On the usefulness of fuzzy SVMs and the extraction of fuzzy rules from SVMs

Christian Moewes[*], **Rudolf Kruse**[†]

Faculty of Computer Science, University of Magdeburg, Germany

Abstract

In this paper we reason about the usefulness of two recent trends in fuzzy methods in machine learning. That is, we discuss both fuzzy support vector machines (FSVMs) and the extraction of fuzzy rules from SVMs. First, we show that an FSVM is identical to a special type of SVM. Second, we categorize and analyze existing approaches to obtain fuzzy rules from SVMs. Finally, we question both trends and conclude with more promising alternatives.

Keywords: Classification, fuzzy rule-based classifiers, fuzzy SVM, SVM

1. Introduction

Kernel-based methods, and support vector machines (SVMs) in particular [1], play one of the most important roles in machine learning today. They are announced to both generalize considerably on unseen data and perform well on high-dimensional input spaces. Nonetheless, the application of these methods is not popular compared to intuitive learning machines.

Especially in automation and control, the application of models based on fuzzy set theory (FST) [2] became substantive. The vague expressions that are used by human beings to describe processes can be modeled gracefully by FST. Fuzzy classifiers (FCs) based on linguistic rules provide a comprehensive way to illustrate underlying concepts of complicated systems. Nowadays, they can be found in many real-world applications [3].

Several attempts have been made to find connections between fuzzy models and SVMs. Essentially, two directions can be distinguished in the research community. First of all, we find approaches that try to incorporate FST directly into SVMs, *i.e. Fuzzy Support Vector Machines (FSVMs)* [4, 5]. The main motivation is the fact that SVMs are quite sensitive to outliers and noise. FST provides an appropriate toolbox of methods to tackle those problems.

The second direction focuses on the generation of fuzzy classifiers based on the output of an SVM. In essence, we encounter methods to extract fuzzy-rule based classifiers from SV machines for both settings, *i.e.* classification [6, 7] and regression [8]. The objectives are different compared to the first direction. Fuzzy models become cumbersome in complex systems with dozens of input variables since they suffer from “the curse of dimensionality”. Thus combining the generalization of SVMs with the interpretability of FCs might be a striking idea to overcome these difficulties.

Whereas we originally thought that these two directions can be unified [9], we see things more realistic now having studied most existing approaches. To get started, we will briefly introduce SVMs in Section 2. Afterwards we will give an overview of fuzzy SVMs in Section 3. Then in Section 4 we present four different ways to extract fuzzy rules from an SVM. Section 5 will discuss major drawbacks of both research trends. Furthermore, we question the usefulness of some presented ideas based on reasonable facts. Practical alternatives to both trends will be given. Finally, we will conclude with by summarizing the main thoughts in Section 6.

2. Support Vector Machines

Suppose we are given an input space $\mathcal{X}$ (not necessarily a vector space) and an output space $\mathcal{Y}$. Since we deal with a binary classification problem, $\mathcal{Y} = \{\pm 1\}$. We observe l training patterns $(x_i, y_i) \in \mathcal{S} \subseteq \mathcal{X} \times \mathcal{Y}$ where $i = 1, \ldots, l$. They have been drawn i.i.d. from an unknown distribution. If $\mathcal{X} \subset \mathbb{R}^n$, then $x_i \mapsto \mathbf{x}_i$. Our goal is to separate the data with a linear hyperplane $\{\mathbf{x} \mid \langle \mathbf{w}, \mathbf{x} \rangle + b = 0\}$ where $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are the norm vector and the bias of the hyperplane, respectively. The decision function of a hyperplane classifier which shall predict y' for any $\mathbf{x}$ corresponds to

$$f(\mathbf{x}) = \operatorname{sgn}\left(\langle \mathbf{w}, \mathbf{x} \rangle + b\right). \tag{1}$$

