membership of the *i*th "coordinate" x_i of the input vector $\mathbf{x} = [x_1, \ldots, x_N]$ at the *i*th fuzzy set of the *k*th SV, *CloseToSV(k, i)*. The later sets *CloseToSV(k, i)* fuzzify the numerical distance of the x_i input coordinate to the x_i^k coordinate of the *k*th support vector. A Gaussian function of the form: $\mu_i^k(x_i) = \exp(-\frac{1}{2}\left(\frac{x_i^k - x_i}{\sigma_k}\right)^2)$, computes the membership by quantifying the proximity of the input value component x_i to the value x_i^k of the *i*th component of the support vector SV_k . Also, the parameters c_k are real constants, i.e. $c_k \in \Re$. We choose product as the fuzzy conjunction operator, addition for fuzzy rule aggregation and Center Of Area (COA) defuzzification. The resulting model becomes a special case of Takagi-Sugeno (TS) fuzzy model [13].

The input-output mapping $\mathbf{F'}$ that the SVFI model performs and the decision function for classification problems $\mathbf{F}(\mathbf{x})$ can be expressed as

$$\mathbf{F}'(\mathbf{x}) = \frac{\sum_{k=1}^{M} c_k \prod_{i=1}^{N} \mu_i^k(x_i)}{\sum_{k=1}^{M} \prod_{i=1}^{N} \mu_i^k(x_i)} = \sum_{k=1}^{M} c_k R_k(x) \qquad (1)$$

$$\mathbf{F}(\mathbf{x}) = sgn\{\mathbf{F}'(\mathbf{x})\}\tag{2}$$

where $\mathbf{x} = [x_1, \ldots, x_N]^T \in \Re^N$ is the input, M is the number of rules, N is the number of conjunctive clauses of the kth rule which is equal to the dimensionality of the input vector \mathbf{x} and $\mu_i^k(x_i)$ computes the membership

of the input variable x_i in the fuzzy set CloseToSV(k, i). The term contributed by rule k to the numerator of Equation 1 is $c_k \cdot \prod_{i=1}^{N} \mu_k^i(x_i)$, and the kth rule's relative strength, $R_k(\mathbf{x})$, with which this rule is involved at the decision for input vector \mathbf{x} is:

$$R_{k}(\mathbf{x}) = \frac{\prod_{i=1}^{N} \mu_{i}^{k}(x_{i})}{\sum_{k=1}^{M} (\prod_{i=1}^{N} \mu_{i}^{k}(x_{i}))}$$
(3)
$$= \frac{K'(\mathbf{x}, \mathbf{z}_{k})}{A}$$
$$= \frac{\exp\left(\frac{-1}{2} \|\frac{\mathbf{x} - \mathbf{z}_{k}}{\sigma_{k}}\|^{2}\right)}{A}$$

where $\mathbf{z}_k = [z_k^1, z_k^2, \dots, z_k^N]^T \in \mathbb{R}^N$ controls the location parameters (i.e. the Gaussian centers) of $\mu_i^k, k = 1, \dots, M, \ i = 1, \dots, N$. The denominator $\sum_{k=1}^M (\prod_{i=1}^N \mu_i^k(x_i))$ attains a constant value denoted by A. Since the denominator is constant it does not affect the classification decision and thus we consider equivalent problems, already scaled properly. Therefore, at the expansion of equation 4 (presented below) we consider only the numerator $K'(\mathbf{x}, \mathbf{z}_k)$. It is now evident that $K'(\mathbf{x}, \mathbf{z}_k)$ has the general form of the Gaussian kernel that is used to implement the Radial-Basis SVM.

