Computing and Informatics, Vol. 30, 2011, 901–912

MAXIMUM COVERAGE METHOD FOR FEATURE SUBSET SELECTION FOR NEURAL NETWORK TRAINING

Štefan Boor

Institute of Information Engineering, Automation and Mathematics
Faculty of Chemical and Food Technology
Slovak University of Technology in Bratislava
Radlinského 9
812 37 Bratislava, Slovakia
e-mail: stefan.boor@stuba.sk

Communicated by Vladimír Kvasnička

Abstract. Every real object having certain properties can be described by a number of descriptors, visual or other, e.g. mechanical, chemical etc. A set of descriptors (features) characterizing a given object is described in the paper by a vector of descriptors, where each entry of the vector determines a value of some feature of the object. In general, it is important to describe the object as completely as possible, which means by a large number of descriptors. This paper deals with a problem of selection of a proper subset of descriptors, which have the most substantial influence on the properties of the object, so that irrelevant descriptors could be excluded. For this purpose, we introduce a new method, Maximum Coverage Method (MCM). This method has been combined with optimization by a classical genetic algorithm. The described method is used for a data pre-processing, with the resulting selected features serving as an input for a neural network.

Keywords: Neural network, cluster, coverage, significant, shift, prediction, correctness, eliminating, separation

1 BASIC DESCRIPTION OF THE USED NEURAL NETWORK

One of the basic problems in every area of science is a search for a relation or a function between the structure of its objects and their properties. Let there be

a set of objects $\mathbf{A}$ and a set of their properties $\mathbf{B}$. Let there exist a function

$$\boldsymbol{F} : \mathbf{A} \to \mathbf{B} \tag{1}$$

which assigns a property $\boldsymbol{y} \in \mathbf{B}$ to each object $\boldsymbol{x} \in \mathbf{A}$. With such a function, it is possible to predict a property of an unknown object characterized by its descriptors. The ultimate goal is to construct a function in an analytical form, which describes the relation of properties of objects to their structure. Unfortunately, in many cases this goal is either unreachable, or reachable only with great difficulties. This drawback can be removed relatively successfully by using nonlinear mapping determined by multilayer neural networks.

During classification and prediction of objects with multilayer neural networks we start from the assumption that there exists a set of classified objects $\mathbf{A}$, which will be consequently divided into a training set $\mathbf{A}_{train}$ and a testing set $\mathbf{A}_{test}$, so that:

$$\mathbf{A} = \mathbf{A}_{train} \cup \mathbf{A}_{test}. \tag{2}$$

Let every object of the set $\mathbf{A}$ be classified by n features (descriptors) which means that we can assign n-dimensional vectors $\vec{\boldsymbol{d}}$ of descriptors to the objects of the set $\mathbf{A}$. These vectors are assigned both to objects from the set $\mathbf{A}_{train}$ and from the set $\mathbf{A}_{test}$:

$$\begin{aligned} \vec{\boldsymbol{d}}_{train} &= \left(d_{train}^{(1)}, d_{train}^{(2)}, \ldots, d_{train}^{(n)}\right) \\ \vec{\boldsymbol{d}}_{test} &= \left(d_{test}^{(1)}, d_{test}^{(2)}, \ldots, d_{test}^{(n)}\right). \end{aligned} \tag{3}$$

Let us presume that for each object $\boldsymbol{x} \in \mathbf{A}$ there exists a $\boldsymbol{y} \in \mathbf{B}$ so that $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{y}$, which means that to every vector $\vec{\boldsymbol{d}}$ a value $\boldsymbol{y} \in \mathbf{B}$ is assigned. Our ultimate goal is to find a function $\boldsymbol{F}$. The purpose of neural networks is to find such a function $\boldsymbol{G}(\boldsymbol{x}, \boldsymbol{w})$, which arguments are from the set $\mathbf{A}_{train}$ and $\boldsymbol{w}$ is such a parameter (or parameters) of mapping $\boldsymbol{G}$ that functional values $\boldsymbol{G}(\boldsymbol{x}, \boldsymbol{w})$ are as close as possible to the ideal results of the function $\boldsymbol{F}(\boldsymbol{x})$, i.e., to the required values. Let us define an objective function

