International Journal of Computational Methods
Vol. 13, No. 3 (2016) 1650018 (26 pages)

DOI: 10.1142/S0219876216500183

World Scientific
www.worldscientific.com

A Complete Algorithm for the Reduction of Pattern Data in the Classification of Interval Information

Piotr A. Kowalski* and Piotr Kulczycki†

AGH University of Science and Technology
Faculty of Physics and Applied Computer Science
Division for Information Technology and Biometrics
al. Mickiewicza 30, 30-059 Cracow, Poland

and

Polish Academy of Sciences, Systems Research Institute
ul. Newalska 6, 01-447 Warsaw, Poland
**pkowal@agh.edu.pl; pakowal@ibspan.waw.pl*
†*kulczycki@agh.edu.pl; kulczycki@ibspan.waw.pl*

Received 27 February 2015
Revised 16 November 2015
Published 22 January 2016

The aim of this paper is to present a novel method of data sample reduction that can be applied, in particular, to the classification of interval type imprecise information. Its concept is based on the sensitivity method, inspired by artificial neural networks, while the goal is to increase the number of apposite classifications, and, consequently, to increase calculation speed. As evident in this paper, the use of reduction algorithm eliminates the particular elements of all data sample patterns which have insignificant or negative influence on the correctness of classification. The methodology was tested on pseudo-random and real data, as well as by way of comparative analysis with similar task algorithms. The presented procedure was also tested for use in situations in which the data sample representing the individual classes had been obtained by the *k-means* clustering procedure.

Keywords: Data sample reduction; sensitivity method for artificial neural networks; data analysis; classification of imprecise information; interval data.

1. Introduction

In current literature, a reducing algorithm dedicated to the issue of the classifying interval type information practically does not exist. Recently, interest in interval analysis [Drakengren and Jonsson (1997)] has, however, grown, notably amongst individuals pushing the boundaries of practical procedure applications [Jaulin *et al.* (2001)]. For this reason, the following methodology was designed using advanced algorithms based on statistical and computational intelligence methods. The basis here is the assumption that the only accessible information about the investigated

*Corresponding author.

quantity $x \in \mathbb{R}$ is that it fulfils the condition $\underline{x} \leq x \leq \overline{x}$, and consequently, can be treated as the interval $[\underline{x}, \overline{x}]$.

Classification of imprecise information of an interval type was firstly described in Zhao *et al.* [2005], in which the authors used the SVM procedure [Duda *et al.* (2000)]. In this algorithm, information containing interval type data was treated as an additional attribute type of the material under consideration (as a supplement to the classical continuous-type data). As presented in the cited papers, the algorithm was described very briefly and rather seems to be used as a means of indicating new complications than solving the set problem. Another relatively simple method for classifying interval type information is the procedure for patterns counting, which consists of reckoning how many elements of the learning sample are contained in the interval which is under consideration [Kulczycki and Kowalski (2011)].

The main subject of the research presented here is the derivation of a reduction data sample for application within classification procedures. The tested element is given as an interval vector, but the data representing each class consists of elements defined uniformly e.g., deterministic when applying common statistical methods [Gil and Hryniewicz (2009)] or sharp when utilizing fuzzy logic [Kacprzyk (1997)]. A classification procedure [Duda *et al.* (2000); Ghost *et al.* (2006)] has been worked out for reducing samples inclusive of elements which have negligible or even negative influence on the correctness of classification. Its concept is based on the sensitivity method, inspired by artificial neural networks, while the goal is to increase the number of apposite classifications, as well as, in consequence, to enhance calculation speed. The concept of classification is based on the Bayes approach, ensuring a minimum of potential losses arising from misclassification. For a such-formulated problem, the methodology of statistical kernel estimators is used, as this frees the investigated procedure from arbitrary assumptions with respect to the 'shapes' of the utilized samples sets.

A reduction of the sample size can be achieved by the way of the employment of a broad spectrum of diverse kinds of methods. The first such group of algorithms is based on the idea of random reduction, which consists of rejecting in advance a certain percentage of elements from the sample set [Han and Kamber (2006)]. Another group of methods is built upon the notion of *k-nearest neighbors* (*k-NN*), as introduced in the work by Pal and Mitra [Mitra *et al.* (2002)]. These methods assume the replacement of some elements of data set components by other more representative sample elements. Another example of such recognized procedures is directly related to the main methodology used in the employment of the principal algorithm. As to the use of statistical kernel estimators, the reduction based on *Weighted Parzen Windows* (WPW) can be included [Babich and Camps (1996)]. In this procedure, in order to reduce sample size, the algorithm uses information about the value of the probability density function for each data, as derived from a learning sample set. In so doing, during the reduction process, data points are weighted on a sliding basis and the least interesting deleted. The last group of now commonly utilized reduction procedures are that of the use of algorithms based

on computational intelligence methods which are dedicated for employment within data mining procedures [Lukasik and Kulczycki (2011)].

The task of classifying an interval information set as based on a precise data pattern set, can be interpreted illustratively with the example wherein the patterns present actual, precisely measured quantities, while the intervals being classified represent uncertainties and imprecision in plans, estimations or measurements that are difficult to make. In particular, such pattern sets may consist of very accurate measurements (in which errors are practically ignored), while the classified interval constitutes a measurement taken from another, much less accurate apparatus or carried out in much worse conditions. Another example of the application of this kind of classification is the possibility of treating precise data as actual information from the past, e.g. temperature or currency exchange rates, while the classified element represents a prognosis, which by nature, is limited in precision. In these cases, the cardinality of the pattern data is often very large. Therefore, it is advisable to use an intelligent method to reduce the unnecessary or redundant components inside the pattern sets.

This paper is organized as follows. The first part of the paper is devoted to the description of the investigated methodology. In the next section, certain preliminary aspects of the techniques under consideration are presented. The first part of this section is concentrated upon the statistical kernel estimation methodology used in this work, after this, a short description of a classification algorithm is to be introduced. In the following section of this paper, neural network sensitive analysis is put forward. In the subsequent part, a data reduction algorithm will be described, then the numerical results based on benchmark and synthetic data sets as generated by random value generators are to be shown. Finally, some concluding remarks with respect to the presented approach are set out.

The preliminary version of this investigation was presented as a conference short paper by Kowalski and Kulczycki [2010].

2. Classification of Imprecise Information

Initially, within this section, statistical kernel estimators methodology will be shortly described. As shown, this method is useful as an algorithm for pattern reduction, as well as a procedure for classification. In the second part of this section, Bayes classification of interval information will be presented.

2.1. *Statistical kernel estimators*

Consider an n-dimensional random variable, with a distribution having the density f. Its kernel estimator $\hat{f}\colon \mathbb{R}^n \to [0,\infty)$ is based on the m-elements data sample $x_1, x_2, \ldots, x_m$ and can be defined as

$$\hat{f}(x) = \frac{1}{mh^n} \sum_{i=1}^{m} K\left(\frac{x - x_i}{h}\right), \tag{1}$$

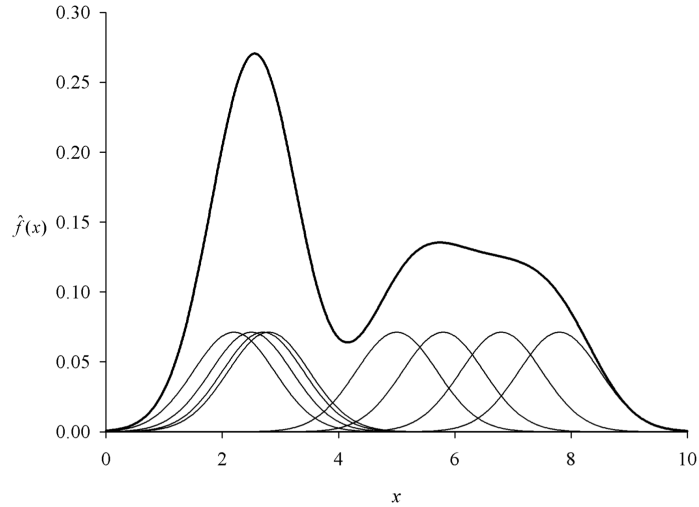


Fig. 1. Kernel estimator.

where the positive coefficient h is called a 'smoothing parameter'. The function $K(x)\colon \mathbb{R}^n \to [0,\infty)$, measurable, symmetrical relative to zero, with a weak global maximum at this point, and fulfilling the condition $\int_{\mathbb{R}^n} K(x)\,dx = 1$, is known as a 'kernel'.

The interpretation of the above definition is illustrated in Fig. 1 for a one-dimensional random variable and a 8-element pattern sample.