We are looking for the hyperplane that maximizes the margin between every training pattern and the hyperplane. Such a hyperplane is called optimal since it is unique and has the best generalization performance on unseen data. If all points $(x_i, y_i) \in \mathcal{S}$ can be separated linearly by a hyperplane, we can obtain the optimal hyperplane by solving a quadratic optimization problem with linear inequality constraints. Usually not all training patterns can be separated perfectly. Therefore we introduce slack variables ξ_i with $i = 1, \ldots, l$ in order

[*] E-Mail: cmoewes@ovgu.de

[†] E-Mail: kruse@iws.cs.uni-magdeburg.de

to relax the optimization problem to

$$\begin{array}{lcr} \min_{\mathbf{w},b,\xi} & \tau(\mathbf{w},\xi) = \frac{1}{2}\|\mathbf{w}\| + C\sum_{i=1}^{l}\xi_i & (2) \\ \text{subject to} & y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i & (3) \\ \text{and} & \xi_i \geq 0,\ \forall i = 1,\dots,l. & (4) \end{array}$$

Here, $\xi = (\xi_1, \dots, \xi_l)$ corresponds to the slack variables ξ_i and C is a global parameter that has to be determined by the user. The bigger C, the easier training patterns may violate the constraint (3). By introducing the Lagrangian of the primal problem (2), we end up solving the dual

$$\begin{array}{lcr} \max_{\alpha} & \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,i'=1}^{l} y_i y_{i'} \alpha_i \alpha_{i'} \langle \mathbf{x}_i, \mathbf{x}_{i'}\rangle & (5) \\ \text{s.t.} & \sum_{i=1}^{l} y_i \alpha_i = 0 & (6) \\ \text{and} & 0 \leq \alpha_i \leq C,\ \forall i = 1,\dots,l. & (7) \end{array}$$

In practice, only few problems can be solved by a linear classifier. Hence the problem has to be reformulated in a nonlinear way. This is done by mapping the input space $\mathcal{X}$ to some high-dimensional feature space $\mathcal{H}$ by $\Phi : \mathcal{X} \mapsto \mathcal{H}$ where Φ satisfies Mercer's condition [10]. We can thus solve our nonlinear optimization problem linearly in $\mathcal{H}$ by computing the scalar product $K(x, x') = \langle \Phi(x), \Phi(x')\rangle$ which is called kernel. We simply replace the occurrence of the scalar product in (5) with a chosen kernel function. Finally, the discrimination function (1) becomes

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{l} y_i \alpha_i K(x, x_i) + b\right).$$

Hence only points x_i with a positive weight α_i will contribute to the above equation. Such a point is called support vector (SV).

To give some examples, let us have a look at the following two kernel functions[1]. First of all, we can apply the linear kernel

$$K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}'\rangle = \sum_{d=1}^{n} [\mathbf{x}]_d [\mathbf{x}']_d.$$

which performs the identical mapping $\Phi : \mathcal{X} \mapsto \mathcal{X}$. Second, kernel functions $K(\mathbf{x}, \mathbf{x}') = K(\|\mathbf{x} - \mathbf{x}'\|)$ generate radial basis functions, *e.g.* the Gaussian kernel

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right).$$

3. Fuzzy support vector machines

A fuzzy SVM has been proposed as extension to standard SVM. But before we actually come to the definition of an FSVM, let us mention that the term "fuzzy SVM" came up one year before the real FSVM formulation. In 2001, Inoue and Abe [11] already used the normalized distance $\langle \mathbf{w}, \mathbf{x}\rangle + b$ of a point $\mathbf{x}$ to the hyperplane expressed by $\mathbf{w}$ and b. Since this distance of a point inside of the margin lies in $[0, 1]$, it can be used to express a vague classification[2]. This is especially useful when dealing with more than two classes. This kind of fuzzy SVM [11] does have a right to exist. We will now introduce another kind of fuzzy SVM [4, 5] for which we are absolutely not sure about its usefulness.

