Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics. Volume I: Foundations

Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics Volume II: Applications

Editors

Krassimir T. Atanassov
Michał Baczyński
Józef Drewniak
Janusz Kacprzyk
Maciej Krawczak
Eulalia Szmidt
Maciej Wygralak
Sławomir Zadrożny

Editors

Krassimir T. Atanassov
Władysław Homenda
Olgierd Hryniewicz
Janusz Kacprzyk
Maciej Krawczak
Zbigniew Nahorski
Eulalia Szmidt
Sławomir Zadrożny

SRI PAS IBS PAN

Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics Volume II: Applications

Systems Research Institute
Polish Academy of Sciences

Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics
Volume II: Applications

Editors

Krassimir T. Atanassov
Władysław Homenda
Olgierd Hryniewicz
Janusz Kacprzyk
Maciej Krawczak
Zbigniew Nahorski
Eulalia Szmidt
Sławomir Zadrożny

IBS PAN SRI PAS

Systems Research Institute
Polish Academy of Sciences
Newelska 6, 01-447 Warsaw, Poland
www.ibspan.waw.pl

ISBN 9788389475305

On time series envelopes for classification problems

Maciej Krawczak[1, 2] and Grażyna Szkatuła[1]

[1]Systems Research Institute, Polish Academy of Sciences
Newelska 6, Warsaw, Poland
[2]Warsaw School of Information Technology
Newelska 6, Warsaw, Poland
{krawczak, szkatulg}@ibspan.waw.pl

Abstract

In this paper we developed a technique of an envelopes approximation of time series, and an application of the envelopes for solving a classification problem of time series. Next, we introduced upper and lower envelopes as well as a way to compress them by a neural network. Obtained in such a way essential attributes were used to generate rules for time series classification. The obtained rules can classify the testing example with 100 % accuracy.

Keywords: time series, machine learning from examples, times series envelopes, essential attributes, heteroassociation, decision rules.

1 Introduction

In e.g. medicine, finance, industry, climate and so on there are considered time series, and we can face a classification problem of them. In literature one can fined some specialized algorithms dealing with such kind of problems, e.g. including decision trees (Rodriguez, Alonso [14]), neural networks (Nanopoulos, Alcock, Manolopoulos [1]), Bayesian classifiers (Wu, Chang [18]), etc.

Additionally we can observe an increasing interest in time series data mining e.g. in Xi et al. [19]).

In Krawczak and Szkatuła ([6]) it was shown that machine learning methods from examples creating decision rules of "IF … THEN …" type can be used for solving time series classification problems. Representation of class descriptions in the form of rules was considered more legible than other representations.

Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics. Volume II: Applications *(K.T.Atanassow, W.Homenda, O.Hryniewicz, J.Kacprzyk, M.Krawczak, Z.Nahorski, E. Szmidt, S. Zadrożny, Eds.), IBS PAN - SRI PAS, Warsaw, 2009.*

In this paper we considered a problem of time series classification on a base of an aggregated time series representation.

In this case we recalled the concept of time series envelopes and then introduced upper and lower envelopes of time series. In fact the upper and lower envelopes are piecewise approximations of time series envelopes. Next, we introduced aggregated forms of the upper and lower envelopes approximations of time series. The use of heteroassociative memory allowed us to obtain an essential attributes for time series representation. At the end decision rules of "IF … THEN …" type was generated to classify time series. For the calculation the Irvine University time series data benchmarks were used.

2 The concept of envelopes

Let us consider the following time series

$$\{x_k(n)\}_{k=1}^{k=K} = [x_1(n), x_2(n), \ldots, x_K(n)]^T \tag{1}$$

for $k = 1, 2, \ldots, K$, $n = 1, 2, \ldots, N$.

For each time series (1) we introduce the *m*-step upper and lower envelopes in the following manner.

Let us denote the upper envelope of (1) by $\{x_k^2(n)\}_{k=1}^{k=\left\lfloor \frac{K}{m} \right\rfloor m}$ described in the following way:

$$\begin{aligned}
&x_1^2(n) = \max\{x_1(n), x_2(n), \ldots, x_m(n)\}\\
&x_2^2(n) = \max\{x_1(n), x_2(n), \ldots, x_m(n)\}\\
&\ldots\\
&x_m^2(n) = \max\{x_1(n), x_2(n), \ldots, x_m(n)\}\\
&x_{m+1}^2(n) = \max\{x_{m+1}(n), x_{m+2}(n), \ldots, x_{2m}(n)\}\\
&x_{m+2}^2(n) = \max\{x_{m+1}(n), x_{m+2}(n), \ldots, x_{2m}(n)\}\\
&\ldots\\
&x_{2m}^2(n) = \max\{x_{m+1}(n), x_{m+2}(n), \ldots, x_{2m}(n)\}\\
&\ldots\\
&x_{\left\lfloor \frac{K}{m} \right\rfloor m-m+1}^2(n) = \max\{x_{\left\lfloor \frac{K}{m} \right\rfloor m-m+1}(n), x_{\left\lfloor \frac{K}{m} \right\rfloor m-m+2}(n), \ldots, x_{\left\lfloor \frac{K}{m} \right\rfloor m}(n)\}
\end{aligned} \tag{2}$$