In this approach, the generalized (one-dimensional) Cauchy kernel

$$K(x) = \frac{2}{\pi(x^2+1)^2} \tag{2}$$

will be used. Within a multidimensional case, the function (2) will be generalized by the product kernel notion

$$K(x) = K\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}\right) = \mathcal{K}(x_1)\cdot\mathcal{K}(x_2)\cdot\ldots\cdot\mathcal{K}(x_n), \tag{3}$$

where $\mathcal{K}$ denotes the one-dimensional Cauchy kernel described above.

The value of the smoothing parameter h can be obtained by way of a very effective algorithm called the 'plug-in method' [Wand and Jones (1995)]. In a n-dimensional case, the parameter must be denoted as $h_1, h_2, \ldots, h_n$ and calculated separately for each dimension. Accordingly, the smoothing parameter is then given as

$$h = \left[\frac{W(\mathcal{K})}{U(\mathcal{K})^2}\,\frac{8\sqrt{\pi}}{3m}\,\hat{\sigma}^9\right]^{\frac{1}{5}}, \tag{4}$$

where

$$W(\mathcal{K}) = \int_{R^n} \mathcal{K}(x)^2 dx, \tag{5}$$

$$U(\mathcal{K}) = \int_{R^n} x^2 \mathcal{K}(x) dx, \tag{6}$$

while $\hat{\sigma}$ denotes an estimator of standard deviation obtained from sample $x_1, x_2, \ldots, x_m$. For the Cauchy kernel (2) suggested above, coefficients (5) and (6) amount to $W(\mathcal{K}) = 1$ and $U(\mathcal{K}) = 5/4\pi$.

In dealing with practical mathematical applications, significantly better results are obtained by introducing coefficients individuating parameters h for each kernel $\mathcal{K}$. In such investigations, a modification of the smoothing parameter will be applied. Moreover, it should be noted that in a classification task, this procedure is particularly important with regard to ensuring better estimation of the probability density especially at the borders regions of individual classes.

Consider, therefore, the non-negative modification coefficients

$$s_i = \left(\frac{\hat{f}_*(x_i)}{\overline{s}} \right)^{-c}, \tag{7}$$

for $i = 1, 2, \ldots, m$, where $c \geq 0$ is called the 'modification intensity', and $\overline{s}$ denotes the geometrical mean of the kernel estimators values $\hat{f}_*(x_i)$, given in its basic form (1). When $c = 0$, the lack of smoothing parameter modification takes place, and with the increase in value of the parameter c, the intensity of the procedure grows. Based on the results from the mean-square criterion, the following value has been recommended

$$c = 0.5. \tag{8}$$

Finally, the kernel destiny estimator defined in Eq. (1) can be written in the following form taking into account the coefficients of modifications smoothing parameter

$$\hat{f}(x) = \frac{1}{m h_1 h_2 \ldots h_n} \sum_{i=1}^{m} \frac{1}{s_i} \mathcal{K}\left(\frac{x_1 - x_{i,1}}{h_1 s_i} \right) \mathcal{K}\left(\frac{x_1 - x_{i,2}}{h_2 s_i} \right) \ldots \mathcal{K}\left(\frac{x_1 - x_{i,n}}{h_n s_i} \right) \tag{9}$$

with natural notations as used hence

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad x_i = \begin{bmatrix} x_{i,1} \\ x_{i,2} \\ \vdots \\ x_{i,n} \end{bmatrix}. \tag{10}$$

More information concerning statistical kernel estimation methodology can be found in Kulczycki [2009], Silverman [1986] and Wand and Jones [1995].

2.2. *Bayes classification of interval information*

In this subsection, a method of classifying imprecise information will be presented. The main reduction algorithm described in next part of the on-going paper is particularly dedicated to being used with the procedure for classifying interval information, and the application of both algorithms made mention of herein, results in a very good classification performance. The effects of numerical verification as shown in Sec. 5 is a confirmation of this thesis. The classification algorithm is the original idea of the author of this publication, and it was the main subject of his doctoral dissertation [Kowalski (2009)] and other scientific research Kulczycki and Kowalski (2015).

If we denote $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_J$ as particular kernel density estimators associated with the data samples representing every considered class, according to the Bayes approach, which ensures a minimum of potential losses due to misclassification, if samples size $m_1, m_2, \ldots, m_J$ are proportional to the 'frequency' of occurrence of the elements from each class, then the tested element $\widetilde{x} \in \mathbb{R}$ belongs to this class for which the value $m_1\hat{f}_1(\widetilde{x}), m_2\hat{f}_2(\widetilde{x}), \ldots, m_J\hat{f}_J(\widetilde{x})$ is the largest. Consequently, in situations wherein the desired information is given by the interval $[\underline{x}, \overline{x}]$, the tested element belongs to this class for which the value

$$\frac{m_1}{\overline{x}-\underline{x}}\int_{\underline{x}}^{\overline{x}} \hat{f}_1(x)\,dx, \frac{m_2}{\overline{x}-\underline{x}}\int_{\underline{x}}^{\overline{x}} \hat{f}_2(x)\,dx, \ldots, \frac{m_J}{\overline{x}-\underline{x}}\int_{\underline{x}}^{\overline{x}} \hat{f}_J(x)\,dx \tag{11}$$

is the largest. In the above formula, the positive constants $1/(\overline{x}-\underline{x})$ can be omitted as being negligible for the optimization problem under consideration, therefore, finally, it can be presented in the following form

$$m_1\int_{\underline{x}}^{\overline{x}} \hat{f}_1(x)\,dx, m_2\int_{\underline{x}}^{\overline{x}} \hat{f}_2(x)\,dx, \ldots, m_J\int_{\underline{x}}^{\overline{x}} \hat{f}_J(x)\,dx. \tag{12}$$

Moreover, for every $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_J$ that is fixed

$$\int_{\underline{x}}^{\overline{x}} \hat{f}(x)\,dx = \hat{F}(\overline{x}) - \hat{F}(\underline{x}), \tag{13}$$

where $\hat{F}$ means the primitive of the function $\hat{f}$. For the Cauchy Kernel (2) used here, the following analytical formula can then be obtained

$$\hat{F}(x) = \frac{1}{m}\sum_{i=1}^{m}\left[\frac{(x^2 - 2xx_i + x_i^2 + h^2)\arctan(\frac{x-x_i}{h}) + h(x-x_i)}{x^2 - 2xx_i + x_i^2 + h^2} + \frac{\pi}{2}\right] \tag{14}$$

(note that the constant $1/\pi$ could be again omitted). The above formula for the additional procedures that are often employed in the practical applications utilizing kernel estimators was easily generalized through the modification of the smoothing parameter $s_1, s_2, \ldots, s_m$ [Kulczycki (2008)]. Finally, formulas (12)–(14) define a complete algorithm of classification in the one-dimensional case.

In the multidimensional case, when information is represented by the interval vector

$$[[\underline{x}_1, \overline{x}_1], [\underline{x}_2, \overline{x}_2], \ldots, [\underline{x}_n, \overline{x}_n]]^T \tag{15}$$

the tested element belongs to the class for which the value

$$m_1 \int_E \hat{f}_1(x)\, dx, m_2 \int_E \hat{f}_2(x)\, dx, \ldots, m_J \int_E \hat{f}_J(x)\, dx \tag{16}$$

is the largest. In (16) $E = [\underline{x}_1, \overline{x}_1] \times [\underline{x}_2 \times \overline{x}_2] \times \cdots \times [\underline{x}_n, \overline{x}_n]$. According to the properties of the product kernel used here, calculations of the values of the n-dimensional integrals stated above, can be decomposed to the n-independent one-dimensional tasks, due to the dependence

$$\int_E K(x)\, dx = [\mathcal{J}(\overline{x}_1) - \mathcal{J}(\underline{x}_1)][\mathcal{J}(\overline{x}_2) - \mathcal{J}(\underline{x}_2)] \ldots [\mathcal{J}(\overline{x}_n) - \mathcal{J}(\underline{x}_n)], \tag{17}$$

where $\mathcal{J}$ means the primitive function of the (one-dimensional) kernel $\mathcal{K}$. Taking into account the formulas (13) and (14) obtained earlier for the one-dimensional case, it completes the algorithm for classification of interval information in the multidimensional case.

3. Data Sample Reduction

This section is the central part of this paper. The aim of this section is to present a novel method of data sample reduction for the classification of interval information that was shown in the previous section. Its concept is based on sensitivity analysis, and was inspired by artificial neural networks research. The main goal of this procedure application is to increase the number of proper classifications, while reducing the cardinality of the pattern data, and, consequently, decreasing calculation load.

Initially, in this section, sensitivity analysis, as used within the area of artificial neural networks, will be presented. In the second part of this section, the main algorithm for data sample reduction will be introduced, and its properties will be discussed.

3.1. *Sensitivity analysis in feedforward neural networks*

While solving complex tasks using artificial neural networks, particular elements of an input vector have diverse influence on the appearing phenomena. Thus, the information introduced by such elements could be important, neutral or even harmful in the considered context. In order to determine the importance, when a desired goal is being striven for, of particular elements, the sensitivity analysis of a neural network response for a training data set can be applied [Engelbrecht *et al.* (1999); Engelbrecht (2001); Yeung *et al.* (2010)].