The acquisition of data in most real-world applications is usually vague, uncertain and/or not complete. Therefore it might be good to embody the abstracted information by fuzzy sets. Especially SVMs seem to be quite sensitive to noise and points that were rather improbably drawn from the underlying data generating distribution. The only free parameter of an SVM is C which regularizes the penalty term in (2) and hence the classification error. This parameter is usually fixed for every input pattern during the training process. Prior to training, all patterns are treated the same. That might be crucial for the SVM due to outliers and noise. So, the learning machine may suffer from overfitting.

As a consequence, the concept of a fuzzy support vector machine (FSVM) has been introduced independently from two different research groups at the same time [4, 5]. In particular, a membership value μ_i is assigned to every training pattern x_i. Thus the training sample $\mathcal{S}$ is mapped to a fuzzy training sample

$$\mathcal{S}_f = \{(x_1, y_1, \mu_1), \dots, (x_l, y_l, \mu_l)\}$$

where the membership values for positive and negative class are denoted as μ_i^+ and μ_i^-, respectively. Both values are assigned independently.

Similar to the constrained optimization problem of (2), FSVM tries to optimize the same variables. However, it fuzzifies the penalty term containing the regularizer C. The optimal hyperplane using FSVM can obtained solving

$$\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\mu}} \tau(\mathbf{w}, \boldsymbol{\xi}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\| + C\sum_{i=1}^{l} \mu_i^m \xi_i \qquad (8)$$

subject to constraints (3) and (4) where $\boldsymbol{\mu} = (\mu_1, \dots, \mu_l)$ and m regularizes the fuzziness of the fuzzified penalty term. The dual problem for FSVM can be obtained by deriving the Lagrangian of (8) and hence only differs in constraining the α_i's: Maximize (5) subject to (6) and

$$0 \leq \alpha_i \leq C\mu_i^m, \quad \forall i = 1, \dots, l.$$

In order to apply FSVM, the membership values $\boldsymbol{\mu}$ have to be defined. In [4], the authors suggested to learn these values as follows. First they removed outliers and then fuzzified the remaining positive and negative instances independently by some membership functions. Finally, both sets were combined to $\mathcal{S}_f$.

[1] See [10] for a collection of kernel functions and further details on SVMs.

[2] A Bayesian interpretation of a probabilistic SVM output is given in [12].

It is important to note that the same optimization problem has been obtained in practice [13]. Here, the author neglected the global regularizer C in Equation (2). Instead, a set $\boldsymbol{C}$ of individual $C_i \in [0, C]$ with $i \in \{1, \ldots, l\}$ have been used which leads to

$$\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{C}} \tau(\mathbf{w}, \boldsymbol{\xi}, \boldsymbol{C}) = \frac{1}{2}\|\mathbf{w}\| + \sum_{i=1}^{l} C_i^m \xi_i. \qquad (9)$$

Hence the training instances were weighted by some user preferences. Thus prior knowledge about the importance of data points was included into the learning process. Setting each membership value μ_i^m to C_i/C, Equations (8) and (9) become identical. So, for $C_i = C$ Equation (9) is equivalent to Equation (2). With $C_i = C$ and $\mu_i^m = C_i/C$, we conclude that $\mu_i = 1$. Yet every standard C-SVM implementation can be extended to an FSVM by moving C into the sum of Equation 2 as C_i. So, we naturally question the usefulness of FSVMs. Why to fuzzify the slack variables ξ_i?! They anyway express the fuzziness of the hard border C through $\xi_i/C \in [0, 1]$.

4. Fuzzy rule extraction from SVM

The second issue we are going to talk about relates to the extraction of fuzzy rules from an SVM. As far as we know, there are four different ways to obtain fuzzy rules from an SVM. Grid-based approaches use predefined fuzzy partitions to define a fuzzy grid over the complete input-output space. The final rule base will consist of a subset of possible combinations of linguistic values. Approaches based on local granules do not use any predefined grid. The fuzzy rules are directly induced by the set of SVs. Hybrid approaches typically exploit clustering algorithms to reduce the number of possible rules. Last not least, the sets of positive and negative SVs might be viewed as one fuzzy rule, respectively. We call these approaches kernel-independent since the choice of the kernel is subordinate.