$$E(\boldsymbol{w}) = \frac{1}{2} \sum_{i=1}^{|\mathbf{A}_{train}|} \left(\boldsymbol{G}(\boldsymbol{x}_i, \boldsymbol{w}) - \boldsymbol{y}_i\right)^2. \tag{4}$$

The requirement that the function $\boldsymbol{G}(\boldsymbol{x}, \boldsymbol{w})$ maps the objects $\boldsymbol{x}_{train}$ as close as possible to the required values $\boldsymbol{y}_{train}$ is realized by minimization of the function $E(\boldsymbol{w})$ with respect to the parameters $\boldsymbol{w}$. Let $\bar{w}$ be an optimal value of a parameter (parameters) $\boldsymbol{w}$ determined by the minimization process of the function $E(\boldsymbol{w})$

$$\bar{\boldsymbol{w}} = \arg \min_{w \in W} E(\boldsymbol{w}) \tag{5}$$

where $\boldsymbol{W}$ is a set (space) of admissible values of the parameter $\boldsymbol{w}$. By the function $\boldsymbol{G}(\boldsymbol{x}, \boldsymbol{w})$ (shortly $\boldsymbol{G}(\boldsymbol{w})$ an adapted neural network [2] simulates the original ideal

function $\boldsymbol{F}(\boldsymbol{x})$ for the values of the argument from the training set $\mathbf{A}_{train}$ on the basis of the minimization criterion (5).

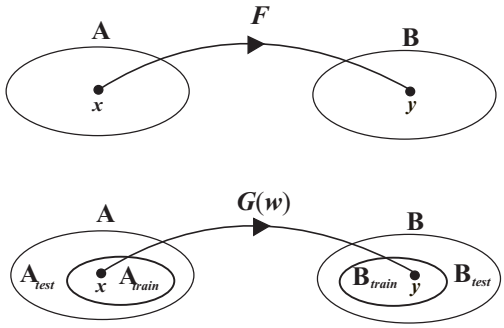

Fig. 1. Schematic description of the mapping $\boldsymbol{F} : \mathbf{A} \to \mathbf{B}$. By narrowing of this mapping onto the subset $\mathbf{A}_{train}$ we get a new model mapping $\boldsymbol{G}(\boldsymbol{w})$, whose function is determined by the parameter (parameters) $\boldsymbol{w}$

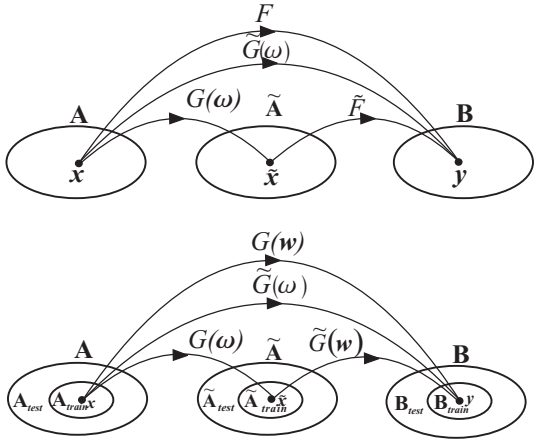

Fig. 2. If there exists such a function $\widetilde{\boldsymbol{G}}(\omega) : \mathbf{A} \to \mathbf{B}$, that $\boldsymbol{G}(\omega) : \widetilde{\boldsymbol{A}} \to \boldsymbol{A}$ with a parameter (parameters) ω and $\widetilde{\mathbf{G}}(\boldsymbol{w}) : \widetilde{\mathbf{A}} \to \mathbf{B}$, then the model mapping $\boldsymbol{G}(\boldsymbol{w})$ is replaceable by a (model) mapping $\widetilde{\boldsymbol{G}}(\omega)$ with a parameter (parameters) ω. Then the function F is replaced by three-layer neural network representing mapping $\widetilde{\boldsymbol{G}}(\omega) : \mathbf{A} \to \mathbf{B}$, where $\mathbf{A}$ is a set of target elements for the set $\mathbf{A}$ with a reduced number of descriptors.