The decision rule of the output of the fuzzy system of equation 1 can be expressed in terms of kernel functions as:

$$\mathbf{F}(\mathbf{x}) = sgn\{\sum_{k=1}^{M} c_k K'(\mathbf{x}, \mathbf{z}_k)\}$$
(4)

The kernel K' is a translation invariant kernel defined as $K'(\mathbf{x}, \mathbf{z}_k) = \prod_{i=1}^{N} \mu_i^k (x_i - z_i^k)$ and each μ_i^k membership function is of the familiar one-dimensional Gaussian type. In order to implement $K'(\mathbf{x}, \mathbf{z}_k)$ we use scaled and shifted Gaussians, therefore $K(\mathbf{x}, \mathbf{z}_k) = K'(||\mathbf{x} - \mathbf{z}_k||)$, where the location, of the Gaussians are specified with the location vector \mathbf{z}_k .

The SV learning algorithm constructs a fuzzy system with N inputs and M number of rules (one rule for every SV). The number of rules M is derived after the solution to the SVM quadratic programming problem.

The M fuzzy rules can be parameterized with a set of location parameters $\{\mathbf{z}_1, \ldots, \mathbf{z}_M\} \in \Re^N$ for the Gaussian centers that determine the membership functions of the if-part fuzzy rules, and a set of real numbers $(\{c_0, \ldots, c_N\} \in \Re)$ for the constants of the then-part of the fuzzy rules.

3 Interpretable rules

Linguistic rule extraction is a very important issue within Knowledge-Based Neurocomputing. Support Vector Machines as well as the equivalent SVFI system interpolate relatively easily large sets of data and provide a means for effective generalization. However, 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 perform an Interpretable Fuzzy Set (IFS) approximation of the SVFI system with one based on a priori specified interpretable fuzzy sets. Specifically, for each feature dimension f the domain expert can define a set of interpretable fuzzy sets that are meaningful at the particular application domain. For example at a medical diagnosis application, for the ArterialPressure feature, fuzzy sets such as VervLow, Low, Medium, High, VeryHigh can offer direct interpretation and intuition. Clearly, according to the application domain of interest, we have to decide on the fuzzy set types for the interpretable fuzzy sets (e.g. triangle shaped, trapezoids, Gaussian etc.) and on their names (proper names improve the readability of the extracted interpretable rules). A graphical Java interface allows the user to define conveniently these characteristics of the fuzzy system.

After the explicit definition of domain specific fuzzy sets the task of generating fuzzy rules that are expressed in terms of these sets from the SVFI system is completely computational and proceeds without the intervention of the human expert.

For a support vector sv with scalar value $sv_f, sv_f \in$ \Re , for its feature dimension f, the degree of membership $\mu_{IFS_{f_i}}(sv_f)$, of sv_f at every Interpretable Fuzzy Set $IFS_{f,i}$ of feature f, is evaluated. Since the emphasis is on obtaining a small set of interpretable and comprehensible rules, for each feature f, we keep as a candidate for clause generation only the interpretable fuzzy set with the maximum membership, denoting it as $IFS_{f,max}$. We consider the case that sv_f is "sufficiently within" the interpretable fuzzy set $IFS_{f,max}$ of feature f, i.e. $\mu_{IFS_{f,max}}(sv_f) > \beta$, where $\beta \in \Re$ is a threshold parameter. At this case, for each $CloseTo(IFS_{f,max}, sv_f)$ fuzzy clause we create an approximate interpretable fuzzy clause in terms of $IFS_{f,max}$. The threshold parameter β determines the number and the quality of the derived rules. Clearly, with larger thresholds we construct fewer rules but of better quality.

The membership values $\mu_{IFS_{f,max}}(sv_f)$ are used to compute a measure of the accuracy with which the original SVFI rule is approximated. We define a *Support Vector Rule Similarity* (SVRS) parameter for a possible interpretable rule r extracted from the support vector **sv** as:

$$SVRS_r = \prod_{f=1}^N \mu_{IFS_{f,max}}(sv_f) \tag{5}$$

where N is the dimensionality of \mathbf{sv} . Thus, the $SVRS_r$ parameter for a rule r is defined as a product of the similarities of all its clauses to the corresponding SVFI clauses (i.e. the parameters $\mu_{IFS_f}(sv_f)$). The product is justified by the conjunctive structure of the rules.

