Theoretical and Applied Informatics
ISSN 1896–5334
Vol.24 (2012), no. 3
pp. 243–254
DOI: 10.2478/v10179-012-0015-2

Analysis of new method of initialisation of neuro-fuzzy systems with support vector machines

KRZYSZTOF SIMIŃSKI

Institute of Informatics, Silesian University of Technology
ul. Akademicka 16, 44-100 Gliwice, Poland
Krzysztof.Siminski@polsl.pl

Received 16 March 2012, Revised 2 October 2012, Accepted 5 October 2012.

Abstract: The correspondence between support vector machines and neuro-fuzzy systems is very interesting. The full equivalence for classification and partial for regression has been formally shown. The equivalence has very interesting implication. It is a base for a new method of initialization of neuro-fuzzy systems, ie. for creating of fuzzy rule base. The commonly used methods are based on reversion of item: the premises of fuzzy rules split input domain into region, thus premises of fuzzy rules can be elaborated by partition of input domain. This leads to three main classes of partition of input domain. The above mentioned equivalence results in new way of creating the rule base. Now the input domain is not partitioned, but the premises of fuzzy rules are extracted from support vector. The objective of the paper is to examine the advantages and disadvantages of this new method for creation of fuzzy rule bases for neuro-fuzzy systems.

Keywords: support vector machine, neuro-fuzzy system, classification, regression

1. Introduction

The objective of the paper is the research of equivalence of support vector machines (SVM) and neuro-fuzzy systems (NFS) and testing the possibilities that are created by that equivalence.

The neuro-fuzzy systems are known for good generalisation abilities. The crucial part of the NF system is the fuzzy rule base. It comprises fuzzy implications (or fuzzy rules). The premises of fuzzy rules split the input domain into regions. The creation of fuzzy rule bases is done in reversed order: first the input domain is split into subregions, then the subregions are then transformed into premises of fuzzy rules.

There are three main methods of domain partition [8,1]:

1. In *grid partition* the values of each attribute (dimension) are split into *a priori* established number of intervals. This method is vulnerable to curse of dimensionality – the number of regions is in exponential relation with number of dimensions. This leads to the supernumerosity of regions, what needs subsequent reduction [12,13].
2. The *scatter partition* is based on clustering. This approach avoids the curse of dimensionality, but has two main disadvantages. Firstly the number of clusters has to be proposed before the clustering procedure starts. Most often the data are clustered into various number of groups and then the best number is selected [5,9]. Secondly this partition may leave some area of the input domain that are not covered by any region.
3. The *hierarchical partition* splits he input domain into regions and if needed the regions are recursively split into subregion [11,17,18,14,15]. This approach joins the advantages of above mentioned method: number of clusters does not need to be proposed before partition and the are no areas uncovered by any rule.

The papers [2,4] analyses the equivalence between support vector machines (SVM) and neuro-fuzzy systems (NFS). The most important conclusion from this paper is the new method of creation of fuzzy rule base. This new method is based on different idea that above described methods. This method does not split input domain into regions, but creates premises of fuzzy rules basing of support vectors. The support vectors are selected from the presented data by SVM for classification or regression (respectively to the task). So this method is quite a new class of creation of fuzzy rule base for neuro-fuzzy systems. The above mentioned papers [2,4] list the advantages of this approach. The number of support vector (and number of fuzzy rules) is not greater than the number of data tuples in the data set. The SVM are capable of handling high dimensional tasks, so the paper [2] claims this is the way to avoid the curse of dimensionality.

The objective of this paper is the analysis of applicability of this very interesting method in neuro-fuzzy systems.

The paper is organised in the following way. First the fuzzy models and support vector machines for classification and regression are presented in sections 2 and 3. Sec. 4.1 introduces data sets and Sec. 4.2 discusses the elaborated results. Finally Sec. 5 presents conclusions.

2. Fuzzy models

The paper [2] analyses the zeroth order TSK [20,19] fuzzy system. This system uses the fuzzy rules of the form:

$$R_j : \text{IF } \underline{\mathbf{A}}_j^1 \text{ AND } \underline{\mathbf{A}}_j^2 \text{ AND } \ldots \text{ AND } \underline{\mathbf{A}}_j^n \text{ THEN } b_j, \tag{1}$$

where $\underline{\mathbf{A}}_j^k$ is the fuzzy set with membership functions μ_j^k. Fuzzy conjuction (AND) operator (T-norm) is product, fuzzy aggregation operator – adding and COA defuzzification. To avoid the situation in which the data example is not covered by any rule, the extra rule is added. This rule is always fired and the consequence of the rules is a constant value.