4.1. Grid-based approaches

The most prominent approach named FREx by Chaves *et al.* [14, 15, 16] selects the grid cells with maximum membership degree for every SV. Every SV thus corresponds to one fuzzy rule. FREx results in many fuzzy rules with the same antecedent. If the consequent is always consistent, then only one rule is stored. If not, then such conflict is resolved by choosing the rule with the highest coverage of data points. FREx usually leads to few rules.

Another grid-based algorithm can be found in [17]. Again, every SV represents one rule. The key idea is the same as it is for FREx. However, only fuzzy sets having a membership degree greater than $\beta \in [0, 1]$ are considered in a rule's antecedent. It is assumed that fuzzy rules should represent the prototypical points of the data distribution. The problem is that SVs are by nature very far away from any prototypes. They might be even located on the “wrong side”. Thus the obtained rules do not reflect the underlying data density. This can be shown by a simple thought experiment.

Consider the XOR problem and random data points generated from $[-1, 1]^2$ as it is discussed in [17]. Using three membership functions *low, medium, high* for both dimensions, it actually turns out that the SVs will activate *medium* absolutely the most in both dimensions. The obtained rules would thus model the most uninteresting part of the XOR distribution.

The chosen fuzzy partition is crucial to the success of all grid-based approaches. If the grid is chosen fine enough, then the approximation will be arbitrarily good (but at very high computational costs). A wrong choice of the grid may skip extrema. Also, with an increasing number of dimensions the number of possible rules is growing exponentially. That is why Chaves *et al.* suggest to use feature selection methods [16].

4.2. Local granulation

In high-dimensional input spaces, a global granulation leads to an exponential rule growth with increasing dimensionality. Individual fuzzy rules based on local granulation avoid this circumstance. Chen and Wang [18, 6, 19] were the first who showed that certain SVMs can be interpreted as zero-order TSK fuzzy rule-based classifiers (FRBCs). Simply, every SV relates to one fuzzy rule with individual membership functions. In general, we expect a better modeling of local system properties. Also, the number of rules is only bounded by the number of SVs, which is nice in high dimensions. However, the interpretability of fuzzy sets suffers severely since many overlapping membership functions are produced. Furthermore only certain types of fuzzy sets can be used, *i.e.* positive definite reference functions.

An attempt to use arbitrary fuzzy sets such as trapezoids or asymmetric triangles has been performed by the authors [20]. The trick is to use a generalized SVM [21] that can be solved by successive overrelaxation [22, 23]. Solving a different optimization problem, however, leads to worse classification accuracies [20].

Nevertheless, local granulation always lacks interpretation. Projecting all fuzzy sets onto one variable will usually not lead to meaningful linguistic values.

4.3. Hybrid approaches using clustering

Hybrid approaches are based on different learning algorithms. Juang *et al.* [24, 25, 26] proposed a self-organizing Takagi-Sugeno-type fuzzy network with support vector learning (SOTSFN-SV). They successfully applied it to several real-world datasets.

The rule antecedents are constructed through an online clustering method [27]. The original data space is transformed to a space of membership degrees that $\mathbf{x}$ belongs to the clusters $1, \ldots, k$. A linear SVM is used to determine the optimal consequent parameters. Although the number of rules k can be very low, clustering needs to be applied first.