$$x^2_{\left\lfloor \frac{K}{m} \right\rfloor m-m}(n) = \max\{x_{\left\lfloor \frac{K}{m} \right\rfloor m-m+1}(n), x_{\left\lfloor \frac{K}{m} \right\rfloor m-m+2}(n), \ldots, x_{\left\lfloor \frac{K}{m} \right\rfloor m}(n)\}$$

$$\ldots$$

$$x^2_{\left\lfloor \frac{K}{m} \right\rfloor m}(n) = \max\{x_{\left\lfloor \frac{K}{m} \right\rfloor m-m+1}(n), x_{\left\lfloor \frac{K}{m} \right\rfloor m-m+2}(n), \ldots, x_{\left\lfloor \frac{K}{m} \right\rfloor m}(n)\}$$

Similarly, we introduce *m*-step lower envelopes denoted by $\{x^3_k(n)\}_{k=1}^{k=\left\lfloor \frac{K}{m} \right\rfloor m}$:

$$
\begin{aligned}
&x^3_1(n) = \min\{x_1(n), x_2(n), \ldots, x_m(n)\} \\
&x^3_2(n) = \min\{x_1(n), x_2(n), \ldots, x_m(n)\} \\
&\ldots \\
&x^3_m(n) = \min\{x_1(n), x_2(n), \ldots, x_m(n)\} \\
&x^3_{m+1}(n) = \min\{x_{m+1}(n), x_{m+2}(n), \ldots, x_{2m}(n)\} \\
&x^3_{m+2}(n) = \min\{x_{m+1}(n), x_{m+2}(n), \ldots, x_{2m}(n)\} \\
&\ldots \\
&x^3_{2m}(n) = \min\{x_{m+1}(n), x_{m+2}(n), \ldots, x_{2m}(n)\} \\
&\ldots \\
&x^3_{\left\lfloor \frac{K}{m} \right\rfloor m-m+1}(n) = \min\{x_{\left\lfloor \frac{K}{m} \right\rfloor m-m+1}(n), x_{\left\lfloor \frac{K}{m} \right\rfloor m-m+2}(n), \ldots, x_{\left\lfloor \frac{K}{m} \right\rfloor m}(n)\} \\
&x^3_{\left\lfloor \frac{K}{m} \right\rfloor m-m}(n) = \min\{x_{\left\lfloor \frac{K}{m} \right\rfloor m-m+1}(n), x_{\left\lfloor \frac{K}{m} \right\rfloor m-m+2}(n), \ldots, x_{\left\lfloor \frac{K}{m} \right\rfloor m}(n)\} \\
&\ldots \\
&x^3_{\left\lfloor \frac{K}{m} \right\rfloor m}(n) = \min\{x_{\left\lfloor \frac{K}{m} \right\rfloor m-m+1}(n), x_{\left\lfloor \frac{K}{m} \right\rfloor m-m+2}(n), \ldots, x_{\left\lfloor \frac{K}{m} \right\rfloor m}(n)\}
\end{aligned}
\tag{3}
$$

The representation of the above envelopes idea can be visualized as attempt to piecewise constant approximations of the original time series.

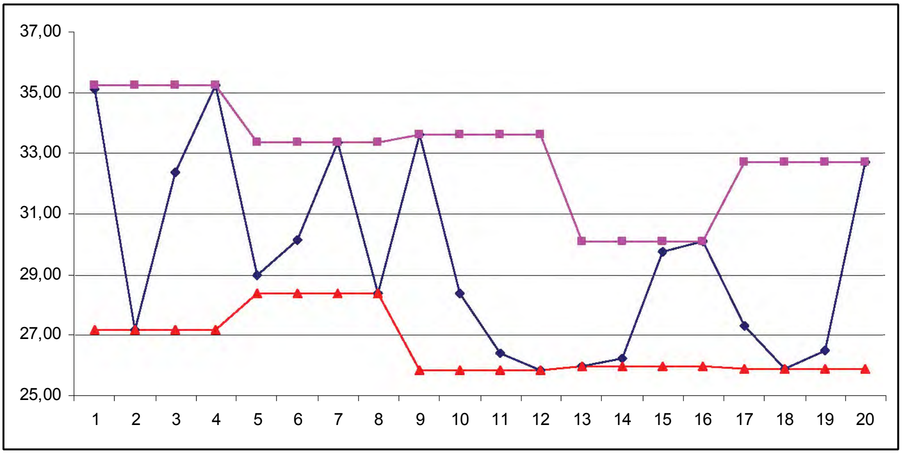


Figure 1: *4*-step upper and lower envelopes for the first 20 values of an exemplary time series

The main aim to introduce the envelopes is to reduce the dimensionality of the original time series. Namely, a given time series of arbitrary length K can be reduced to a string of arbitrary length $\left\lfloor \frac{K}{m} \right\rfloor$, where m denotes a number of sequent time series values.