The adapted neural net will be tested on the testing set, whether it approximates the ideal function learned from the training set. When the adapted function $\boldsymbol{G}(x, w)$ performs well also on the testing set, it is possible to use it also for prediction of properties of an unknown object $\boldsymbol{x}$ described by a corresponding vector of descriptors. The function $\boldsymbol{G}(x, w)$ which will be constructed suitably for the problem with

its parameter(s) $\boldsymbol{w}$ determined from a minimization of the objective function $E(\boldsymbol{w})$ so that it maps the training and testing set objects as close as possible to the required values, will be regarded as an analytical formula between the structure of studied objects and their properties, i.e., as the function $\boldsymbol{F}(\boldsymbol{x})$ (Figure 1).

Adaptation of a neural network (function $\boldsymbol{G}(x, w)$ is a rather difficult and computationally demanding process. Its complexity is influenced by several factors. One of them is the length of a vector of descriptors assigned to the objects from the set $\mathbf{A}$. This length corresponds to the number of features which describe these objects and on the basis of which they are classified. The features are in many cases introduced "ad-hoc" without a more detailed study of their interdependences and significance for the classification of the objects. In that case the function $\boldsymbol{G}(x, w)$ might have superfluous number of parameters $\boldsymbol{w}$, which is proportional to the size of a neural network and to the complexity of its adaptation process. The problem of excluding of descriptors, which do not have a substantial influence on the classification of objects can be transformed onto a problem of finding a function $\boldsymbol{G}(x, \omega)$ (shortly $\boldsymbol{G}(\omega)$ with a parameter (parameters) ω, which will map a set $\mathbf{A}$ onto a set $\widetilde{\mathbf{A}}$ so that every vector of descriptors (3a), resp. (3b) (Figure 3) will be mapped onto the vector

$$\begin{aligned} \widetilde{\boldsymbol{d}}_{train} &= \left(\tilde{d}^{(1)}_{train}, \tilde{d}^{(2)}_{train}, \ldots, \tilde{d}^{(m)}_{train}\right) \\ \widetilde{\boldsymbol{d}}_{test} &= \left(\tilde{d}^{(1)}_{test}, \tilde{d}^{(2)}_{test}, \ldots, \tilde{d}^{(m)}_{test}\right) \end{aligned} \tag{6}$$

while $m < n$. Then the functions $\boldsymbol{F}$ and $\boldsymbol{G}(\boldsymbol{w})$ are replaced by functions $\widetilde{\boldsymbol{F}}$ and $\widetilde{\boldsymbol{G}}(\boldsymbol{w})$, respectively (Figure 2), while the function $\boldsymbol{G}(\omega)$ reduces the number of descriptors of the objects from the set $\mathbf{A}$. This function will map an n-dimensional space $\boldsymbol{R}^n$ onto an r-dimensional space $\boldsymbol{R}^r$:

$$\boldsymbol{G}(\omega) : \boldsymbol{R}^n \to \boldsymbol{R}^r. \tag{7}$$

The subject of our following consideration will be construction of a function $\boldsymbol{G}(\omega)$ and finding its parameters ω so that the conditions

$$\forall \boldsymbol{x} \in \mathbf{A}_{train} : \widetilde{\boldsymbol{G}}(\omega) : \boldsymbol{x}_{train} \to \boldsymbol{y}_{train} \tag{8}$$

$$\forall \boldsymbol{x} \in \mathbf{A}_{test} : \widetilde{\boldsymbol{G}}(\omega) : \boldsymbol{x}_{test} \to \boldsymbol{y}_{test} \tag{9}$$

will be satisfied as fully as possible.