4.4. Kernel-independent approach

In [28] a so-called λ-FRBC is directly formed from an SVM. The authors show that every SVM with $f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b\right) = \operatorname{sgn}(h(\mathbf{x}))$ equals a TSK FRBC

$$R_1 : \textbf{if } h(\mathbf{x}) \textbf{ is } I_{(0,\infty)} \textbf{ then } y = 1,$$
$$R_2 : \textbf{if } h(\mathbf{x}) \textbf{ is } I^*_{(0,\infty)} \textbf{ then } y = -1$$

with

$$I_{(0,\infty)}(a) = \begin{cases} 1 & \text{if } 0 < x < \infty \\ 0 & \text{if } -\infty < x < 0 \end{cases} \qquad (10)$$
$$I^*_{(0,\infty)}(a) = 1 - I_{(0,\infty)}(a).$$

So, the set of positive and negative SVs represent one fuzzy rule, respectively. Approximating (10) leads to $I_{(0,\infty)} \approx \frac{1}{1+\exp(-\lambda x)}$ and $I^*_{(0,\infty)} \approx \frac{1}{1+\exp(\lambda x)}$. For $\lambda \to \infty$, the proposed FRBC is obtained. Experiments with $\lambda \approx 20$ showed good results [28]. The clue here is that every SV corresponds to a fuzzy clause "distance between the SV and $\mathbf{x}$" in the antecedent. Depending on the kernel, the antecedent might be, *e.g.* a disjunction or conjunction. Naturally, all kernels are possible leading to only two rules. However, the high number of SVs still causes readability problems of both rules.

5. Recommendations

Regarding the FSVM, we generally recommend not to use FSVM based on the ideas of [4, 5]. The reason is simple. Statistical learning theory (including SVM) and FST are well-defined theories. The formulation of FSVM in (8), however, is somewhat hard to motivate from a statistical learning point of view. Fortunately, there exists a theoretically nice approach to fuzzy SVM based on a weighted margin [29]. Not surprisingly, Tao and Wang call their approach new fuzzy support vector machine (NFSVM). Their notion of a fuzzy support vector (FSV) actually coincides with a prototypical point. Whereas a standard SV is very close to the hyperplane and thus far from any prototype (or cluster center), an FSV "may be not" [29].

The problem of finding prototypical points in the data also counts when extracting fuzzy rules from an SVM. Using SVs only to obtain fuzzy rules instead of the complete training data imposes this problem naturally. No matter which way is used to obtain fuzzy rules by SV learning, SVs will never be prototypical. A pure renaming of them into "fuzzy rules" or mapping onto a fuzzy partition will not bring meaningful results. Reconstruction methods such as in [30] for crisp rules could be used. Such a heuristic method, however, would further complicate the rule generation process.

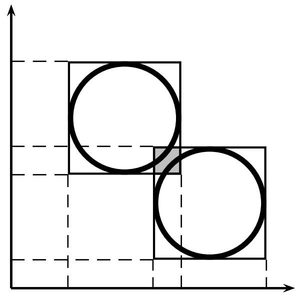

Figure 1: Information loss by projection. The rule shape of SVs is spherical. A projection of an SV might lead to intersecting hypercubes which contains the original hypersphere.

This is in line with Vapnik's idea of imperative learning [31]:

> "When solving a problem of interest, do not solve a more general problem as an intermediate step. Try to get the answer that you really need but not a more general one."

Although FRBCs are universal approximators, a practitioner in favor for FRBCs rather wants to understand data than being very accurate. On the other hand, an SVM is not suitable to interpret data. Its accuracy, however, is known to be outstanding.

Naturally, we question the sense of extracting fuzzy rules from SVs. We recommend to use simpler methods that output prototypical points(*e.g.* clustering approaches) or some kind of threshold-based rules (see *e.g.* [32]). In general, all but the last presented approach in Subsection 4.4 have the same disadvantage, *i.e.* the potential loss of information. This becomes clear when first cylindrically extend and then intersect the projected fuzzy sets to obtain the fuzzy rules. An exemplary loss of information is depicted in Figure 1. The additional information corresponds to the set of points that has not been covered by the SVM before. Another problem is the fact that an SV uses all attributes and so does its corresponding fuzzy rule. If certain clauses are not necessary in the antecedent parts of the rules, computationally costly feature selection methods or other heuristics can be applied.