The membership function μ used in premises of fuzzy classifier is a function generated by location transformation of reference function. The reference function [6] is a function $a : \mathbb{R} \to [0, 1]$ non-increasing on $[0, \infty)$ for which $a(x) = a(-x)$ and $a(0) = 1$. Location transformation of reference function a generates membership function μ for location parameter v

$$\mu(x) = a(x - v). \tag{2}$$

The paper [2] lists examples of reference functions, among them the gaussian one. For gaussian function

$$\mu(x) = \exp\left(-\frac{(x - v)^2}{s^2}\right) \tag{3}$$

the extraction of parameters (core value v and fuzziness s) is needed. The core value is easily extracted from support vectors. The values of attributes of support vectors become cores (centres) of gaussian functions. The paper [4] states that fuzziness parameter for gaussian function is calculated in the way described in paper [21]. Unfortunately it is not obvious how the authors determine the desired values.

The theoretical considerations are accompanied by experiments. In [4] the SVM-NFS correspondence is discussed also for regression. Some solutions proposed in the paper are discutable.

3. SVM

The SVM can be used for initialization of fuzzy system both for classification and regression. Let $\mathbb{X} = \{(\underline{\mathbf{x}}_1, y_1), (\underline{\mathbf{x}}_2, y_2), \ldots, (\underline{\mathbf{x}}_X, y_X)\}$ denote the set of data tuples in form

$$(\underline{\mathbf{x}}, y) = [x_1, x_2, \ldots, x_A, y]. \tag{4}$$

For classification $y \in \{-1, +1\}$, for regression $y \in \mathbb{R}$. In classification the SVM tries to find the separation hyperplanes H^+ and H^- for which the distance between them is maximised. The hyperplanes are expressed by formulae:

$$H^+ : \langle \underline{\mathbf{w}}, \underline{\mathbf{x}}_i \rangle + w_0 = +1, \tag{5}$$

$$H^- : \langle \underline{\mathbf{w}}, \underline{\mathbf{x}}_i \rangle + w_0 = -1, \tag{6}$$

where $\langle \underline{\mathbf{w}}, \underline{\mathbf{x}} \rangle$ stands for results of function $k : \mathbb{R}^A \times \mathbb{R}^A \to \mathbb{R}$. It may be scalar product of vectors or some kernel function.

Maximising of distance between hyperplanes H^+ and H^- is equivalent to minimising of $\|\underline{\mathbf{w}}\|$ with constrains

$$y_i \left(\langle \underline{\mathbf{w}}, \underline{\mathbf{x}}_i \rangle + w_0 \right) \geqslant 1, \qquad i = 1, 2, \ldots, X, \tag{7}$$

where X is number of data tuples.

For regression the slack variables ξ^+ and ξ^- are introduced. The SVM approach minimises the fuction

$$L = \frac{1}{2} \|\underline{\mathbf{w}}\|^2 + C \sum_{i=1}^{X} \left(\xi_i^+ + \xi_i^- \right), \tag{8}$$

with constraints

$$y_i - \langle \underline{\mathbf{w}}, \underline{\mathbf{x}}_i \rangle - w_0 \leqslant \varepsilon + \xi_i^+ \tag{9}$$

$$-y_i + \langle \underline{\mathbf{w}}, \underline{\mathbf{x}}_i \rangle + w_0 \leqslant \varepsilon + \xi_i^- \tag{10}$$

$$\xi_i^+ \geqslant 0 \tag{11}$$

$$\xi_i^- \geqslant 0 \tag{12}$$

where ξ^+ and ξ^- are slack variables, $C > 0$ determines the tradeoff, ε corresponds to ε-insensitive loss function (Vapnik's linear loss function with ε-insensitivity)

$$E_\varepsilon(\Delta t) = \begin{cases} 0, & |\Delta t| \leqslant \varepsilon \\ |\Delta t| - \varepsilon, & |\Delta t| > \varepsilon \end{cases} \tag{13}$$

or

$$E_\varepsilon(\Delta t) = \max\left(0, |\Delta t| - \varepsilon\right). \tag{14}$$

Both for classification and regression the Langrange multipliers and quadratic programming are used to minimise the above mentioned values. For the support vectors the values of multipliers α are greater than 0, other vector (data tuples) have $\alpha = 0$. This means that only support vectors determine the hyperplanes.