For m-step upper envelopes denoted by

$$\left\{x_k^2(n)\right\}_{k=1}^{k=\left\lfloor \frac{K}{m} \right\rfloor m} = \left[x_1^2(n), x_2^2(n), \ldots, x^2_{\left\lfloor \frac{K}{m} \right\rfloor m}(n) \right]^T, \text{ for } n = 1, 2, \ldots, N$$

m succeeding equal values are treated as a one single value, and in result a new time series is obtained

$$\left\{x_k^G(n)\right\}_{k=1}^{k=\left\lfloor \frac{K}{m} \right\rfloor} = \left[x_1^G(n), x_2^G(n), \ldots, x^G_{\left\lfloor \frac{K}{m} \right\rfloor}(n) \right]^T \qquad (4)$$

in the following way

$x_1^2(n)$ $x_2^2(n)$ … $x_m^2(n)$	$\Rightarrow$	$x_1^G(n) = \max\{(x_1(n), x_2(n),...,x_m(n)\}$
$x_{m+1}^2(n)$ $x_{m+1}^2(n)$ … $x_{2m}^2(n)$	$\Rightarrow$	$x_2^G(n) = \max\{(x_{m+1}(n), x_{m+2}(n),...,x_{2m}(n)\}$
…		
$x^2_{\left\lfloor \frac{K}{m} \right\rfloor m-m+1}(n)$ $x^2_{\left\lfloor \frac{K}{m} \right\rfloor m-m}(n)$ … $x^2_{\left\lfloor \frac{K}{m} \right\rfloor m}(n)$	$\Rightarrow$	$x^G_{\left\lfloor \frac{K}{m} \right\rfloor}(n) = \{\max\{(x_{\left\lfloor \frac{K}{m} \right\rfloor m-m+1}(n),...,x_{\left\lfloor \frac{K}{m} \right\rfloor m}(n)\}$

In the case of the reduced lower envelopes

$$\left\{x_k^D(n)\right\}_{k=1}^{k=\left\lfloor \frac{K}{m} \right\rfloor} = \left[x_1^D(n), x_2^D(n), \dots, x^D_{\left\lfloor \frac{K}{m} \right\rfloor}(n)\right]^T \tag{5}$$

the procedure is similar.

The reduced envelopes are visualized in Figure 2.

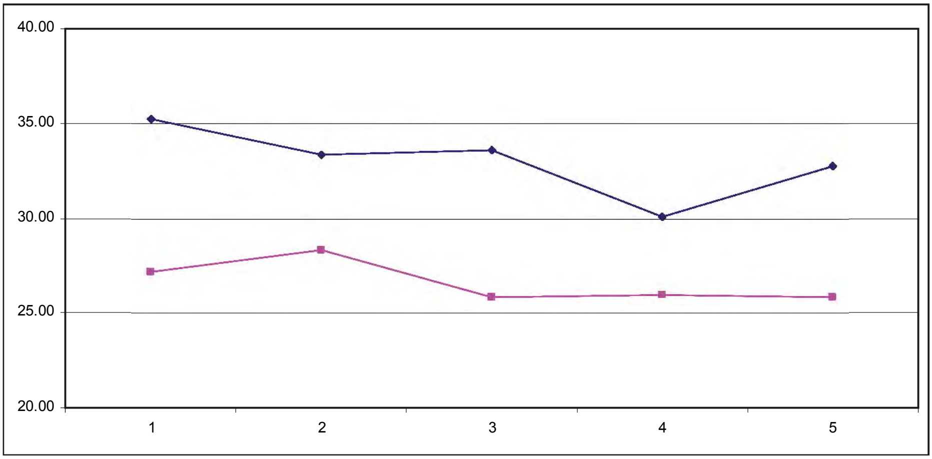


Figure 2: Aggregated *4*-step upper and lower envelopes
for the first 20 values of an exemplary time series

3 Essentials attributes generation

This section is devoted to compression of information about the class of time series involved in aggregated upper or lower envelopes.