6. Conclusions

We presented two existing ways to use FST for SVM. The former one relates to FSVM whereas the latter one deals with fuzzy rule extraction from an SVM. We discussed the usefulness of both recent trends in fuzzy methods in machine learning. We showed that an SVM is identical to a special type of an FSVM with $\mu_i = 1$. We categorized and critically analyzed many existing approaches to obtain fuzzy rules from SVMs. Finally, we questioned both trends and mentioned promising alternatives.

Acknowledgment

We thank the two anonymous reviewers for their valuable comments. They essentially helped to improve the understandability and reasoning of this paper by asking challenging questions.

References

[1] Vladimir Naumovič Vapnik. *Statistical Learning Theory.* Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons, Ltd., New York, NY, USA, September 1998.

[2] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, June 1965.

[3] Kai Michels, Frank Klawonn, Andreas Nürnberger, and Rudolf Kruse. *Fuzzy Control: Fundamentals, Stability and Design of Fuzzy Controllers.* Springer-Verlag, Berlin / Heidelberg, Germany, 2006.

[4] Han-Pang Huang and Yuan-Hung Liu. Fuzzy support vector machines for pattern recognition and data mining. *International Journal of Fuzzy Systems*, 4(3):826–835, 2002.

[5] Chun-Fu Lin and Sheng-De Wang. Fuzzy support vector machines. *IEEE Transactions on Neural Networks*, 13(2):464–471, 2002.

[6] Yixin Chen and James Z. Wang. Support vector learning for fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, 11(6):716–728, 2003.

[7] Jung-Hsien Chiang and Pei-Yi Hao. Support vector learning mechanism for fuzzy Rule-Based modeling: A new approach. *IEEE Transactions on Fuzzy Systems*, 12(1):1–12, February 2004.

[8] Pei-Yi Hao and Jung-Hsien Chiang. A fuzzy model of support vector regression machine. *International Journal of Fuzzy Systems*, 9(1):45–50, March 2007.

[9] Christian Moewes and Rudolf Kruse. Unification of fuzzy SVMs and rule extraction methods through imprecise domain knowledge. In José Luis Verdegay, Luis Magdalena, and Manuel Ojeda-Aciego, editors, *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU-08)*, pages 1527–1534, Torremolinos (Málaga), June 2008.

[10] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels – Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, December 2001.

[11] Takuya Inoue and Shigeo Abe. Fuzzy support vector machines for pattern classification. In *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN '01)*, volume 2, pages 1449–1454. IEEE Press, August 2001.

[12] John C. Platt. Probabilities for SV machines. In Peter L. Bartlett, Bernhard Schölkopf, Dale Schuurmans, and Alexander J. Smola, editors, *Advances in Large Margin Classifiers*, Neural Information Processing, pages 61–74. MIT Press, Cambridge, MA, USA, October 2002.

[13] Christian Moewes. *Application of support vector machines to discriminate vehicle crash events.* Diploma thesis, School of Computer Science, University of Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany, October 2007.

[14] Adriana da Costa F. Chaves, Marley Maria B. R. Vellasco, and Ricardo Tanscheit. Fuzzy rule extraction from support vector machines. In *Hybrid Intelligent Systems, 2005. HIS '05. Fifth International Conference on*, page 6 pp., 2005.

[15] Adriana da Costa F. Chaves, Marley Maria B. R. Vellasco, and Ricardo Tanscheit. Fuzzy rules extraction from support vector machines for multi-class classification. In *Analysis and Design of Intelligent Systems using Soft Computing Techniques*, volume 41 of *Advances in Soft Computing*, pages 99–108. Springer-Verlag, Berlin / Heidelberg, Germany, 2007.

[16] Adriana da Costa F. Chaves, Marley Maria B. R. Vellasco, and Ricardo Tanscheit. Fuzzy rules extraction from support vector machines for multi-class classification with feature selection. In *Advances in Neuro-Information Processing*, volume 5507 of *Lecture Notes in Computer Science*, pages 386–393. Springer-Verlag, Berlin / Heidelberg, Germany, 2009.