4. Experiments

The SVM used for experiments is SVM-Light [7], the NFS used is ANNBFIS [5].

4.1. Data sets

In experiments we used 'Mackey-Glass', 'TwoSpirals' and 'Iris' data sets shortly described below.

4.1.1. 'Mackey-Glass'

The data sets represents chaotic time series generated by differential equation [10]:

$$\frac{dx}{dt}(t) = \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} - bx(t), \tag{15}$$

where $a = 0.2, b = 0.1$ and $\tau = 17$. The data tuples are created with template

$$\underline{\mathbf{x}} = \left[x\left(t_{-8}\right), x\left(t_{-7}\right), \ldots x\left(t_{0}\right), x\left(t_{1}\right)\right]. \tag{16}$$

The last item of the tuple $x\left(t_1\right)$ is the value predicted on the base of previous 9 items. The train data set is composed by 200 tuples $x(501) - x(700)$, the test set comprises tuples $x(701) - x(1000)$.

4.1.2. 'TwoSpirals'

'TwoSpirals' data set represents the widely known problem of two class classification. The spirals are created according to formula:

$$x(t) = r(t)\cos\alpha(t) \tag{17}$$

$$y(t) = r(t)\sin\alpha(t) \tag{18}$$

where

$$r(t) = r_0 + at. \tag{19}$$

The angle is calculated separately for $+$ and $-$ classes:

$$\alpha^{+}(t) = \Delta\alpha \cdot t, \tag{20}$$

$$\alpha^{-}(t) = \Delta\alpha \cdot t + \pi. \tag{21}$$

In our data set $r_0 = a = 2$, $\Delta\alpha = \frac{\pi}{16}$. The data set has 96 points in each spiral. Each spiral has 3 rotations. The Fig. 1 presents the 'TwoSpirals' data set.

4.1.3. 'Iris'

This commonly known data set consists of 150 samples of Iris plants divided into three classes. Each data tuple is represented by 4 attributes (sepal length, sepal width, petal length and petal width).

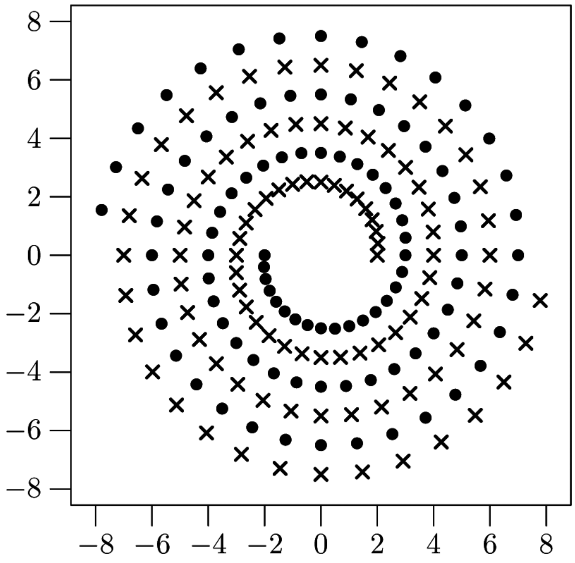

Fig. 1. ‘TwoSpirals’ data set. Points represented by dots are labelled with $-$, croses denote points labelled with $+$

4.2. Discussion of results

Paper [3] presents the results elaborated for the commonly used data set ‘Mackey-Glass’ with support vector regression with two values of insensitive parameter ($\varepsilon = 0.07|0.02$). The results cited from article [3] are presented in left part of Tab. 1. This results are suprisingly poor in two aspects. Firstly: the number of support vector (what implies the same number of rules in fuzzy rule base of NFS). Such high number of rules is totally incompatible with the idea of intelligibility of neuro-fuzzy systems. Secondly: the high error. For the same data set the results elaborated by ANNBFIS neuro-fuzzy system [5] for **five** rules (premises of rules are elaborated by clustering) are 0.000255948 for train data set and 0.000311285 for test data set (both values represent RMSE – root mean square error). The right part of Tab. 1 gathers the results elaborated by ANNBFIS system. The table contains the errors for 13 and 30 rules – these numbers of rules where taken from [4]. The fuzzy models created with clustering as a method of partition of input domain is much more accurate that model described in [4].