The construction of the "then" part and therefore of the class label of the rules is straightforward and depends on the sign of the c_r parameters that constitute the "then" part of the SVFI rules. The c_r parameters are computed with the SVFI algorithm presented in Section 2. However, a bit more technical is the extraction of information for the strength of each rule from the SVFI training results. To accomplish this, we detect the minimum and maximum values of the values $c_r = y_i \cdot \alpha_i, c_r \in \Re$. Clearly, their range depends on many factors, e.g. the specific problem, the particular training set, the RBF-SVM parameters C (complexity regularization parameter) and σ (spreading of Gaussian centers parameter) etc. Although, in absolute terms the c_r values do not have a particular meaning, their relative magnitude indicates the "weight" (or significance) of the corresponding rule. Therefore, an additional rule pruning step can be performed by avoiding to consider those SVFI rules that do not contribute significantly either for the positive or the negative class. As a particular example one rule with $c_r = 0.8$ is a "weak" one if the class range is [-31.7, 35.2] since it affects slightly the classification but the same rule is a "strong" in favor for the positive class one if the class range is [-0.8, 0.9]. Thus, we detect the minimum and maximum values of the class range (i.e. values c_r) and we normalize this range to [-1.0, 1.0] in order to obtain effectively the "weight" w_r of each rule r. The normalization unbiases the weight parameter from the range of c_r values. The "weight" parameter corresponds to the strength of the corresponding rule at the SVFI system.

Recapitulating, the weight parameter w_r quantifies the classification strength of the SVFI rule, while the formely described $SVRS_r$ the accuracy of its interpretable rule "version". Thus, the multiplication of the weight parameter w_r with the Support Vector Rule Similarity parameter, $SVRS_r$, adjusts the weight of the interpretable rule considering also the accuracy of the interpretable rule in representing the original SVFI rule. We denote the combined quality measure for each interpretable fuzzy rule r as Significancer, i.e.

$$Significance_r = w_r \cdot SVRS_r \tag{6}$$

A useful concept, especially for high dimensional datasets, for the reduction of the syntactic complexity of the interpretable rules is the one of the *default* interpretable fuzzy set. At the frequent case in many applications where a variable most often attains the highest memberships to a particular fuzzy set, that fuzzy set can be treated as the default interpretable fuzzy set. Clauses expressed in terms of the default interpretable fuzzy sets are not displayed explicitly at the representation of the interpretable fuzzy rules. For example, since most genes at a gene expression experiment are not affected significantly by the experiment's condition, the linguistic variable "Unchanged" can be implicitly assumed for all genes not appearing at the clauses of an interpretable rule. The concept of the default interpretable fuzzy set allows the construction of readable rules that involve a small number of fuzzy clauses at their antecedent part.

Frequently in practice we can specify easily interpretable fuzzy sets for many input feature dimensions. For example, at a gene expression analysis experiment with normalized data where -1(+1) is the maximum underexpression (overexpression) and the value 0 (zero) corresponds to absolutely unaffected genes, we can specify a variety of fuzzy sets according to our apriori knowledge. However, there can exist also features for which interpretable fuzzy sets, cannot be a priori specified. At these cases we can simply ignore the corresponding feature dimensions at the rule extraction. Alternatively, for those features, data-driven interpretable fuzzy rule extraction algorithms like the hierarchical fuzzy partitioning algorithm proposed in [14] can be utilized.