[17] Stergios Papadimitriou and Konstantinos Terzidis. Efficient and interpretable fuzzy classifiers from data with support vector learning. *Intelligent Data Analysis*, 9(6):527–550, 2005.

[18] Yixin Chen and James Z. Wang. Kernel machines and additive fuzzy systems: classification and function approximation. In *The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ '03*, volume 2, pages 789–795, 2003.

[19] Yixin Chen. Support vector machines and fuzzy systems. In *Soft Computing for Knowledge Discovery and Data Mining*, pages 205–223. Springer Science+Business Media, LCC., New York, NY, USA, 2008.

[20] Christian Moewes and Rudolf Kruse. Learning fuzzy rules with arbitrary reference functions using GSVM [abstract]. In *Programme and Abstracts of CFE 09 and ERCIM 09*, page 102, Limassol, Cyprus, October 2009. Abstract

no. E024.

[21] Olvi L. Mangasarian. Generalized support vector machines. In Alexander J. Smola, Peter L. Bartlett, Bernhard Schölkopf, and Dale Schuurmans, editors, *Advances in Large Margin Classifiers*, page 135–146. MIT Press, Cambridge, MA, USA, October 2000.

[22] Olvi L. Mangasarian and David R. Musicant. Successive overrelaxation for support vector machines. *IEEE Transactions on Neural Networks*, 10(5):1032–1037, 1999.

[23] Olvi L. Mangasarian and David R. Musicant. Data discrimination via nonlinear generalized support vector machines. In Michael C. Ferris, Olvi L. Mangasarian, and Jong-Shi Pang, editors, *Complementarity: Applications, Algorithms and Extensions*, volume 50 of *Applied Optimization*, pages 233–251. Springer, 2001.

[24] Chia-Feng Juang, Shih-Hsuan Chiu, and Shu-Wew Chang. A Self-Organizing TS-Type fuzzy network with support vector learning and its application to classification problems. *IEEE Transactions on Fuzzy Systems*, 15(5):998–1008, 2007.

[25] Chia-Feng Juang and Shen-Jie Shiu. Using self-organizing fuzzy network with support vector learning for face detection in color images. *Neurocomputing*, 71(16-18):3409–3420, October 2008.

[26] Chia-Feng Juang, Wen-Kai Sun, and Guo-Cyuan Chen. Object detection by color histogram-based fuzzy classifier with support vector learning. *Neurocomputing*, 72(10-12):2464–2476, June 2009.

[27] Chia-Feng Juang and Chin-Teng Lin. An On-Line Self-Constructing neural fuzzy inference network and its applications. *IEEE Transactions on Fuzzy Systems*, 6(1):12–32, February 1998.

[28] J. L. Castro, L. D. Flores-Hidalgo, C. J. Mantas, and J. M. Puche. Extraction of fuzzy rules from support vector machines. *Fuzzy Sets and Systems*, 158(18):2057–2077, September 2007.

[29] Qing Tao and Jue Wang. A new fuzzy support vector machine based on the weighted margin. *Neural Processing Letters*, 20(3):139–150, 2004.

[30] Haydemar Nú nez, Cecilio Angulo, and Andreu Català. Rule extraction from support vector machines. In *ESANN'2002 Proceedings - European Symposium on Artificial Neural Networks*, pages 107–112, Bruges, Belgium, April 2002.

[31] Vladimir Naumovič Vapnik. *Estimation of Dependences Based on Empirical Data.* Number 3816 in Information Science and Statistics. Springer, New York City, New York, USA, 2nd edition, 2006.

[32] Christian Moewes and Rudolf Kruse. Evolutionary fuzzy rules for ordinal binary classification with monotonicity constraints. In *Proceedings of the World Conference on Soft Computing (WConSC 2011)*, San Francisco, CA, USA, May 23–26, 2011. (accepted for presentation).