Consider a three layer neural network [Zurada (1992)] with sigmoid activation function

$$F(z_0, z_1, z_2, \ldots, z_k) = \frac{1}{1 + \exp(-b \sum_{i=0}^{k} (w_i z_i))}, \tag{18}$$

where z_i and w_i denote ith input and its weight, and the positive coefficient b is called a 'shape parameter'. Additionally, assume that

$$\begin{aligned} \mathcal{X}_1 &= (z_{1,0}, z_{1,1}, \ldots, z_{1,m}), \\ \mathcal{X}_2 &= (z_{2,0}, z_{2,1}, \ldots, z_{2,q}), \\ \mathcal{X}_3 &= (z_{3,0}), \end{aligned} \tag{19}$$

are a neuron outgoing signal that forms the input, hidden and output layers, respectively. Moreover, assume in the presented paper, that q is established (for the particular classification problem under consideration) as an integer part of value $\sqrt{m}$. In addition, assume that $\mathcal{W}_2$ and $\mathcal{W}_3$ are also denoted as matrices of the neural networks hidden and output layers weights, respectively. Hence, training data v, as a pair, are defined as

$$v = (\mathcal{X}_1^{(p)}, T^{(p)}), \tag{20}$$

where $T^{(p)}$ is the target learning vector and p denotes the number of elements in a learning set ($p = 1, 2, \ldots, P$).

The main idea of the applied 'sensitive method' consists of calculating, after a learning phase, the values of coefficients representing the influence of each element of the input data $z_1, z_2, \ldots, z_m$, on the output y. For each neural network input, the coefficient is defined as

$$S_i^{(p)} = \frac{\partial y(z_1, z_2, \ldots, z_m)}{\partial z_i}, \tag{21}$$

for $i = 1, 2, \ldots, m$. These coefficients are based on the parameters given above, and can be calculated as

$$S_i^{(p)} = F'^{(II)} \sum_{k=1}^{q} F_k'^{(I)} w_{3,1,k} w_{2,k,1}, \tag{22}$$

where $F'^{(II)}, F'^{(I)}$ are the value of the derivative of the output and the hidden layer activation function, respectively. To aggregate parameters (22) for all P training pairs (20), the mean square average sensitivity coefficients are defined by

$$S_i = \sqrt{\frac{\sum_{p=1}^{P} \left(S_i^{(p)}\right)^2}{P}}. \tag{23}$$

The final result of this methodology is a proper (with respect to the problem under investigation) modification of neural network weights and structure, based on the sensitivity coefficients obtained in such a way.

3.2. *Data reduction procedure*

As stated earlier, in many practical tasks, the individual elements of the particular samples set that is of interest could be important, neutral or even harmful from the point of view of deriving the appropriate results of the classification procedure considered here. A sensitivity analysis of neural networks will be used in the following so as to show how to improve its quality, as well as, to enhance its calculation speed.

Consider the generalized definition of the kernel estimator

$$\hat{f}(x) = \frac{1}{mh^n} \sum_{i=1}^{m} w_i K\left(\frac{x - x_i}{h}\right), \tag{24}$$

where the non-negative coefficients $w_1, w_2, \ldots, w_m$ are such, that $\sum_{i=1}^{m} w_i = m$ can be interpreted as revealing the 'importance' of particular elements of the sample $x_1, x_2, \ldots, x_m$.

For the data reduction procedure, the neural networks were constructed for every class separately, according to the information presented in Sec. 3.1. Now let us assume that class index ($j = 1, 2, \ldots, J$) will be fixed. In a learning procedure, inputs (20) of these networks correspond to particular elements of the samples using the kernel function (2)

$$\mathcal{X}_1 = ((K(x_p|_{x_1,h,s_1}), K(x_p|_{x_2,h,s_2}), \ldots, K(x_p|_{x_m,h,s_m})) \tag{25}$$

while the outputs are defined as particular values of the kernel estimator (9) for argument x_p, where p index denotes the individual elements of the artificial neural network learning sample. In this case, the value p is equal to the cardinality of the sample m.

In so-doing, the following neural network learning parameters were used: maximal number of epochs — 100, maximal learning error — 0.015, and, finally the learning rate (learning speed) parameter — 0.3 (an explanation of the adopting of the afore-mentioned parameters of the learning process will be put forward in the next part of this section).

After employing the back-propagation learning algorithm, a sensitivity analysis takes place according to the method presented in the previous subsection. Using Eq. (23), coefficients describing sensitivity S_i are used to calculate the preliminary value of the weights

$$\widetilde{w}_i = \left(1 - \frac{S_i}{\sum_{j=1}^{m} S_j}\right), \tag{26}$$

in such way, the obtained values are normalized according to the following formula

$$w_i = \left(m \frac{\widetilde{w}_i}{\sum_{j=1}^{m} \widetilde{w}_j}\right). \tag{27}$$

In formula (26) it is assumed that the constructed neural network is the most sensitive to outlier elements, as well as to redundant elements. Regarding the generalized definition of the kernel estimator Eq. (24), for the previous mentioned values, the lower values $\widetilde{w}_i$ and weights w_i are needed. Hence, the values of the coefficients (27) represent then the 'importance' of the sample elements within the classification problem.

By way of empirical research, it can be naturally assumed that elements are going to be removed for which $w_i < 1 (i = 1, \ldots, m)$. This fixed value can be called a 'boundary weight'. Here, it is good to recall that elements w_i were normalized in (27), this implies that the average value will equal 1. The act of decreasing such assumes that the threshold value in process research results in a significant reduction in the degree of erasing patterns, and that the number of misclassification will only gently arise. Moreover, in the neighborhood of a point 1, the effect on the quality of classification is practically negligible. On the other hand, an increase of the threshold value results in the rapid growth of the number of misclassifications. This effect is brought about by the loss of the important and nonredundant information contained in the set of patterns.

Figure 2 serves to illustrate the above-mentioned procedure. In this example-research, the classification of two classes was worked out. Each class was represented by a 50 element pattern obtained from a normally distributed random numbers generator. Herein, the following parameters are employed: average value 0 for first class and 2 for the second and standard deviation 1 for both. The validation sample contains 1,000 interval type elements for each class.

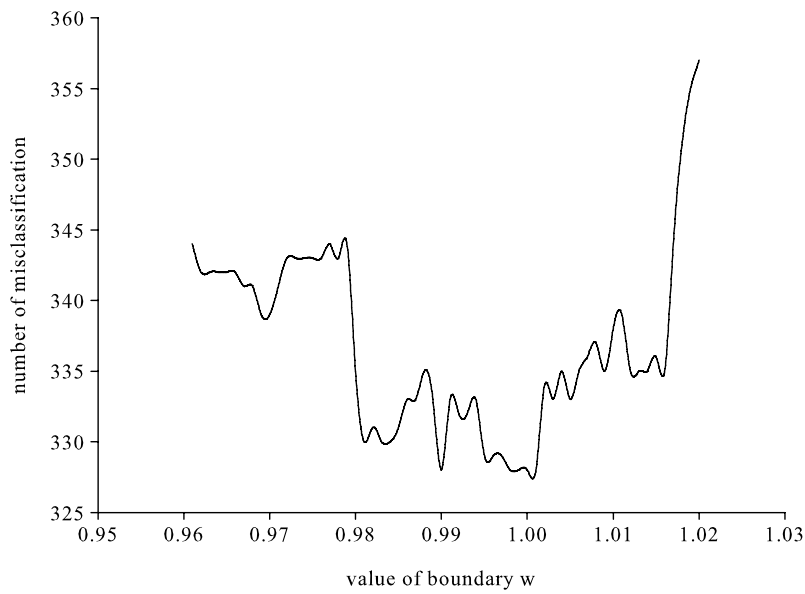


Fig. 2. The number of misclassifications depending on the boundary value w that is in use.

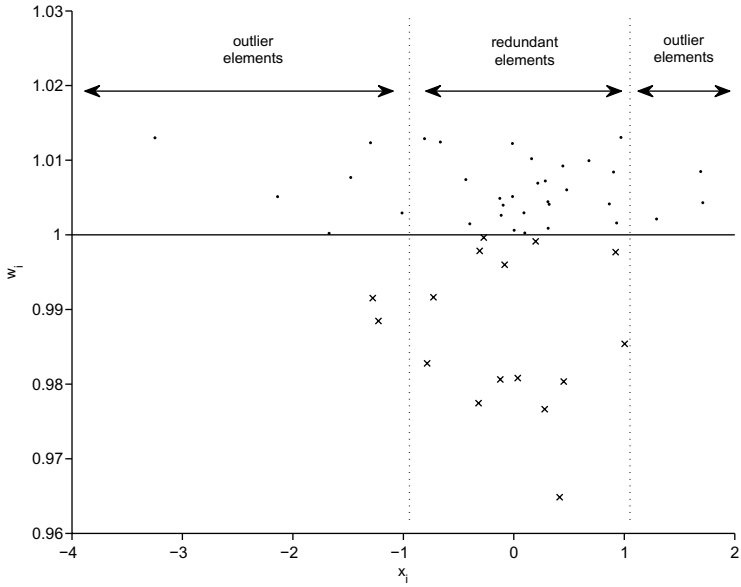


Fig. 3. The first class patterns with corresponding coefficients w_i.