Below we recapitulate the interpretable fuzzy system construction algorithm in pseudo-code 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 $F_{f,i}$ of the *f*th feature variable that is closest to z_{r_f}

(e.g. for the interpretable fuzzy sets *HighExpression*, *LowExpression* a value 0.9 will be closest to the *High-Expression* set)

if $\mu_{F_{f,i}}(z_{r_f}) > \beta$ then

// β is the formely described threshold parameter

/* the support vector feature value z_{r_f} attains enough membership to the interpretable fuzzy set $F_{f,i}$, thus concatenate the new clause */

if $F_{f,i}$ is not the default fuzzy set then

interpretable Clauses = interpretable Clauses and $(x_f \text{ is } F_{f,i})$

(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

 $ruleSupport = ruleSupport^* \mu_{F_{f,i}}(z_{r_f})$

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. */

4 Results

At this section we demonstrate the potentiality of the presented interpretable rule extraction algorithms from the SVFI systems with an example of a multiclass classification experiment with the UCI Iris standard data set. Since the SVFI system implements accurately the RBF-SVM classification decision function, all the results concerning the generalization potential of the RBF-SVM [12, 1, 8, 4] are valid, and thus we do not elaborate on them but instead we focus on the results obtained from the interpretable fuzzy rule extraction subsystem.

4.1 Multi-class classification

The main motivation of our method is to gain insight at the data by discovering interesting rules. Therefore the emphasis is not on building a classifier based on the interpretable rules. However, methods developed from the SVM machinery for multi-class classification are well suited for the exaction of multi-class interpretable classification rules. Specifically, by designing M binary classifiers we derive rules that distinguish each class from the rest ones. With these M experiments we can usually gain much insight at the characteristics of each class. The one-versus-the-rest approach can be used at the SVFI inference. This method constructs a set of binary classifiers f^1, \ldots, f^M , each trained to separate one class from the rest, and then it combines them by doing the multi-class classification according to the maximal output before applying the sgn functions; that is, by taking

$$arg_{max_{j=1,...,M}}g^{j}(x), \ g^{j}(x) = \sum_{i=1}^{m} y_{i}\alpha_{i}K(x,x_{i}) + b^{j}$$
 (7)

We utilize the well known Iris database in the pattern recognition literature. This data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other two; the latter are NOT linearly separable from each other. The predicted attribute is the class of iris plant. The attribute information has as follows: 1. sepal length in cm, 2. sepal width in cm 3. petal length in cm 4. petal width in cm and the three classes are: 1. Iris Setosa, 2. Iris Versicolour, 3. Iris Virginica

The methodology of extracting interpretable rules is to consider first the case that we use the SVFI that distinguishes Iris-Setosa from the rest of the two classes (Iris-Virsinica and Iris-Versicolor) As a consequence we define a set of rules that distinguish Iris-Setosa from the rest two classes. Similarly, we extract rules for the Iris-Virsinica and Iris-Versicolor class.

For each rule we define the rule's *coverage* as the percentage of instances of the predicted class covered by the rule's premise i.e,

$$R_c = \frac{N_{ci}}{N_{ti}} \tag{8}$$

where N_{ci} is the number of covered instances of class C_i by the rule and N_{ti} is the total number of instances of class C_i .

Also for each rule we define the rule's *precision* that indicates from the instances that activate the rule's premise, the percentage that is of the predicted class i.e.

$$R_p = \frac{N_{cp}}{N_{tp}} \tag{9}$$

where N_{cp} is the number of instances of the predicted class that activate the rule's premise and N_{tp} is the total number of instances that activate the rule's premise.

Results for IRIS. For the IRIS data set we used three Gaussian interpretable fuzzy sets for each of the four features.