For example, let the considered a time series or a part of it is described by

$$\left\{x_k^G(n)\right\}_{k=1}^{k=\left\lfloor \frac{K}{m} \right\rfloor} = \left[x_1^G(n), x_2^G(n), \ldots, x_{\left\lfloor \frac{K}{m} \right\rfloor}^G(n) \right]^T \tag{6}$$

then after the compression it is represented by q essentials attributes

$$\left\{y_i^G(n)\right\}_{i=1}^{i=q} = \left[y_1^G(n), y_2^G(n), \ldots, y_q^G(n)\right]^T, \tag{7}$$

where $q<<m$. It is assumed that using the vector y one can reconstruct the vector x as the following vector

$$\left\{\hat{y}_i^G(n)\right\}_{i=1}^{i=q} = \left[\hat{y}_1^G(n), \hat{y}_2^G(n), \ldots, \hat{y}_q^G(n)\right]^T, \tag{8}$$

with some accuracy. Here, for obtaining the essential attributes we used the heteroassociation memory implemented by feed-forward neural networks. The used neural network consists of two layers with one hidden layer, see (Krawczak, Szkatuła [6]).

In this we obtained an additional reduction of time series compare to the original one.

The real values of the essential attributes must be replaced by symbolic values. In order to do such replacement the ranges of the essential attributes are divided into some number of elements. Each element for any essential attribute is named by a different digital number, see (Krawczak, Szkatuła [6]).

After obtaining the symbolic values of the essential attributes values we can start to generate the rules of the following form

IF *some conditions are satisfied*
THEN *the time series belongs to a proper class*

The rules must guarantee some requirements, namely they should describe "all" or "almost all" examples which belong to the considered class as well as they should not describe "all" or "almost all" examples which do not belong to the considered class, additionally they should have minimal length in the sense of a number of conditions, and so on.

The decision rules created in such a way can after be used to classify other examples (time series) for which the class is not known.

4 Computational example

In order to verify the proposed hybrid system concept the data base available on the Irvine University of California was used (Alcock, Manolopoulos [1]):

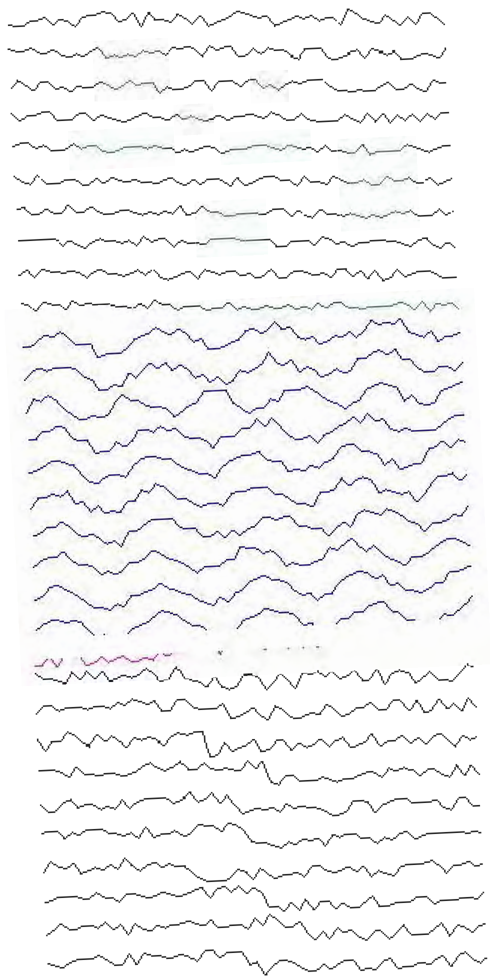
http://kdd.ics.uci.edu/databases/synthetic_control/synthetic_control.data.html

The data base contains artificial 600 time series, each consists of 60 values. The time series belongs to one of six class labeled by A, B, C, D, E, F. Series 1-100 belong to the lass A, 101-200 to the lass B, and so on, and 501-600 to the lass F.

For our purpose we chose three classes:

class 1: series B (25 learning examples)
class 2: series F (25 learning examples)
class 3: series A (25 learning examples)

Figure 3 shows exemplary time series (10 for each class).



Class A: normal

Class B: cycle

Class F: downward shift

Figure 3: Examples of three time series class

Our aim is to generate a set of decision rules for the three considered classes that could be used for classification of other not classified examples before.

We considered the following time series (showed in Figure 4): the first 25 examples belong to the class 1, the next 25 examples belong to the class 2 and the rest 25 examples to the class 3. Each time series has 60 values, as follows

$$\{x_k(n)\}_{k=1}^{k=K} = [x_1(n), x_2(n), \ldots, x_K(n)]^T, \ K = 60 \ \text{for} \ n = 1, 2, \ldots, 75$$