Furthermore, Figs. 3 and 4 show the weights attached to the individual elements of both sets of patterns, as obtained from the sensitivity analysis of the artificial neural network. Here, it can be seen that elements with weight of smaller than the fixed threshold value 1 were assigned to some elements of an atypical nature (outliers), as well as to some elements of the areas of increased data density. The latter were treated by the algorithm as being redundant, and their roles were adopted by elements considered more representative, those with weights greater than 1. Therefore, in Figs. 3 and 4, all components intended for reducing (i.e., the coefficient $w_i < 1$) have been marked with a $\times$ symbol, whereas, elements indicated by a dot (i.e., $w_i \geq 1$) constitute a new set of patterns, in this case, the patterns employed within the classification task. Moreover, within the mentioned figures, a solid line indicates the border defined by threshold value 1. As seen in these graphs, the algorithm for a data reduction mainly focuses on the removal of the redundant, and, in addition, it focuses upon selected items of atypical nature. The prior mentioned fact is particularly important when using the main procedure as a preprocessing algorithm for classification tasks. Removal of all or most of the elements with atypical nature, creates significant change in the boundaries between classes, and, consequently increases the number of misclassifications.

In the following part of this section, the configuration of the neural network used for sensitivity analysis will be explained. As previously mentioned, for the data reduction tasks, a three-layered neural network, in which all neurons have a sigmoidal transfer function (18), is used. As a result of numerical tests, it was found

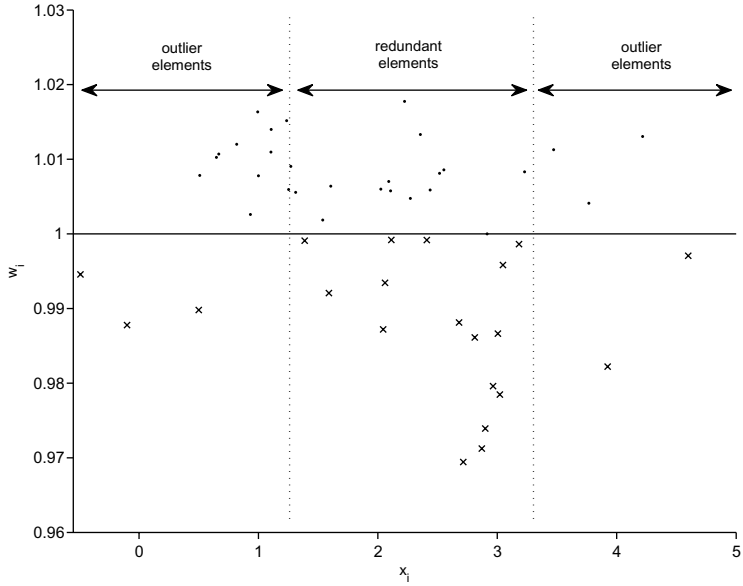


Fig. 4. The second class patterns with corresponding coefficients w_i.

that the most preferred number of neurons existing in the hidden layer (i.e., the second) is an integer part of value $\sqrt{m}$, where m describes the numbers of particular pattern data that have been found within the investigated class. The act of decreasing such established value induces a significant decrease in the quality of neural networks learning. Consequently, this implicates that the number of misclassifications will arise. The afore-mentioned outcome greatly reduces the possibility of applying of such a network.

On the other hand, increasing the number of neurons in the hidden layer results in a slight decrease in mean square error during training network activity, but highly increases the duration of the learning process. In this case, in the course of verification tests, a strong network susceptibility to overfit was reported. The aforementioned induces much worse results within the sensitivity analysis, which implies an increase in the error generated by a particular classifier based on the reduced data.

As a result of a number of numerical simulations, the learning rate (lr) parameter was established and recommended as 0.3. The learning rate is a training parameter that controls the size of weight and bias changes in the training procedure. In Table 1, the influence of this parameter on the speed of learning can be seen.

This table presents the number of iterations (epochs) needed to achieve an established learning Mean Squared Error (MSE), with a fixed lr parameter. During these tests, it was found that the learning wherein the error is reduced below 0.02, gives the desired results, i.e., appropriate quality data reduction, and, hence, brings about an improvement of the classification results. Therefore, achieving a MSE learning

Table 1. Number of epochs in neural training process.

	lr				
Learning error	0.9	0.7	0.5	0.3	0.1
0.02	18	15	19	20	26
0.015	98	87	80	59	77
0.01	350	410	286	223	289

error below 0.015 is recommended as being the terminal condition of the applied neural network training algorithm.

In the table under consideration, for lr parameters larger then 0.3 (i.e., 0.5, 0.7 and 0.9), it is quite quickly evident that an error 0.02 is achieved, but that the further learning process is characterized by stagnation. In this case, this means that the error of 0.015 is obtained in a considerably longer time interval than in the case of the parameter lr being equal to 0.3. For comparison, in the last row of the table, a learning error of 0.01 was provided. The results obtained here only greatly underline these explanations.

Table 2. Convergence of the learning process.

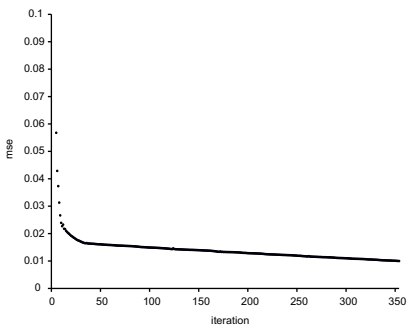
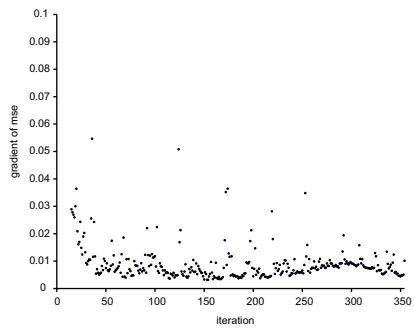
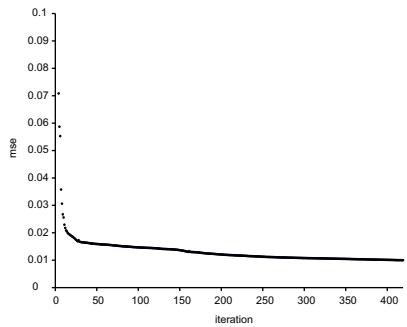
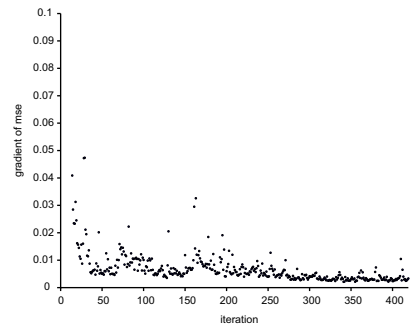
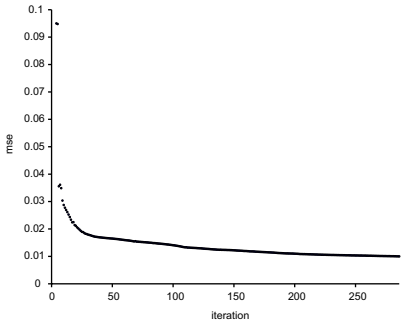
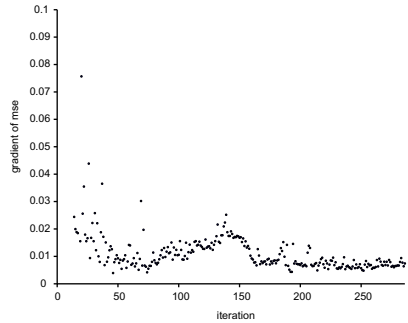
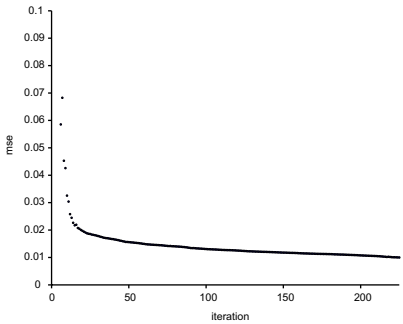
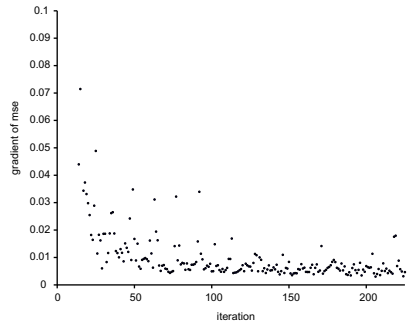
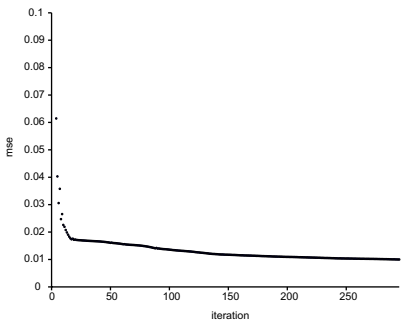
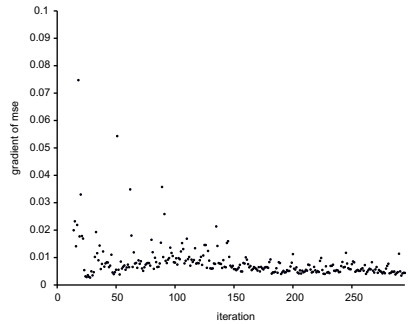
lr	MSE of learning	Gradient of MSE
0.9		
0.7		