The best result is precission 98.38% with more than 35 rules. The ANNBFIS system achieves precision 97.33% (146 examples classified correctly out of 150) with only **three** rules. Further examples of fuzzy model created with SVM are present in [2,4,3]. These models also comprise high number of rules.

The system has two important parameters: insensitive parameter ε (cf. Eqq. 9 and 10) and tradeoff C (cf. Eq. 8). The manipulation with this values can influence to some

SVM-NFS [4]			number	ANNBFIS	
ε	train	test	of rules	train	test
			5	0.0002559	0.0003112
0.07	0.0505	0.0478	13	0.0004905	0.0011990
0.02	0.0134	0.0127	30	0.0003453	0.0035762

Tab. 1. The results elaborated for 'Mackey-Glass' data set. The left part of the table presents the results from [3] achieved with support vector machine, the right part – the results elaborated by ANNBFIS system with scatter partition of input domain.
The abbreviation RSME stands for 'root square mean error'.

precision	mean number of SV
95.46%	62.49
96.22%	47.5
96.38%	35.46
95.97%	28.695
95.55%	26.69

Tab 2. Precision of classification and number of support vectors published in [3]. Tradeoff $C = 100$

extend the number of rules. To examine this influence to experiments were executed for 'Mackey-Glass' and 'TwoSpirals' data sets.

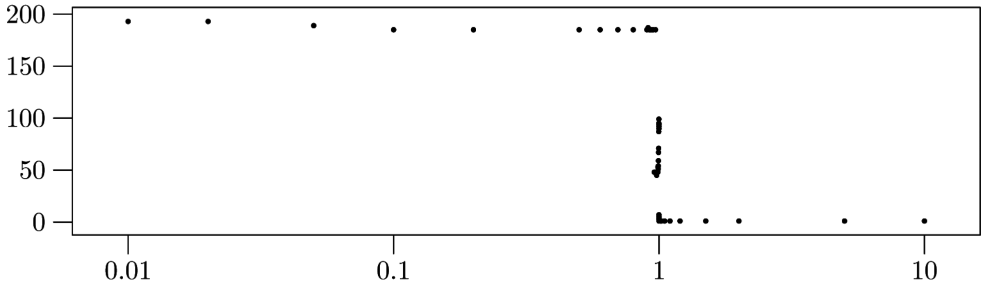

Fig. 2. Number of support vectors selected from 'TwoSpirals' data set in function of ε values

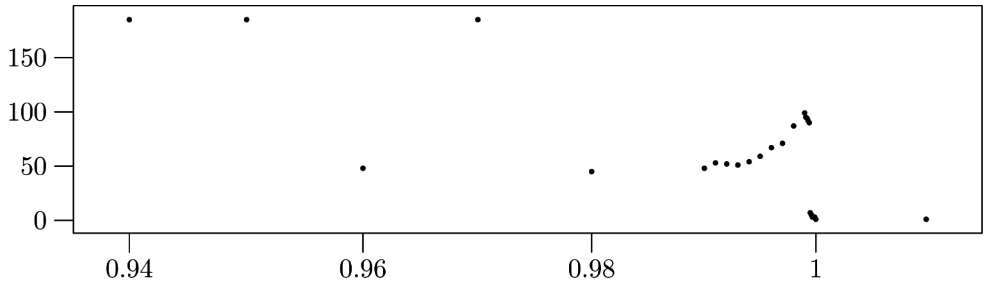

Fig. 3. Number of support vectors selected from 'TwoSpirals' data set in function of ε values. The figure presents the details of the Fig. 2