It is obvious that the requirement (9) is very demanding and it can be satisfied practically only to a certain precision. For the description of the function $\boldsymbol{G}(\omega)$ and finding its parameters we shall use a new MCM method (Maximum coverage method). The problem of excluding superfluous descriptors was solved by the KNN method (K Nearest Neighbour) [1].

2 THE BASIC PRINCIPLES OF MAXIMUM COVERAGE METHOD

Entities from a set A are entities from an n-dimensional left-open and right-closed interval $\mathbf{J} =]a^{(i)}, b^{(i)}]^n$, $i = 1, \ldots, n$, classified by an n-dimensional vector of descriptors $\vec{\boldsymbol{d}}_j = \left(d_j^{(1)}, d_j, \ldots, d_j^{(n)}\right)$ for each entity $x_j \in \mathbf{A}, j = 1 \ldots m$. $\mathbf{J} =]a^{(i)}, b^{(i)}]^n =]a^{(1)}, b^{(1)}] \times]a^{(2)}, b^{(2)}] \times \ldots \times]a^{(n)}, b^{(n)}] = J^{(1)} \times J^{(2)} \times \ldots \times J^{(n)}$. The number of entities $\boldsymbol{x} \in \mathbf{A}$ is m and the set $\mathbf{B}$ is a set of attributes of entities $\boldsymbol{x} \in \mathbf{A}$, and $|\mathbf{B}| < m$. The construction of intervals and end points $A(a^1, \ldots, an)$, $B(b^1, \ldots, b^n)$ of the interval $\mathbf{J}$ is as follows:

1. Make the sequence $d_1^{(i)} < d_2^{(i)} < \ldots < d_{m^{(i)}}^{(i)} \forall i = 1, \ldots, n$, when $m^{(i)} \leq m$.

$$m^{(i)} = m \Leftrightarrow \forall \boldsymbol{x}_p, \boldsymbol{x}_q \in \mathbf{A}; x_p^{(i)} \neq x_q^{(i)}, \forall p \neq q, p, q \in \{1, 2, \ldots, m\}, m = |\mathbf{A}|$$

2. $a_1^{(i)} < a_2^{(i)} < \ldots < a_{m^{(i)}-1}^{(i)} \forall i = 1, \ldots, n; a_j^{(i)} = \frac{d_{j+1}^{(i)} + d_j^{(i)}}{2} \forall j = 1, 2, .., m^{(i)} - 1$, $i = 1, \ldots, n$.
3. Let's put $a^i = a_0^{(i)} = d_1^{(i)} - \frac{d_2^{(i)} - d_1^{(i)}}{2}; b^i = a_{m^{(i)}}^{(i)} = d_{m^{(i)}}^{(i)} + \frac{d_{m^{(i)}}^{(i)} - d_{m^{(i)}-1}^{(i)}}{2}$, $i = 1, \ldots, n$.

This construction assigns just one n-dimensional interval C to $\forall \boldsymbol{x} \in \mathbf{A}$.

Definition 1. Let $\overline{N} = \{1, \ldots, n\}$. The set

$$\mathbf{D}^{(i)} = \left\{ a_j^{(i)} \in \left] a^{(i)}, b^{(i)} \right] ; i \in \bar{N}; j = 1, 2, \ldots, m_i - 1 \right\} \tag{10}$$

will be called a partition $\mathbf{D}^{(i)}$ of interval $\mathbf{J}^{(i)}$ for index j of the element $x_j \in \mathbf{A}$, $j = 1, \ldots, m$, $i = 1, \ldots, n$.