Table 2. (*Continued*)

lr	MSE of learning	Gradient of MSE
0.5		
0.3		
0.1		

Furthermore, increasing the lr parameter results in an increased value of each coordinates in the vector of weights increment in each iteration of learning process — as based on the back-propagation algorithm. As a consequence, there is a rapid dip of error (within the first row of the table) and then the search for a better solution requires more time (epoch).

On the other hand, a reduction of lr (the last column of the table) increases the number of iterations needed to obtain the established error. This is a result of movement in the space of solutions by way of small steps at any one time. The foregoing comments are a natural result of the use of gradient methods for optimization of network weights.

Table 2 introduces plots of the convergence of learning process for the various parameters lr. In the second and third columns, both diagrams of the MSE error and its gradient, respectively, are located. Graphs of these plots, in showing the convergence of the learning process, confirm the above thesis. In addition, whereas the graphs depicting the gradient of the MSE error (during learning of neural network) indicate that with the recommended parameter lr = 0.3, the greatest variability comes about in the first 100 epochs. Moreover, the cases of lr parameter in 0.9 or 0.7 indicate that significant changes in the neural network weights are carried out in a much larger number of iterations: 250 and 200, respectively (thus later than in the previous case), whereas for lr values of 0.5 and 0.1, the value of the gradient of not greater than 0.015 is obtained from about the 150 epoch of the neural network training process. In conclusion, for the recommended parameter learning rate, lr = 0.3, the figure of 100 was adapted as being the maximum number of preferred iterations (epochs).

Apart from Table 2, for confirmation for the establishment of these parameters, Table 3 reveals the values of MSE error that were obtained in 100 training epochs for a fixed learning speed (lr) parameter.

Algorithm 1 Procedure for reduction data set

1: {main loop}
2: **for** $j = 1$ to J **do**
3: X_j - data pattern set in j class
4: Determine the topology of the ANN
5: Create patterns for learning neural network
6: Train artificial neural network
7: Perform sensitivity analysis for learning data X_j
8: Normalize the sensitivity coefficients w_j
9: Sort X_j on the basis of the coefficients w_j
10: Eliminate neutral and negative elements from X_j
11: **end for**
12: **return** X^{red} - reduced data pattern sets

What is more, Algorithm 1 reveals all activities that are related to the reduction algorithm. As mentioned at the beginning of this section, reduction tasks are carried out separately for each of the considered J classes.

Table 3. The MSE in the 100 epoch of the training process.

lr rate	0.9	0.7	0.5	0.3	0.1
mse	0.0149	0.0147	0.0141	0.0131	0.0136

4. Complete Classification Algorithm with Reduced Data Set

In this part of the on-going paper, the authors will present a complete classification algorithm with reduced data set. In doing so, Fig. 5 illustrates all procedures which fall within the main algorithm.

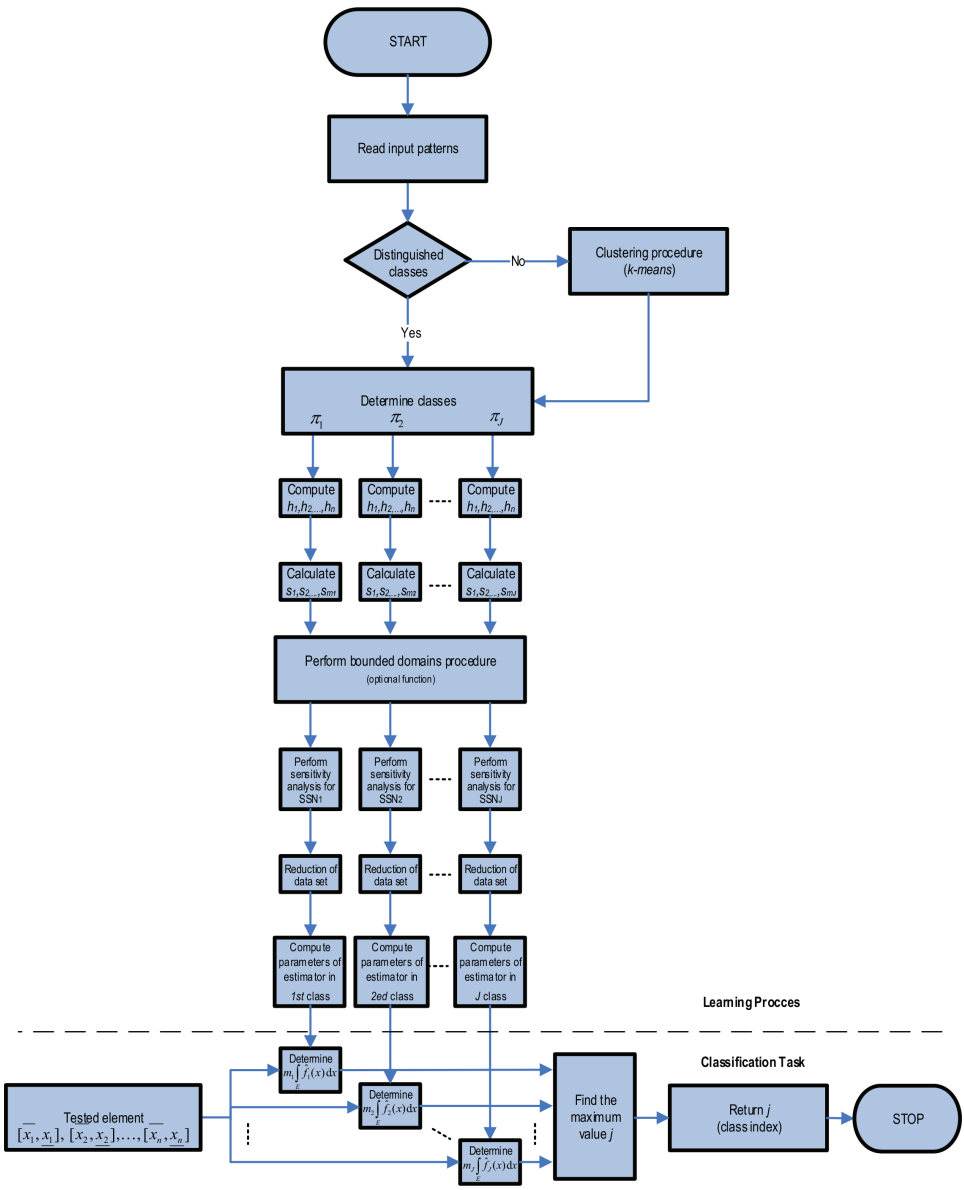


Fig. 5. Flowchart of algorithm for classifying interval information with a reduced data set.

Herein, the proffered flowchart consists of two parts. The first, located above the dashed line, exhibits all relevant steps related to the construction of the interval Bayes classifier that incorporates the reduction procedure. The second, located at the bottom part of Fig. 5, reveals the classification of the new element. Here, the knowledge developed and presented in Sec. 2 will be employed.

When initializing the algorithm, the patterns representing the class must firstly be distinguished. In the absence of so-doing, a clustering procedure must be undertaken. In the algorithm under consideration, the simple and rapid *k-means* method [Wu and Kumar (2009)] is recommended, and by way of its use, the collection of pattern sets representing each class is obtainable.

In the next phase of the algorithm, the parameters of the Kernel Density Estimator, as presented in Sec. 2.1 should be calculated separately within each of the previously selected classes. Thus, in order to obtain a smoothing coefficient $h_j (j = 1, 2, \ldots, n)$, the plug-in procedure should be adopted, and done so in accordance with an assumed type of product kernel (3).