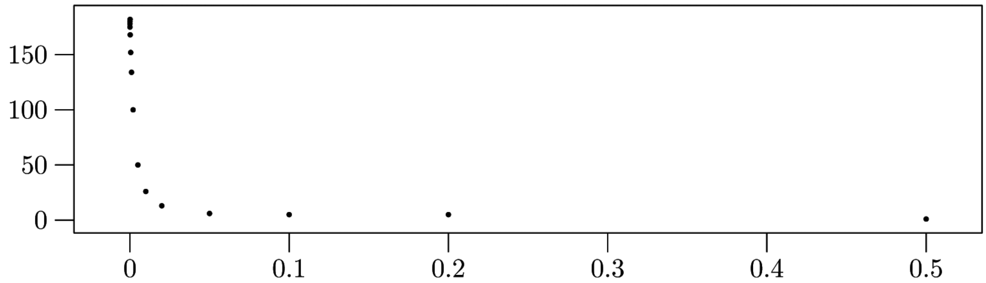

Fig. 4. Number of support vectors selected from 'Mackey-Glass' data set in function of ε values

The results reveal that the change of number of SV changes in a drastic way (for 'TwoSpirals' the critical value of ε is 1 and for 'Mackey-Glass' some low positive value). The relatively small change in values of ε triggers extreme change in number of selected support vectors – this feature is illustated by Figures 2, 3 for 'TwoSpirals' data set and Figure 4 for 'Mackey-Glass'. This feature can be regarded as a disadvantage of the method, because the manipulation with ε is very difficult. We get extreme values of number of SV's: either very low or unacceptably high. Manipulation with ε to get appropriate number of rules in fuzzy rule base, so that the system is precise enough and the rule base is intelligible.

The system is not very vulnerable to the changed of the second parameter – tradeoff C. The experiments show that changed of C does not lead to significant changes in number of SV. The paper [3] presents the results elaborated for 'Iris' data set which we cite in Tab. 3. Some other attempts have been done to examine the vulnerability to tradeoff C. This time for the value of ε where the number of selected SV's has some value between extreme ones. For this the value $\varepsilon = 0.995$ has been selected. The results are presented in Fig. 5. For 'Mackey-Glass' data set the influence of tradeoff on number of SV's for $\varepsilon = 0.01$ is shown in Fig. 6.

The number of support vectors is severely dependent on type of reference function used in system. The paper [3] presents results where number of SV's varies from 391 to 685.

tradeoff C	precision
100	96.81%
1000	96.61%
10000	96.45%

Tab 3. Number of support vectors for various values of tradeoff C parameter for 'Iris' data set. The results are taken from [3]. Numbers of support vectors are not provided

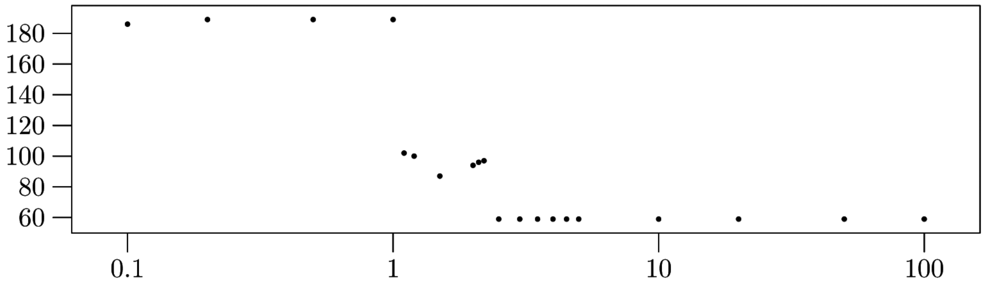

Fig. 5. Number of support vectors selected from 'TwoSpirals' data set in function of C; $\varepsilon = 0.995$

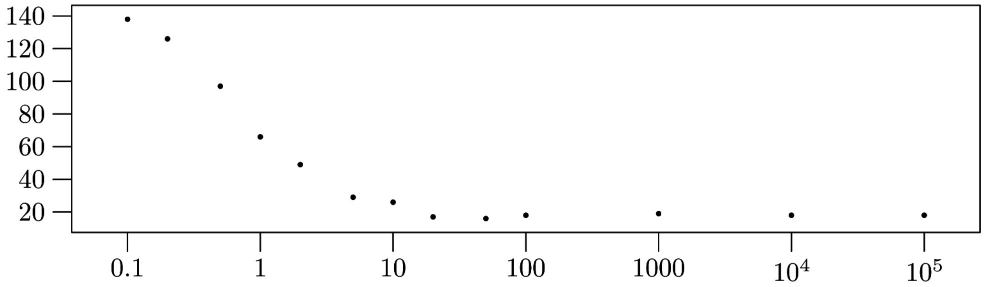

Fig. 6. Number of support vectors selected from 'Mackey-Glass' data set in function of C; $\varepsilon = 0.01$