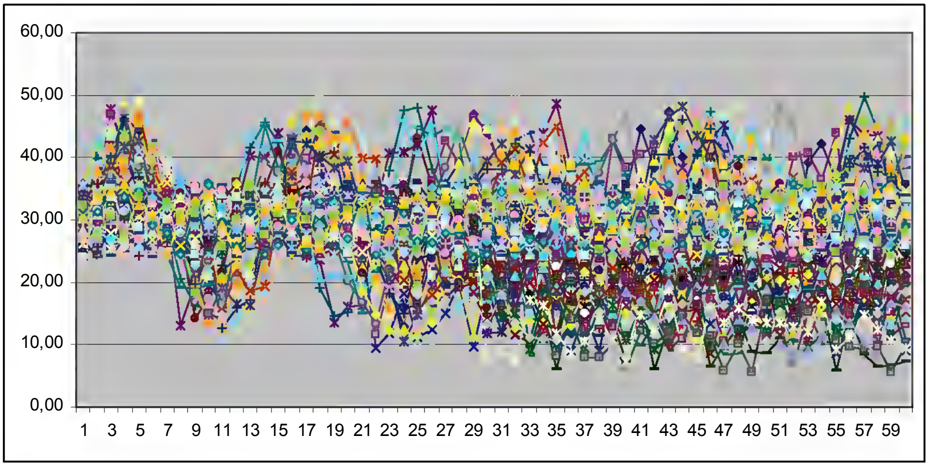


Figure 4: Time series belong to the class 1, 2 and 3

For each considered times series, $n = 1, 2, \ldots, 75$, each of length equal to 60 the *4*-step upper envelope was constructed.

Next, the obtained upper envelopes were aggregated, and in this way we obtained $\left[x_1^G(n), x_2^G(n), \ldots, x_{15}^G(n)\right]^T$, $n = 1, 2, \ldots, 75$, see Figure 5.

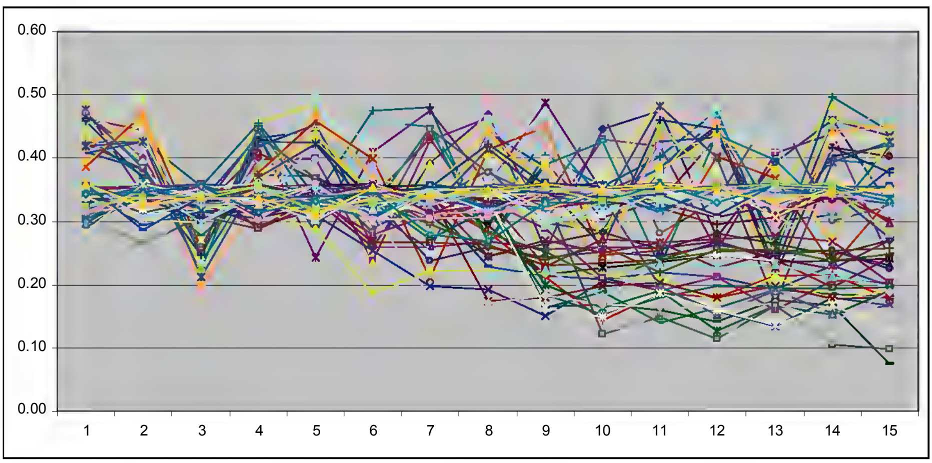
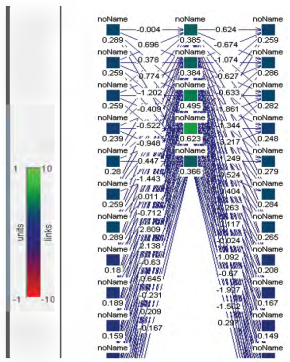
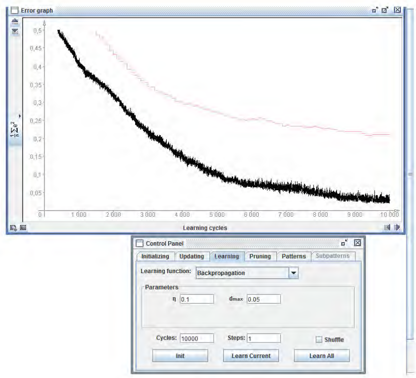


Figure 5: The aggregated *4*-step upper envelopes of time series belonging to the class 1, 2 and 3

In order to compress the time series a two layer feed forward neural network was applied. The network has 15 inputs and 15 outputs, and 5 neurons within the hidden layer. All neurons have a sigmoidal activation functions.

The neurons within the hidden layer create the essential attributes describing the aggregated upper envelopes and in results the original time series. The Java Neural Networks Simulator was used to find the essential attributes. The picture below shows the screen of the JNNS and the learning parameters on it.





The outputs of the hidden layer were multiplied by 1000 (Figure 6), and the nominalization of the essential attributes was arranged in the way shown in Table 1.