Next, the coefficients of modification of smoothing parameter s_i $(i = 1, 2, \ldots, m)$ according to an algorithm based on Eqs. (7) and (8), should be computed.

In the subsequent step, the optional procedure of bounded domains for one- or multi-dimensional case can be used. This operation is very often employed when the natural domain of a density to be estimated, is not the whole real number set, but is an interval bounded on one or both sides. For example, the age data and the length of flowers are measurements of positive quantities, and so it will be preferable for many purposes to obtain density estimates $\hat{f}$, for which $\hat{f}(x)$ is zero for all negative x. However, the bounded domains can also be realized by the use of an identical boundary parameter for every coordinate of domain dimension, and by way of employing different boundary coefficients for each of these. More details on the use of the bounded domains procedure in problems of classification can be found in Kowalski [2009].

The next relevant stage in the main procedure is to create a pattern data set for each class, and to analyze the learning data in accordance with the procedure set out in Secs. 3.1 and 3.2.

Based on the above step, the sensitivity coefficients S_i and the calculated values of weights w_i, a reduction in each of the considered pattern sets takes place. The detailed course of action in this stage was described in the previous section.

Again, the parameters of the kernel density estimator (9) should be calculated — due to the change of the content of the pattern data sets. In particular, this should be done to obtain the smoothing parameters, as well as the coefficients of modification of the smoothing parameter. The derived values are the starting point for classifying the new element which contains interval type information. The new tested (classified) element is then subjected to the classification procedure for either a one-dimensional or a multi-dimensional problem as described in Sec. 2.2.

More information on the classification of interval information can be found in Kowalski [2009] and Kulczycki and Kowalski [2011].

5. Numerical Results

The verification of correctness of the method presented in this paper for the neural reduction of data sets for classifying interval information was conducted by way of numerical simulation.

The following are the results for data obtained by the use of a random number generator with a normal distribution and with a given vector of expected value and covariance matrix (or linear combinations thereof), as derived from the implemented multivariate normal distribution generator based on the concept of Box-Muller [Brandt (1999)].

The quality assessment methods presented here were obtained by generating a random number of the assumed distribution, and by way of an analysis of the results of an assessment of the correctness of the classification procedures for both data types: interval, and (for comparison) unambiguous. In order to ensure the repeatability of the results, for each of the pseudo-random sets, the seed value that defines it, was strictly determined.

After obtaining the sequences of pseudo-random patterns representing the different classes, test data of particular classified items was generated, including, for comparative purposes, an interval type and, occasionally, an unambiguous type. Each class corresponds to a set of size 1,000 items. The classified elements were obtained by way of one of the aforementioned generators, with a normal distribution of the first pseudo-random number, as well as the second, as taken from a generator with uniform distribution defining the location of the first as within an interval of arbitrarily assumed length. This represents a supply of information of an interval type when no circumstances for the considered imprecision are evident, although its size is known. Such an interpretation seems to be the most appropriate for the majority of practical interval analysis applications. The verification tables show a set of results displaying the following size of patterns: 10, 20, 50, 100, 200, 500 and 1,000. In the mentioned tables, each cell contains the results obtained from 100 runs, giving an average classification error, defined on the basis of these 100 random test samples.

The basic form of the research conducted for the classification method developed here is built upon the information interval and on patterns that are given uniformly. With regard to the one-dimensional (i.e., for $n = 1$) pattern, a first class was obtained by the use of a pseudo-random number generator with a normal distribution $N(0, 1)$, while the second class was gained by way of $N(2, 1)$. The final results of these examples are presented in Table 4. These results will provide comparative data in relation to those presented in the subsequent tables, therefore, such results serve as a foundation example reference.

Initially, studies were performed for the reduction of pattern size using the algorithm based on the method of neural network sensitivity for training data (see Sec. 3.1). In the considered case, as a result of the numerical verification, a reduction of 38% of the total elements found within each pattern was achieved.

Table 4. Average classification error for the basic concept of Bayes classification.

	Interval length						
No. of elements	0.00	0.1	0.25	0.5	1.00	2.00	5.00
10	0.1713	0.1720	0.1720	0.1723	0.1729	0.1761	0.1944
20	0.1655	0.1669	0.1669	0.1672	0.1680	0.1713	0.1888
50	0.1602	0.1605	0.1606	0.1609	0.1617	0.1652	0.1848
100	0.1596	0.1601	0.1602	0.1604	0.1615	0.1650	0.1827
200	0.1596	0.1602	0.1604	0.1609	0.1618	0.1650	0.1840
500	0.1591	0.1595	0.1596	0.1602	0.1613	0.1647	0.1844
1,000	0.1579	0.1584	0.1588	0.1591	0.1603	0.1637	0.1833

Table 5. The basic concept of Bayes classification with neural reduction.

	Interval length						
No. of elements	0.00	0.1	0.25	0.5	1.00	2.00	5.00
10	0.1747	0.1759	0.1766	0.1784	0.1823	0.1853	0.2112
20	0.1616	0.1622	0.1629	0.1637	0.1649	0.1689	0.2087
50	0.1521	0.1529	0.1535	0.1541	0.1546	0.1567	0.2065
100	0.1501	0.1505	0.1509	0.1515	0.1521	0.1545	0.2010
200	0.1482	0.1489	0.1495	0.1501	0.1515	0.1530	0.2001
500	0.1472	0.1478	0.1488	0.1494	0.1507	0.1521	0.1967
1,000	0.1461	0.1466	0.1478	0.1487	0.1499	0.1514	0.1921

The full results of the reduction algorithm are presented in Table 5. Initially, within the table, cells can be distinguished for displaying a situation in which the obtained quality was lower than the classification of the basic case (i.e., without applying the reduction procedure) which is performed in Table 4. This relates only to patterns with 10 elements in each pattern set. With such an extremely low set size, this result, however, is satisfactory. This is because, by reducing the data set, the number of elements have decreased to about 6, and the formed pattern sets of each class are not adequately representative of the true pattern characteristics. In succeeding cells, an improved quality of classification can be distinguished. Here, the first of these, mainly display patterns with cardinality of 20 and 50. These results were improved by way of reducing the amount of misclassification by 5%, and subsequent misclassification by 5–10%. It should be noted that the reduction of patterns not only results in a decrease in misclassification quantities, but, above all, generates a significant increase in the speed of calculation.

The next subsections present the results of tests carried out on the compiled methods for the classification of interval type information and neural reduction, as based on the benchmark data. Due to the specific conditions, such type of data was not found in publicly available repositories and web pages, although there are a few sets of interval data that can be found, for example, *oil* [Sato-Ilic (2003)], *fishdataset* [Peng and Li (2006)] or *pre-sliding friction rig* [Chetwynd *et al.* (2006)].

However, these are intended only for clustering procedure tests, or they do not contain unambiguous reference elements (patterns) suitable for the learning process.

5.1. *DataToy 2D*

In this subsection, a comparison of the neural reduction algorithm with other similar reduction procedures using benchmark data is to be shown. The numerical verification utilized was designed to assess both the quantities of misclassification, and, above all, reduced amount of elements in the considered sample.

At the beginning of the comparison, a very simple and intuitively understandable percentage reduction algorithm was used. This procedure assumes removal of a random sample of pre-determined percentage of the elements. Here, the strong nondeterministic character of this algorithm should be noted. More details on the 'percentage' procedure can be found in Han and Kamber [2006].

Another algorithm utilized for verification and clarification is the *k-nearest neighbors* (*k-NN*) procedure. This consists of an iterative designation of *k-NN* for the considered element. When appointed in this way, the neighbors are represented by one point. It is also possible to set out a new point that is representative of all previously designated neighborhoods. This algorithm is based only on the information about the location of the items under consideration, and it does not take into account, for example, concentration, redundancy or the value of a probability density function. What is more, another quite strong constraint is the relatively weak control of the reduced amount of sample items. More information about this algorithm can be found in the work of Mitra and Pal [Mitra *et al.* (2002)].

Finally, the last reduction algorithm that we are to compare with the neural procedure described within this paper, is a reduction function based on the methodology of WPW. This method involves a shift of the individual kernels of the kernel density estimator function of the considered class. During this operation, particular weights which determine the significance of each item in a sample set are made available. In using the weights generated earlier in subsequent iterations, we can remove those items for which the weight is the smallest, and as a result, a reduced data set representing the class takes place. More details on this topic can be found in Babich and Camps [1996].

The employed *Toy* $2D$ benchmark data is published on the website:
http://www.cse.ust.hk/ twinsen/assgn2.pdf. This site features an example of two-dimensional random sample which represents the image of an eclipse of the Moon. Here, the learning sets contain 2,152 elements associated with the first-class, and 2,444 items contained in the second. Moreover, the test sample was drawn from a two-dimensional regular grid, including 26,130 elements from the first class and 34,371 from the second.