The derived rules for the IRIS set are the following:

R1: if petalLength is Small and petalWidth is Small then class = IrisSetosa, Coverage: 1, Precision: 1

R2: **if** petalLength **is** *Medium* **and** petalWidth **is** *Medium* **then** class = IrisVersicolour, *Coverage*: 1, *Precision*: 0.9

R3: if petalLength is Large and petalWidth is Medium then class = IrisVirsinica, Coverage: 0.1, Precision: 0.9

R4: **if** petalLength **is** *Medium* **and** petalWidth **is** *Large* **then** class = IrisVirsinica, *Coverage*: 0.4, *Precision*: 0.9

R5: if petalLength is Large and petalWidth is Large then class = IrisVirsinica, Coverage: 0.5, Precision: 0.9

Although the support vector machine and the corresponding SVFI attain a very high cross-validated leaveone-out generalization performance near 98% the interpretable rule system has a performance of 95% for the Iris Setosa class, 85% for the Iris Virsinica and 82% for the Iris Versicolour. However the resulted rules are very simple and they own the interpretability property for which they were designed.

5 Conclusions

The paper has presented a dual approach to the problem of fuzzy system identification from training examples that extends the work of [6] by building interpretable fuzzy-rule systems on top of the SVFI algorithms proposed in [6].

During the first phase, the SVFI framework of [6] is utilized as a disciplined method for the generation of fuzzy if-then rules from the training data that is capable of achieving remarkable generalization performance. Concerning applications that involve high-dimensional feature spaces, very important is the fact that the Support Vector Machine (SVM) can work very effectively at a high (or even infinite) dimensional feature space. Since the constructed Support Vector Fuzzy Inference (SVFI) system mimicks accurately the support vector machinery it owns all of its generalization efficiency.

However the SVFI system lacks the desired property of interpretability and cognitive meaning to the human domain expert. In order to derive also interpretable rules we have developed a second phase, in which we extract another set of rules by examining the structure of the SVFI rules. At this phase we take as an *a priori* "bias" an approximate specification of the fuzzy sets that the human expert considers as relevant to the application domain of interest. The derived set of rules at this stage although is not as powerful as the SVFI system, can offer very useful insight to the structure of the data. Indeed, the constructed rules tend to be meaningful since they are stated in terms of the fuzzy sets defined by the domain experts. Concise rules that highlight aspects of the data generation process can be revealed.

Acknowledgment

This work was partially supported from a European Union funded EPEAK II project "Arximidis", code 04-3-001/5, performed at the Technological Educational Institute of Kavalas, Dept. of Information Management, Greece.

References

- V. N. Vapnik., 1998, Statistical Learning Theory, New York, Wiley 9. V. N. Vapnik, "An Overview of Statistical Learning Theory", IEEE Trans. On Neural Networks, Vol. 10, No 5, 1999, pp. 988-999
- [2] Simon Haykin, Neural Networks, MacMillan College Publishing Company, Second Edition, 1999
- B. Scholkopf, A. J. Smola, R. C. Williamson, P. L. Bartlett, "New support vector algorithms", Neural Computation:1207-1245, 2000
- [4] Bernhard Scholkopf, Alexander J. Smola, "Learning with Kernels: Support Vector Machines, Regularization and Beyond", MIT Press 2002
- [5] J. S. R. Jang and C. T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference systems", *IEEE Trans. Neu*ral Networks, vol. 4, pp. 156-159, Feb. 1993
- [6] Yixin Chen, James Z. Wang, "Support Vector Learning for Fuzzy Rule-Based Classification Systems", *IEEE Transactions on Fuzzy Sys*tems, Vol. 11, No. 6, December 2003, p. 716-728

- 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
- [8] M. M. S. Lee, S. S. Keerthi, C. J. Ong, and D. DeCoste, "An Efficient Method for Computing Leave-One-Out Error in Support Vector Machines With Gaussian Kernels", *IEEE Transactions on Neural Networks*, Vol. 15, No. 3, May 2004, pp. 750-757
- [9] 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
- [10] 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
- [11] 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
- [12] A. Ifantis, S. Papadimitriou, Support Vector Identification of Seismic Electric Signals, *IJPRAI (International Journal Pattern Recognition and Artificial Intelligence)*, World Scientific, Vol. 17, No. 4 (June 2003), p. 545-566
- [13] T. Takagi, M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control", *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-15, pp. 116-132, Feb 1985
- [14] 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