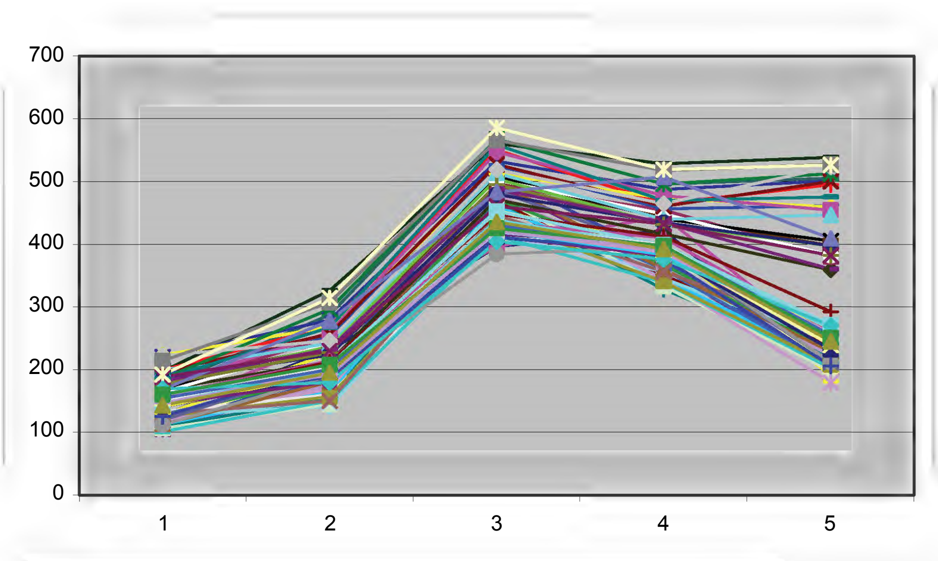


Figure 6: Value of essential attributes describing the time series belong to the class 1, 2 and 3

Table1: The nominalization of the essential attributes

Essential attributes value section	**Nominal value**
> 0 and <= 100	1
> 100 and <= 200	2
> 200 and <= 300	3
> 300 and <= 400	4
> 400 and <= 500	5
> 500 and <= 600	6
> 600 and <= 700	7
> 700 and <= 800	8
> 800 and <= 900	9
> 900 and <= 1000	10

Each example is represented by values of the attributes, in our case the essential attributes.

Now, the compressed and nominalized time series can now be treated as learning data for generation of decision rules in the following form

IF *some conditions are satisfied*
THEN *the time series belongs to a proper class*

The rules can be formed by applying various algorithms of machine learning from examples. Due to our experience, the problem was calculated with the Integer Programming method developed by Szkatuła [15, 16] and Kacprzyk and Szkatuła [3, 4, 5].

In our case, the conditional part of the rules was containing the conjunction of conditions related to the subset of the essential attributes.

The process of generating the decision rules based on the set examples under the assumption that for each class the examples have some common properties which distinguished them from another class.

An examples of the minimal set of rules obtained are shown below; number of described examples through rule is given for each case.

IF (a3=8) $\wedge$ (a5=3) THEN $class = B$ (12)
IF a4=5) THEN $class = B$ (8)
IF (a2=1) THEN $class = B$ (5)
IF (a3=6) THEN $class = B$ (3)
IF (a5=2) THEN $class = B$ (3)
IF (a3=9) $\wedge$ (a5=3) THEN $class = B$ (2)

IF (a5=6) THEN $class = F$ (9)
IF (a5=9) THEN $class = F$ (7)
IF (a5=8) THEN $class = F$ (6)
IF (a5=7) THEN $class = F$ (3)
IF (a2=2) ∧ (a3=7) ∧ (a4=6) THEN $class = A$ (17)
IF (a1=2) ∧ (a5=3) THEN $class = A$ (6)
IF (a2=3) ∧ (a5=4) THEN $class = A$ (4)

The generated decision rules correctly classified 100% time series belong to the class 1, 2 and 3.

5. Conclusions

In this paper we introduced a concept of upper and lower envelopes of time series. Next, such envelopes were aggregated, it allowed to reduce dimensionality of time series compare to original ones. Applying a simply feedforward neural network we could select essential attributes for the envelopes as well as indirectly for the considered time series.

At the end we generated a set of rules to classify exemplary time series. The approach requires much afford to prepare the original data in a accessible way for getting the envelopes as well as the normalization and the nominalization for getting the rules.

The computational example shows that the assumptions about the conception were proper and good. There was possible to reduce length of time series and lose much information. Even so there was conserve information about the time series characteristics. The accuracy of testing the procedure was 100 %.

Acknowledgements

The research was partially supported by the Ministry of Science and Higher Education (Poland) under Grant Nr N N519 384936.

References

[1] Alcock, R. J., Manolopoulos, Y. (1999). Time-Series Similarity Queries Employing a Feature-Based Approach. *7^{th} Hellenic Conference on Informatics*, Ioannina, Greece.

[2] Benedikt, L., Kajic, V., Cosker, D., Marshall, D., Rosin, P. L. (2008). Facial Dynamics in Biometric Identification. In *Proc. of British Machine Vision Conference*, Leeds, 2008.

[3] Kacprzyk J., Szkatuła G. (1998). *IP1* - An Improved Inductive Learning Procedure with a Preprocessing of Data. *Proceedings of IDEAL'98 (Hong Kong)*, Springer-Verlag.