Definition 2. Let $\overline{N} = \{1, \ldots, n\}$,

$$\mathbf{J}_j^{(i)} = \left] a_{j-1}^i, a_j^i \right] ; i \in \bar{N}; j = 1, 2, \ldots, m_i \tag{11}$$

Then the set

$$X^{(i)} = \{\mathbf{J}_j^{(i)}; j = 1, 2, \ldots, m_i; i \in \overline{N}\} \tag{12}$$

will be called a disjunctive coverage of the interval $\mathbf{J}^{(i)} = \left] a^{(i)}, b^{(i)} \right]$.

Definition 3. Let us define the set $covC$ as a disjunctive coverage of an n-dimensional interval $\mathbf{J} = \left] a^{(i)}, b^{(i)} \right]^n$ with the following attribute:

$$covC = \left\{ C \subset \mathbf{J}; C = X_1 \times X_2 \times \ldots \times X_n, \forall X_1 \in X^{(1)}, X_2 \in X^{(2)}, \ldots, X_n \in X^{(n)} \right\} \tag{13}$$

Definition 4. The set $clC \in covC$ will be called the *cluster* of $\mathbf{A}$ if to each entity $\boldsymbol{x} \in C$, which is in $\boldsymbol{A}$, the same value $\boldsymbol{y} \in \mathbf{B}$ is assigned.

A non empty set $clC \subset covC$ with the previous attribute will be called the set of *clusters*.

When preparing data for neural network entry, the following attributes are important for its optimal training and prediction:

- the correctness of data
- eliminating the descriptors not significant for classification of $\boldsymbol{y} \in \mathbf{B}$
- optimal separation of the set $\mathbf{A}$ to $\mathbf{A}_{train} \cup A_{test}$.

Our goal is to introduce a method which will answer the following three problems.

Definition 5. The descriptor $d^{(i)}$ will be called non-significant if there is a set $covC$, for which: $|clC|$ is minimal and $\mathbf{D}^{(i)} = \emptyset$ for each entity $C \in clC$.

Figures 3 a) and 3 b) show points with two different values before and after constructing clusters, when descriptor $d^{(2)}$ is non-significant, i.e., the attributes of entities $\boldsymbol{x} \in \mathbf{A}$ in clusters can be described exactly by $d^{(1)}$.

Definition 6. Let the vectors of descriptors $\vec{\boldsymbol{d}}_i = \left(d_i^{(1)}, d_i^{(2)}, \ldots, d_i^{(n)}\right)$ and $\vec{\boldsymbol{d}}_i = \left(d_j^{(1)}, d_j^{(2)}, \ldots, d_j^{(n)}\right)$ characterize the attributes $\boldsymbol{y}_i$, respectively $\boldsymbol{y}_j$. Then the data consisting of entities in sets $\mathbf{A}$, $\mathbf{B}$ are correct only provided that for every two vectors of descriptors the equality $\vec{\boldsymbol{d}}_i = \vec{\boldsymbol{d}}_i$ implies $\boldsymbol{y}_i = \boldsymbol{y}_j$. Otherwise, the data are incorrect.

Remark. This data regularity definition means that data are regular only if there exists a function $F : \mathbf{A} \to \mathbf{B}$.

3 CONSTRUCTION OF CLUSTERS AND PURPOSE FUNCTION

To the entity $C_1 \in covC$ a value $\boldsymbol{y}_1 \in \mathbf{B}$ of a randomly chosen entity $\boldsymbol{x} \in \mathbf{A}$ is assigned only if $\boldsymbol{x} \in C_1$. Let us denote by c_1 the number of entities $\boldsymbol{x} \in C_1$ for which the value of the entity $\boldsymbol{x}$ does not equal $\boldsymbol{y}_1$.

To the entity $C_i \in covC$ a value $\boldsymbol{y}_i \in \mathbf{B}$ of the entity $\mathbf{x} \in \mathbf{A} - \bigcup_{j=1}^{i-1} C_j$ is assigned. Let c_i be the number of entities $\boldsymbol{x} \in C_i$ whose value does not equal $\boldsymbol{y}_i$.