This data set will be used to compare all the above-mentioned reduction methods. It should be noted that the test data were made-up as intervals, as in the previous example. The following columns of Table 6 show the results for the interval

Table 6. The results of the numerical verification for the data found in *Toy* 2*D* are given as average classification error.

		Interval length					
Reduction type	Reduc.	0.00×0.00	0.10×0.10	0.25×0.25	0.50×0.50	1.00×1.00	2.00×2.00
No reduction	0%	0.0681	0.0737	0.0750	0.0766	0.0828	0.1097
Neural reduction	37%	0.0633	0.0729	0.0740	0.0750	0.0823	0.1055
Random 10%	10%	0.0630	0.0782	0.0795	0.0826	0.0955	0.1230
Random 20%	20%	0.0667	0.0756	0.0766	0.0785	0.0892	0.1198
Random 30%	30%	0.0640	0.0791	0.0796	0.0826	0.0955	0.1250
Random 40%	40%	0.0781	0.0837	0.0848	0.0866	0.0898	0.1366
k-NN ($k = 1$)	26%	0.0622	0.0745	0.0754	0.0784	0.0889	0.1187
k-NN ($k = 2$)	59%	0.0636	0.0751	0.0759	0.0786	0.0903	0.1209
WPW (itr = 230)	10%	0.0675	0.0742	0.0754	0.0771	0.0832	0.1098
WPW (itr = 460)	20%	0.0676	0.0750	0.0758	0.0777	0.0838	0.1100
WPW (itr = 689)	30%	0.0677	0.0753	0.0760	0.0780	0.0842	0.1121
WPW (itr = 919)	40%	0.0685	0.0755	0.0765	0.0786	0.0846	0.1122

depending on its length, from 0.1 to 2.0, respectively. Again, for comparison, the results of a reduction of the unambiguous data was provided. In the second column of Table 6, information about the percentage fraction of the removed (reduced) elements from the learning sets is set out. The following rows correspond to the results obtained for the considered methods of reduction. Initially, the results obtained for the nonreduced data are shown. In the subsequent rows, the results derived from the neural method of reduction (as described in this paper) are revealed. Following these, inside the table, the results of a method of reducing the percentage (based on the approved percentage of reduction of 10% to 40%, respectively), are displayed. The middle part of the table then reveals the results gained for employing the *k-NN* procedure for $k = 1$ and $k = 2$. Finally, the bottom rows of Table 6 display the results derived from utilizing the WPW method. The variable parameter employed here is the duration of the algorithm expressed in the number of iterations (from 10 to 40).

From the consequences of an applied numerical verification (again seen in Table 6), we may conclude that the neural method gives very good results in comparison with the other standard reduction procedures used in manipulating data. As seen, the algorithm of 'percentage' reduction for a small fraction of the deleted elements (10%) gives quite acceptable results. Moreover, with regard to unambiguous data (interval length = 0), even better effects than the neuronal reduction can be observed (note that the main algorithm removes from the sample 37% of all the original items). Yet, with respect to the situation of dealing with interval data, it is also evident that there is a significant deterioration in the quality of classification in relation to both the basic classification algorithm and neural methods of reduction. In addition, further increasing the number of items removed from the sample results in an even more relatively strong deterioration of the classification.

In a comparison to the results obtained by the use of the *k-NN* algorithm, a similar observation can be seen for the unambiguous and the interval data as well. However, if the number of neighbors is $k = 2$, the results are better than the previous random method for the 40% reduction of the items. Hence, it can be said that the decision as to which elements will be removed from the sample is to be taken not randomly, but should be based only on their proper locations within the sample. Finally, it must be pointed out that the most advanced algorithm considered here, also gives poorer results than proposed in this paper neural reduction algorithm, both for the 10% and the 40% reduction as well.

5.2. *Data synthetic two-class problem*

In the following research, the data set was obtained from a well-known example described and referred to in the monograph [Ripley (1996)]. The data set contained within this *Synthetic Two-Class Problem* consists of two predefined subsets, that of learning and testing for each of the two classes, and incorporates, respectively, 250 and 1,000 elements.

The investigation in this subsection is a continuation of the discussion presented in the previous part of the paper. It also concerns the methods used for creating interval data as based on unambiguous data. Table 7 presents the results of a series of comparative tests for the *Synthetic Two-Class Problem*. The conclusions that can be drawn on the basis of these numerical simulations are very consistent with the previous one.

A notable feature in these studies is the appreciable much greater intensity in the differences between the methods described than that within the previous section. Most likely this is due to a significant disproportional cardinality within both sets of tests. In the first case, the learning sample has about 2,200 elements per one class, while in the second, evident are only 125 items per class. In addition, an even more

Table 7. The results of the numerical verification for the *Synthetic Two-Class Problem*.

		Interval length					
Reduction type	Reduc.	0.00×0.00	0.10×0.10	0.25×0.25	0.50×0.50	1.00×1.00	2.00×2.00
No reduction	0%	0.142	0.141	0.148	0.147	0.181	0.258
Neural reduction	36%	0.089	0.082	0.082	0.088	0.106	0.148
Random 10%	10%	0.110	0.101	0.098	0.102	0.117	0.163
Random 20%	20%	0.176	0.181	0.182	0.183	0.187	0.211
Random 30%	30%	0.184	0.185	0.187	0.188	0.191	0.218
Random 40%	40%	0.202	0.205	0.201	0.204	0.207	0.234
k-NN ($k = 1$)	27%	0.094	0.087	0.089	0.090	0.111	0.152
k-NN ($k = 2$)	60%	0.187	0.187	0.187	0.195	0.199	0.219
WPW (itr = 13)	10%	0.092	0.094	0.110	0.153	0.261	0.325
WPW (itr = 25)	20%	0.093	0.088	0.085	0.089	0.112	0.155
WPW (itr = 37)	30%	0.095	0.093	0.092	0.096	0.110	0.152
WPW (itr = 50)	40%	0.096	0.089	0.085	0.092	0.112	0.157

singular note is the fact that the method of neural reduction has a very favorable influence upon the quality classification of interval type information, yet it can be considered as having only a good effect, when compared to other methods, upon the classification of unambiguous data.

5.3. *Reduction of data as obtained in the clustering process*

If there are no previously distinguished classes before the learning process, the data set should be divided into smaller groups by way of employing the clustering process. This situation was also examined during the previous discussed numerical verification, both for the procedure for classifying inaccurate information, as well as with the use of the neural data sample size reduction algorithm. In exemplifying this, the data used in the one-dimensional example (described at the beginning of this section) were divided into two disjoint subsets by way of employing the algorithm *k-means*. In this way, the prototypes of two classes were distinguished.

The results of the algorithm for the basic concept of classification are shown in Table 8, while Table 9 displays the results after employing the application procedure of the neural reduction of sample size. As expected, the results presented in the first table are only slightly better when compared to the original data from Table 4. This is due to the larger merging character of pattern classes obtained during the use of a clustering procedure. What is more, the learning data obtained in this manner are characterized by a lack of both outliers and atypical items, e.g., those which can be found in the middle of the second class.

From the foregoing considerations, it can be concluded that in this case, the precision of the reducing algorithm will not be the most important factor. The confirmation of this thesis is evidenced in the results given in Table 9, which presents the classification error for the data obtained in the process of clustering. Here, one can easily see the significant improvement in classification with respect to the unreduced data displayed in Table 8, and, consequently, to the basic case as shown in Table 4, as well as to basic concept with reduced data as seen in Table 5.

On summarizing the results presented in the section concerning numerical verification, it can be concluded that they have confirmed the correctness of the compiled

Table 8. The basic concept of Bayes classification as used on data obtained by way of the clustering process.

	Interval length						
No. of elements	0.00	0.1	0.25	0.5	1.00	2.00	5.00
10	0.1744	0.1675	0.1677	0.1681	0.1692	0.1732	0.2050
20	0.1661	0.1627	0.1626	0.1627	0.1638	0.1685	0.2004
50	0.1587	0.1572	0.1572	0.1573	0.1588	0.1639	0.1991
100	0.1563	0.1557	0.1556	0.1560	0.1574	0.1629	0.1981
200	0.1558	0.1553	0.1551	0.1552	0.1564	0.1619	0.1970
500	0.1547	0.1546	0.1546	0.1547	0.1566	0.1624	0.1981
1,000	0.1529	0.1529	0.1529	0.1533	0.1546	0.1602	0.1968

Table 9. The basic concept of Bayes classification employing neural reduction on the data obtained through the clustering process.