[4] Kacprzyk J., Szkatuła G. (1999). An inductive learning algorithm with a preanalysis data. *International Journal of Knowledge - Based Intelligent Engineering Systems*, vol. 3, 135-146.

[5] Kacprzyk J., Szkatuła G. (2002). An integer programming approach to inductive learning using genetic and greedy algorithms. W: L.C. Jain and J.Kacprzyk (eds.) *New learning paradigms in soft computing*. Studies in Fuzziness and Soft Computing. Physica-Verlag Heidelberg, 323 - 367.

[6] Krawczak, M., Szkatuła G. (2008) On decision rulet application to time series classification. In: Atanassov K.T., Hryniewicz O., Kacprzyk J., Krawczak M., Nahorski Z., Szmidt E., Zadrożny S.: Advances in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics,Ac. Publ. House EXIT.

[7] Kumar N., Lolla N., Keogh E., Lonardi S., Ratanamahatana C., Wei L. (2005) Time-Series Bitmaps: A Practical Visualization Tool for Working with Large Time Series Databases. In proceedings of SIAM International Conference on Data Mining (SDM '05), Newport Beach, CA, April 21-23, 2005.

[8] Lin, J., Keogh, E., Lonardi, S., Chiu, B. (2003). A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. *Proceedings Data Mining and Knowledge Discovering*, San Diego.

[9] Nanopoulos, A., Alcock, R., & Manolopoulos, Y. (2001). Feature-based Classification of Time-series Data. *International Journal of Computer Research*, 49-61.

[10]Oja E. (1992). Principal components, minor components and linear neural networks. *Neural Networks*, vol.5, 927-935.

[11] Pawlak Z. (1982): Rough Set. *International Journal of Computer and Information Sciences*, Vol. 11, No 5, 341-356.

[12] Pawlak Z. (1991). *Rough Set. Theoretical Aspect of Reasoning about Data*. Kluwer Academic Publishers.

[13] Roddick J. F., Hornsby K., Spilopoulos M. (2001). An updated bibliography of temporal, spatial and spatio-temporal data mining research. In *Proceedings of the International Workshop on Temporal, Spatial and Spatio-Temporal data Mining*. Berlin, Springer, Lecture Notes in Artificial Intelligence, 147-163.

[14] Rodríguez, J.J., Alonso, C.J. (2004). Interval and dynamic time warping-based decision trees. In *Proceedings of the 2004 ACM symposium on applied computing* (SAC), 548-552.

[15] Szkatuła G. (1995). Machine learning from examples under errors in data, Ph.D. thesis, SRI PAS Warsaw, Poland.

[16] Szkatuła G. (2002). *Zastosowanie zmodyfikowanego zadania pokrycia w uczeniu maszynowym*. W: Gutenbaum J. (eds.): Automatyka Sterowanie Zarządzanie. SRI PAS, Warszawa, 431-445.

[17] Wei, L., Keogh, E. (2006). Semi-Supervised Time Series Classification. In *Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006),* pp. 748 - 753, Philadelphia, PA, U.S.A., August 20-23, 2006.

[18] Wu, Y. & Chang, E.Y. (2004): Distance-function design and fusion for sequence data. CIKM '04, 324-333.

[19] X. Xi, E. J. Keogh, C. R. Shelton, L. Wei, and C. A. Ratanamahatana. Fast time series classification using numerosity reduction. In ICML, 2006.

Appendix

The task of machine learning from examples can be formulated as follows: we have a finite set of examples *U*, called the learning set, and a finite set of attributes $A = \{a_1,...,a_K\}$. The set $V_{a_j} = \{v_{j,1}, v_{j,2}, ..., v_{j,L_j}\}$ is a domain of the attribute a_j, $j = 1,...,K$, where L_j denotes number of values of the *j*-th attribute. $V = \bigcup_{j=1,...,K} V_{a_j}$. $f: U \times A \to V$ is a function such that $f(e^n, a_j) \in V_{a_j}$ for every $a_j \in A$ and $e^n \in U$, $n = 1, 2, ..., N$.

Each example $e^n \in U$ is described by *K* attributes $A = \{a_1,...,a_K\}$, and is represented by

$$e^n = \bigwedge_{j=1}^{K} [a_j = f(e^n, a_j)] \qquad \text{(A1)}$$

where $f(e^n, a_j) = v_{j,t(j,n)}$, and $v_{j,t(j,n)} \in V_{a_j}$.

Function $f(e^n, a_j)$ denotes that the attribute a_j taking on a value $v_{j,t(j,n)}$ for example e^n.

The index $t(j,n)$ for $j \in \{1,2,...,K\}$ and $n \in \{1,2,...,N\}$ specifies which value of the *j*-th attribute is used in the *n*-th example.