C_k is the last cluster only if $\mathbf{A} - \bigcup_j = 1^k C_j = \emptyset$. The optimal construction of the set $covC$, as well as the clusters will be reached by optimization of the following function

$$f(\omega) = \sum_{i=1} c_i + \frac{\sum_{i=1}^{n} \left|\mathbf{D}^{(i)}\right|}{n \times \max_{i \in \{1,\ldots,n\}}(m^{(i)})} \tag{14}$$

provided that $a_j^{(i)} = \omega_j^{(i)} a_j^{(i)}$, $i = 1, \ldots, n$, $j = 1, \ldots, m^{(i)}$, where $\omega_j^{(i)}$ is a binary value, whose zero value means that the boundary point $a_j^{(i)}$ is deleted and the neighbouring intervals are joined. This in turn may affect the value of c_i.

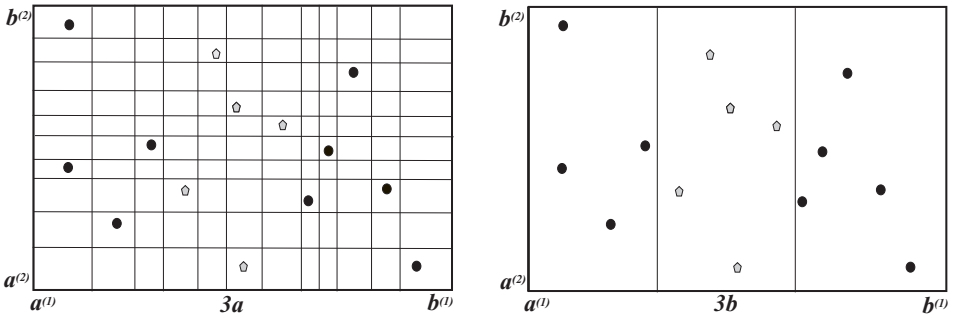

Fig. 3. Randomly generated points on 2-dimensional interval $[a^{(1)}, b^{(1)}] \times [a^{(2)}, b^{(2)}]$ with attributes $y_1 = , y_2 =$ that in some cases can be sorted to clusters, where each cluster does not depend on the i^{th} descriptor, i.e. $d^{(i)} \in [a^{(i)}, b^{(i)}]$ without any dividing points of this interval (3a). In the first stage of the optimization process each element $\boldsymbol{x}_i \in A$ is assigned to the corresponding cluster with attribute $\boldsymbol{y}_i \in \mathbf{B}$. In the first phase of the optimisation process, each element $\boldsymbol{x}_i \in \mathbf{A}$ is assigned to a cluster with a corresponding attribute $\boldsymbol{y}_i \in \mathbf{B}$. After the optimization (3b), the clusters formed by objects $\boldsymbol{x} \in \mathbf{A}$, which have the same attributes $\boldsymbol{y} \in \mathbf{B}$, are of the maximum size. This figure shows that the entities from set $\boldsymbol{x} \in \mathbf{A}$ may be sorted to clusters with identical attributes, where $d^{(2)}$ is non-significant from the view of adaptation process of the neural network. The attributes $\boldsymbol{y} \in \mathbf{B}$ are exactly determined by the descriptor $d^{(1)}$.

4 USING THE GENETIC ALGORITHMS (GA)

Let the population [3, 4] P be composed of p randomly generated mapping set vectors (chromosomes) $\omega^{(i)} \in \{0,1\}^{m^{(i)}-1} \forall i \in \overline{N}$. Let the objective function be given by equation. During iterations the population P changes through an application of crossover and mutation to its vectors. The selection of vectors to the process of crossover and further selection is realized by the so-called roulette, which means by a random selection of a chromosome with the probability proportional to its fitness F defined using the objective function by the formula

$$F(\omega) = \frac{\frac{1}{f(\omega)}}{\sum_{\omega} \in P \frac{1}{f(\omega)}}. \tag{15}$$

The crossover performs an exchange of randomly selected parts of two vectors chosen for a crossover process. The *crossover* operator is applied to 20 % of selected vectors; the remaining 80 % are not affected by the operation.