5. Summary

The correspondence between support vector machines and neuro-fuzzy systems is very interesting. The full equivalence for classification has been shown in [3,2]. For regression the equivalence is only partial [3]. The equivalence has very interesting implication. It is a base for new method of initialization of neuro-fuzzy systems, ie. for creating of fuzzy rule base. The commonly used methods are based on reversion of obvious item: the premises of fuzzy rules split input domain into region, thus premises of fuzzy rules can be elaborated by partition of input domain. This leads to three main classes of partition of input domain. The above mentioned equivalence results in new way of creating the rule base. Now the input domain is not partitioned, but the premises of fuzzy rules are extracted from support vector. The paper [3] shows simple transformation of SV to premises of fuzzy rules (unfortuately this transformation is not described precisely in all aspects).

This new way of creating is very interesing and may lead to amelioration of results elaborated by NFS. Unfortunately this method has some very important disadvanteges:

1. The very large number of support vector what implies large number of rules. This makes the fuzzy rule base far from intelligible.
2. Although the rule base contain many rules, the results are significantly poorer then in neuro-fuzzy systems with scatter partition of input domain.
3. There is some possibility of reducing number of support vector for regression by precise tuning of insensibility parameter ε. But it is very difficult to tune the parameter without many experiments.
4. The tradeoff C seems to have lower influence on number of support vectors. This parameter is used in [3] for tuning the system. Our experience points that first the proper value of ε has to be found then C can be manipulated.
5. Creating of premises of fuzzy rules with SVM requires selection of kernel function for SVM.
6. The selection of membership functions for premises in fuzzy rules has been analysed, some remarks can be found in [16].

Acknowledgements

This work was supported by the European Union from the European Social Fund (grant agreement number: UDA-POKL.04.01.01-00-106/09).

References

1. A.M.R. Aguirre: *Sistema híbrido neuro-fuzzy-genético para mineração automática de dados*, Master's thesis, Pontifíca Universidade Católica do Rio de Janeiro, 2004.
2. Y. Chen: *Support vector machines and fuzzy systems*, In Oded Maimon and Lior Rokach, editors, Soft Computing for Knowledge Discovery and Data Mining, pages 205–223. Springer, 2008.
3. Y. Chen, J.Z. Wang: *Support vector learning for fuzzy rule-based classification systems*, IEEE Transactions on Fuzzy Systems, 11(6):716–728, 2003.
4. J.-H. Chiang, P.-Y. Hao: *Support vector learning mechanism for fuzzy rule-based modeling: a new approach*, IEEE Transactions on Fuzzy Systems, 12(1):1–12, 2004.

5. E. Czogała, J. Łęski: *Fuzzy and Neuro-Fuzzy Intelligent Systems*, Series in Fuzziness and Soft Computing. Physica-Verlag, A Springer-Verlag Company, Heidelberg, New York, 2000.

6. D. Dubois, H. Prade: *Operations on fuzzy numbers*, International Journal of Systems Science, 9(6):613–626, 1978.

7. Thorsten Joachims: *Making large-scale SVM learning practical*, In Bernhard Schölkopf, Christopher J.C. Burges, and A. Smola, editors, Advances in Kernel Methods – Support Vector Learning, Cambridge, MA, USA, 1999. MIT Press.

8. J. Łęski: *Neuro-fuzzy systems (in Polish: Systemy neuronowo-rozmyte)*, Wydawnictwa Naukowo-Techniczne, Warszawa, 2008.

9. J. Łęski, E. Czogała: *A new artificial neural network based fuzzy inference system with moving consequents in if-then rules and selected applications*, Fuzzy Sets and Systems, 108(3):289–297, 1999.

10. M.C. Mackey, L. Glass: *Oscillation and chaos in physiological control systems*, Science, 197(4300):287–289, July 1977.