An example e^n in (A1) is composed of *K* ”attribute-value” pairs (*selectors*), denoted $s_j = [a_j = v_{j,t(j,n)}]$.

Conjunction of $l \le K$ ”attribute-value” pairs, i.e.

$$\underset{j \in I}{\wedge} s_j = C^I \tag{A2}$$

where $I \subseteq \{1,...,K\}$, $card(I) = l$ is called a *complex*.

A complex C^I *covers* an example e^n if all the conditions on attributes given as *j*- th selectors are covered by (equal to) the values of the respective attributes in the example, $\forall j \in I$.

Suppose that we have a decision attribute a_d, where $\{a_d\} \cap A = \varnothing$ and $V_{a_d} = \{v_{d,1}, v_{d,2}, ..., v_{d,L_d}\}$ is a domain of the attribute a_d.

We can perform the partition of the entire set of examples into the disjoint classes with respect to the values taken by this attribute. The elements of the set *A* are referred to as *conditional attributes*. We assume that the number and character of attributes are sufficient for the correct split of examples belonging to different classes.

Thus, we have the sets $\{Y_{v_{d,l}} : l = 1,...,L_d\}$, where $Y_{v_{d,l}} = \{e \in U : f(e, a_d) = v_{d,l}\}$, $\forall v_{d,l} \in V_{a_d}$ and $Y_{v_{d,1}} \cup ... \cup Y_{v_{d,L_d}} = U$, $Y_{v_{d,i}} \cap Y_{v_{d,j}} = \varnothing$ for $i \ne j$.

Thus, the decision attribute splits the set of examples into the non-empty, disjoint and exhaustive subsets, that we call the *decision classes*.

The sets of the learning examples determined in this manner along with their division into classes, are the starting point in the process of machine learning, which is supposed to lead to the descriptions of the classes considered.

The process of formation of a class description on the basis of the set of examples having certain common properties, which distinguish a given class from the other ones, is characterized by the adopted language of data representation and the applied algorithm of machine learning.

These descriptions can be represented either in the form of rules of “IF *certain conditions are fulfilled* THEN *membership in a definite class takes place*” type; or in the form of decision trees; or in the form of the appropriately selected connection weights in neural networks and their structure.

Representation of class descriptions in the form of rules is considered more legible than other representations, and so in this paper classes are modeled in the form of rules.

In our case, the conditional part of the rules will contain the conjunction of conditions related to the subset of attributes selected for the description of the examples.

An implication $R_k:\ C^{I_k} \rightarrow [a_d = v_{d,l}],\ \ l \in \{1,...,L_d\}$ is called the *k*-th elementary rule for the class $Y_{v_{d,l}}$, where $C^{I_k} = \underset{j \in I_k}{\wedge} [a_j = v_{j,t(j,k)}]$ is description of example in terms of condition attributes a_j, $j \in I_k$, $I_k \subseteq \{1,...,K\}$ and this example belongs to class $Y_{v_{d,l}}$.

The rules, mentioned above, can be formed by applying various algorithms of machine learning from examples.

Due to our experience, the problem was calculated with the method developed by Szkatuła and Kacprzyk [3, 4, 5, 15, 16], which creates the rules successively for each class. Each rule satisfies weakened requirements, i.e. must correctly describe "all" or "almost all" the examples belonging to a class and do not describe "all" or "almost all" of the examples not belonging to this class. They should have minimum „length" (e.g. in terms of the number of conditions forming them).

The papers presented in this Volume 2 constitute a collection of contributions, both of a foundational and applied type, by both well-known experts and young researchers in various fields of broadly perceived intelligent systems.
It may be viewed as a result of fruitful discussions held during the Eighth International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets (IWIFSGN-2009) organized in Warsaw on October 16, 2009 by the Systems Research Institute, Polish Academy of Sciences, in Warsaw, Poland, Centre for Biomedical Engineering, Bulgarian Academy of Sciences in Sofia, Bulgaria, and WIT – Warsaw School of Information Technology in Warsaw, Poland, and co-organized by: the Matej Bel University, Banska Bistrica, Slovakia, Universidad Publica de Navarra, Pamplona, Spain, Universidade de Tras-Os-Montes e Alto Douro, Vila Real, Portugal, and the University of Westminster, Harrow, UK:

http://www.ibspan.waw.pl/ifs2009

The Eighth International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets (IWIFSGN-2009) has been meant to commence a new series of scientific events primarily focused on new developments in foundations and applications of intuitionistic fuzzy sets and generalized nets pioneered by Professor Krassimir T. Atanassov. Moreover, other topics related to broadly perceived representation and processing of uncertain and imprecise information and intelligent systems are discussed.

We hope that a collection of main contributions presented at the Workshop, completed with many papers by leading experts who have not been able to participate, will provide a source of much needed information on recent trends in the topics considered.

ISBN-13 9788389475305
ISBN 838947530-8
9 788389 475305