The *mutation* operator is applied to all mapping set vectors created by the *crossover*. The *mutation* randomly changes single entries of the vector, each with probability 0.002 %. After the operations of *crossover* and *mutation* are realized, a newly created population replaces the "parental" population of chromosomes (vectors of descriptors). Each iteration consists in calculation of objective function values

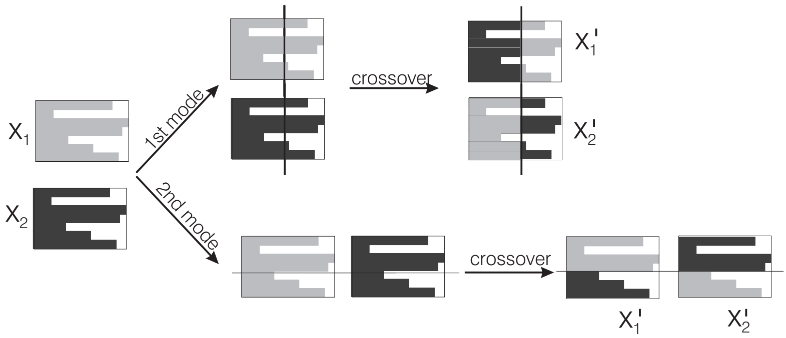

Fig. 4. Two types of crossover for chromosomes $\omega^{(i)} \in \{0,1\}^{m^{(i)}-1} \forall i \in \overline{N}$. Its initial values are generated randomly from the set $\{0,1\}$.

for all the members of the population, followed by replacing this parental population by a new population created by an application of the *crossover* and *mutation* operators to the vectors of the parental population.

The GA was terminated after 5 000 generations, when the best solution was usually achieved after 3 000 generations.

5 TESTING OF THE MCM METHOD

Testing of the MCM method is realized on a sample 100 randomly generated points in a plane, where all points $\boldsymbol{x}$ from the rectangles $]0,0.2]\times]0.1,0.8];]0.5,0.7]\times]0.4,0.9]$ have the value $\boldsymbol{y}=1$ and all points $\boldsymbol{x}$ from the rectangles $]0.2,0.5]\times]0,0.5];]0.7,1]\times]0.4,0.7]$ have the values $\boldsymbol{y}=2$. The population P is composed of 10 mapping sets $\omega^{(i)} \in \{0,1\}^{m^{(i)}-1} \forall i \in \overline{N}$. The crossover process is applied to 20 % of the selected vectors, the remaining 80 % is not changed by this processing of chromosomes. The mutation probability is $P_{mut}=0.01$. In the 3 072$^{\text{th}}$ iteration the best "mask" set vectors was discovered that guaranteed the existence of four clusters from the data set.

$$\begin{array}{lcl}]-0.0047, 0.2153] \times]0.0803, 0.8708]\,;]0.4959, 0.6999] \times]0.0803, 0.8708]\, y &=& 1 \\ ]0.2153, 0.4959] \times](0.0803, 0.8708)]\,;]0.6999, 0.9804] \times]0.0803, 0.8708]\, y &=& 2 \end{array}$$

The value of the objective function is $f(\omega)=0.04500000$.
[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1]... $d^{(1)}$
[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1]... $d^{(2)}$
It is obvious from the mapping set that for all clusters the extent of the descrip-

tor $d^{(2)}$ is the same and maximal, thus the clusters are practicable assigned with descriptor $d^{(1)}$.