11. O. Nelles, R. Isermann: *Basis function networks for interpolation of local linear models*, Proceedings of the 35th IEEE Conference on Decision and Control, 1:470–475, 1996.

12. O. Nelles, A. Fink, R. Babuška, M. Setnes: *Comparison of two construction algorithms for Takagi-Sugeno fuzzy models*, International Journal of Applied Mathematics and Computer Science, 10(4):835–855, 2000.

13. R. Senhadji, S. Sanchez-Solano, A. Barriga, I. Baturone, F.J. Moreno-Velo: *Norfrea: An algorithm for non redundant fuzzy rule extraction*, 2002 IEEE International Conference on Systems, Man and Cybernetics, 1:604–608, Oct. 2002.

14. K. Simiński: *Neuro-fuzzy system with hierarchical partition of input domain*, Studia Informatica, 29(4A (80)):43–53, 2008.

15. K. Simiński: *Patchwork neuro-fuzzy system with hierarchical domain partition*, In Marek Kurzyński and Michał Woźniak, editors, Computer Recognition Systems 3, volume 57 of Advances in Intelligent and Soft Computing, pages 11–18. Springer-Verlag, Berlin, Heidelberg, 2009.

16. K. Simiński: *Remark on membership functions in neuro-fuzzy systems*, In Krzysztof A. Cyran, Stanisław Kozielski, James F. Peters, Urszula Stańczyk, and Alicja Wakulicz-Deja, editors, Man-Machine Interactions, volume 59 of Advances in Intelligent and Soft Computing, pages 291–297. Springer Berlin/Heidelberg, 2009. 10.1007/978-3-642-00563-3_30.

17. F.J. de Souza, M.B.R. Vellasco, M.A.C. Pacheco: *Load forecasting with the hierarchical neuro-fuzzy binary space partitioning model*, Int. J. Comput. Syst. Signal, 3(2):118–132, 2002.

18. F.J. de Souza, M.M.R. Vellasco, M.A.C. Pacheco: *Hierarchical neuro-fuzzy quadtree models*, Fuzzy Sets and Systems, 130(2):189–205, 2002.

19. M. Sugeno, G.T. Kang: *Structure identification of fuzzy model*, Fuzzy Sets Syst., 28(1):15–33, 1988.
20. T. Takagi, M. Sugeno: *Fuzzy identification of systems and its application to modeling and control*, IEEE Trans. Systems, Man and Cybernetics, 15(1):116– 132, 1985.
21. L.X. Wang, J.M. Mendel: *Fuzzy basis functions, universal approximation, and orthogonal least-squares learning*, IEEE Transactions on Neural Networks, 3(5):807–814, 9 1992.

Analiza nowej metody inicjalizacji systemów neuronowo-rozmytych z wykorzystaniem maszyn wektorów wspierających

Streszczenie

Związek pomiędzy maszynami wektorów podpierających i systemami neuronowo-rozmytymi jest bardzo interesujący. Została wykazana pełna odpowiedniość między tymi systemami dla klasyfikacji i częściowa dla regresji. Odpowiedność ta ma bardzo ważną konsekwencję. Jest podstawa do opracowania nowego sposobu tworzenia bazy reguł dla systemu neuronowo-rozmytego. Dotychczasowe metody opierają się na podziale przestrzeni wejściowej, a następnie przekształcenia tak powstałych regionów w przesłanki rozmytych reguł. Tutaj możliwe jest przekształcanie wektorów wspierających na przesłanki reguł rozmytych. Celem artykułu jest przebadanie możliwości stosowania takiego podejścia do inicjalizacji systemów neuronowo-rozmytych. Eksperymenty wykazują dość istotną wadę tego podejścia. W jego wyniku powstają bardzo liczne zbiory reguł rozmytych, co zupełnie przeczy idei interpretowalności wiedzy w systemach neuronowo-rozmytych. Manipulacja pewnymi parametrami umożliwia zmiejszenie liczby reguł, jednak manipulacja ta jest trudna i wymaga wielu prób. Drugą dość istotną wadą jest wyraźnie wyższy błąd wypracowywany przez systemy inicjalizowane przez SVM w porównaniu do systemów, których bazy reguł tworzone są poprzez podział przestrzeni wejściowej.