	Interval length						
No. of elements	0.00	0.1	0.25	0.5	1.00	2.00	5.00
10	0.1745	0.1681	0.1681	0.1682	0.1690	0.1735	0.2087
20	0.1602	0.1549	0.1549	0.1551	0.1561	0.1608	0.1944
50	0.1497	0.1474	0.1473	0.1474	0.1485	0.1543	0.1916
100	0.1473	0.1461	0.1461	0.1463	0.1482	0.1536	0.1890
200	0.1460	0.1457	0.1456	0.1456	0.1465	0.1523	0.1877
500	0.1448	0.1446	0.1447	0.1450	0.1464	0.1525	0.1882
1,000	0.1430	0.1430	0.1431	0.1431	0.1445	0.1504	0.1866

herein neural reduction method dedicated for the classification procedure of imprecise interval-type information. As seen, the results gained by undertaking this procedure were compared with a case in which classified information was derived from unambiguous testing data, and also with the results of the use of other algorithms employed in reducing sample size.

A verification of numerical procedures also confirmed the usefulness of reducing pattern size in both terms: in improving the quality of classification as well as in inducing a significant decrease in calculation time. In dealing with cases of size patterns incorporating 200–1000 elements, the presented algorithm obtained an average of 10% reduction errors, while reducing the patterns to an amount that approximates 38% of the original. Furthermore, it enabled a consequent reduction in calculation time directly related to the classification, in a similar order of magnitude.

In all the conducted investigations, both with and without neural reduction, an increase of pattern sizes resulted in a decrease of both the average classification error and its standard deviation. This, in practice, allows the progressive improvement of the quality of classification as new data is obtained. However, with an increasing length of the interval, the classification error increases to minor degree. To a certain limit, this is reasonable in the context of the data structure.

The above conclusions are worth accentuating from the application point of view. This is because utilizing the aforementioned procedure makes it possible to increase the quality of the classification during the progression of the information available in the form of larger number patterns and more accurate classification. Moreover, employing a reducing algorithm provides further increases in the accuracy of classification by removing items that have a neutral or negative character. In most practical problems, it is necessary to establish a compromise between the quantity of available data and the quality of results, but in this case, by using the neural reduction algorithm, it is possible to attach to a patterns set, from the new data, only those elements that have a significant impact on the problem of classification. More details regarding the employment of numerical verifications of classification with data reduction procedure can be found in Kowalski [2009].

6. Summary and Conclusions

The subject of the investigations presented here was a procedure for reducing a data sample applied to a classification algorithm where the classified element is given as the interval vector, while the data representing pattern of each class consists of elements defined uniformly. The classification procedure has been worked out by way of employing a data reducing procedure based on the sensitivity method inspired by artificial neural networks. Herein, its goal is to increase the number of proper classifications, as well as, consequently, enhance calculation speed.

As seen within this paper, in particular, during numerical testing, it was established that, after applying the procedure for reducing pattern sets, the number of wrong classifications was lowered by approximately 10%, while the size of patterns was reduced by approximately 38%. The conjunction of these results is particularly worthy of attention: while appropriately reducing pattern sizes (which does imply a significant increase in calculation speed), the classification quality is also importantly improved.

The proposed reduction method has been analyzed in terms of properties of the recommended neural network parameters, in addition, the analysis of the convergence of the learning algorithm, presenting both the MSE error and its gradient has been discussed. What is more, the presented neural reduction algorithm was compared with a very simple and natural random reduction process, as well as with the $k - NN$ procedure, and also with the reduction algorithm inspired by WPW methodology. In all these cases, the investigated reduction method based on neural networks sensitivity analysis provided much better results. Additionally, as evident in the proofs provided within this paper, in utilizing the neural reduction algorithm, there is no need to arbitrarily ascribe parameters to an algorithm. This, hence, is another positive aspect held within the procedure described herein.

References

Babich, G. A. and Camps, O. I. [1996] "Weighted Parzen windows for pattern classification," *IEEE Trans. Pattern Anal. Mach. Intell.* **18**, 567–570.

Brandt, S. [1999] *Data Analysis* (Springer Verlag, New York, Berlin, Heidelberg).

Chetwynd, D., Worden, K. and Manson, G. [2006] "An application of interval-valued neural networks to a regression problem," *Proc. R. Soc. A* **462**, 3097–3114.

Duda, R. O., Hart, P. E. and Stork, D. G. [2000] *Pattern Classification* (Wiley, New York).

Drakengren, T. and Jonsson, P. [1997] "Eight maximal tractable subclasses of Allen's algebra with metric time," *J. Artif. Intell. Res.* **7**, 25–45.

Engelbrecht, A. P., Fletcher, L. and Cloete, I. [1999] "Subharmonic solutions with prescribed minimal period for nonautonomous Hamiltonian systems," *Int. Joint Conf. Neural Networks*, *IJCNN' 99*, Vol. 3, pp. 1829–1833.

Engelbrecht, A. P. [2001] "Sensitivity analysis for selective learning by feedforward neural networks," *Fundam. Inf.* **46**(3), 219–252.

Gil, M. A. and Hryniewicz, O. [2009] "Statistics with imprecise data," in *Encyclopedia of Complexity and Systems Science*, ed. Meyers, R. A. (Springer, Heidelberg), pp. 8679–8690.

Ghost, A. K., Chaudhuri, P. and Sengupta, D. [2006] "Classification using kernel density estimation: Multiscale analysis and visualization," *Technometrics* **48**, 120–132.

Han, J. and Kamber, M. [2006] *Data Mining*: *Concepts and Techniques* (Morgan Kaufmann, San Francisco).

Jaulin, L., Kieffer, M., Didrit, O. and Walter, E. [2001] *Applied Interval Analysis* (Springer, Berlin).

Kacprzyk, J. [1997] *Multistage Fuzzy Control*: *A Model-Based Approach to Fuzzy Control and Decision Making* (Wiley, Chichester).

Kowalski, P. A. [2009] "Bayesian classification of imprecise interval-type information," Ph.D. Thesis, *Systems Research Institute*, *Polish Academy of Sciences* (in Polish).

Kowalski, P. A. and Kulczycki, P. [2010] "Data sample reduction for classification of interval information using neural network sensitivity analysis, Artificial Intelligence: Methodology, Systems, and Applications, Vol. 6304 of the Series, Lecture Notes in Computer Science, Springer-Verlag, Berlin, Heidelberg, pp. 271–272.

Kulczycki, P. [2008] "Kernel estimators in industrial applications," in *Soft Computing Applications in Industry*, ed. Prasad, B. (Springer-Verlag, Berlin), pp. 69–91.

Kulczycki, P. [2009] "Statistical kernel estimators for data analysis and exploration tasks: Theory and applications," *Proc. 14th WSEAS Int. Conf. Applied Mathematics*, pp. 257–262.

Kulczycki, P. and Kowalski, P. A. [2011] "Bayes classification of imprecise information of interval type," *Control Cybern.* **40**(1), 101–123.

Kulczycki, P. and Kowalski, P. A. [2015] "Bayes classification for nonstationary patterns," *Int. J. Comput. Methods* **12**(2), 1550008-1–19.

Lukasik, S. and Kulczycki, P. [2011] "An algorithm for sample and data dimensionality reduction using fast simulated annealing," in *Advanced Data Mining and Applications*, eds. Tang, J., King, I., Chen, L. and Wang, J., Lecture Notes in Artificial Intelligence, Vol. 7120 of the Series, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 152–161.

Mitra, P, Murthy, C. A. and Pal, S. K. [2002] "Density-based multiscale data condensation," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 734–747.

Peng, W. and Li, T. [2006] "Interval data clustering with applications," *Proc. 8th IEEE Int. Conf. Tools with Artificial Intelligence*, Arlington, 13–15 November, pp. 355–362.

Ripley, B. D. [1996] *Pattern Recognition and Neural Networks* (Cambridge University Press, Cambridge).

Sato-Ilic, M. [2003] "Weighted principal component analysis for interval-valued data based on fuzzy clustering," *Proc. Int. Conf. Systems*, *Man and Cybernetics*, Washington, Vol. 5, 5–8 September, pp. 4476–4482.

Silverman, B. W. [1986] *Density Estimation for Statistics and Data Analysis* (Chapman and Hall, London).

Wand, M. P. and Jones, M. C. [1995] *Kernel Smoothing* (Chapman and Hall, London).

Wu, X. and Kumar, V. [2009] *The Top Ten Algorithms in Data Mining* (Chapman and Hall, Boca Raton).

Yeung, D. S., Cloete, I., Shi, D. and Ng, W. W. Y. [2010] *Sensitivity Analysis for Neural Networks* (Springer-Verlag, Berlin Heidelberg).

Zhao, Y., He, Q. and Chen, Q. [2005] "An interval set classification based on support vector machines," *Proc. Joint Int. Conf. Autonomic and Autonomous Systems, 2nd Int. Conf. Networking and Services*, Silicon Valley, 25–30 September 2005, pp. 81–86.

Zurada, J. [1992] *Introduction to Artificial Neural Neural Systems* (West Publishing Co, St. Paul).