6 ILLUSTRATIVE EXAMPLE

In [3] the authors dealt with construction of the neural network for the prediction of ^{13}C NMR (nuclear magnetic resonance) of chemical shifts of acyclic alkanes – saturated hydrocarbons. The chemical shift is a local property of carbon atoms; therefore, the referred carbon atom will be described by its closest neighbourhood (Figure 5). The used description of the neighbourhood consists in a creation of a vector describing a frequency of presence of some in advance chosen fragments in the alkane, i.e. how often the pertaining fragment appears in the alkane with the specified carbon atom.

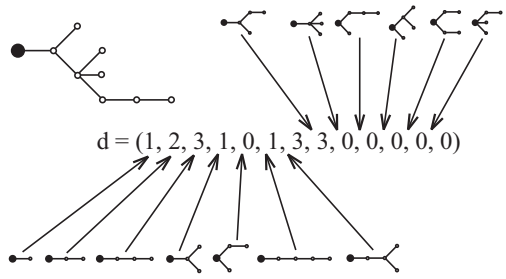

Fig. 5. Diagram of the description of a carbon atom denoted by a heavy dot in the molecule of the acyclic hydrocarbon. The neighborhood of this atom is described by a vector of descriptors $\vec{\boldsymbol{d}}$, containing 13 integers corresponding to the "frequencies of appearance" of the fragments shown at the ends of the arrows.

In order to create training and a testing set, this vector was enlarged by an experimentally measured value of the chemical shift. The objects of the set $\mathbf{A}$ from Figure 2 are now represented by vectors of descriptors $\vec{\boldsymbol{d}}$ and the set $\boldsymbol{B}$ is determined by the corresponding experimentally measured chemical shifts. A well adapted neural network approximates the function $\boldsymbol{F}$ (Figure 1). A question arises: "Do all the descriptors have an influence on the chemical shift of protons in carbon atom?" We shall try to get the answer by a maximum coverage method (MCM). The population was composed of 16 randomly generated mapping chromosomes, where No. 1 was generated with probability 0.24. The crossover process was applied to 20 % of the population and the mutation process was carried out with probability 0.021. After 464 iterations the value of the optimized function (14) is 5.32867133 and the resulting chromosome provides the following results.

1. Descriptors $d^{(5)}, d^{(9)}, d^{(10)}, d^{(11)}, d^{(13)}$ are non-significant in the meaning of Definition 5.

Fig. 6. The resulting chromosome, consisting of binary mapping 0, 1 determining the points of division $\mathbf{D}^{(i)}$ of the interval $\mathbf{J}^{(i)}$. The best result shows that $\mathbf{J}^{(5)}, \mathbf{J}^{(9)}, \mathbf{J}^{(10)}, \mathbf{J}^{(11)}, \mathbf{J}^{(13)}$ do not have (save the boundary points) any points of division, which means that the mapping of the elements $\boldsymbol{x} \in \mathbf{A}$ does not depend on the corresponding descriptors.

2. Descriptors $d^{(2)}, d^{(8)}$ also have a small significance for the clustering, since the whole interval is divided by one dividing point only.

7 COMPARISON OF RESULTS OF KNN AND MCM METHODS

Let us define weights $\boldsymbol{w}^{(i)}$ of significance of descriptors as

$$\boldsymbol{w}^{(i)} = \frac{\boldsymbol{m}_i - 2}{\boldsymbol{n}_i - 2}, \boldsymbol{n}_i > 2 \tag{16}$$

where $\boldsymbol{m}_i$ is the number of number of intervals $\mathbf{J}^{(i)}$ reduced according to the chromosome, where 0 entries do not define division of the interval and n_i is the number of all possible dividing points of the given interval.

$w^{(i)}$	1	2/6	4/5	2/5	0	3/4	1/2	1/7	0	0	0	1/2	0
ω	1	1	0	1	1	0	1	1	0	1	0	0	1

Fig. 7

The first row of the table shows weights of significance of descriptors calculated from results of the MCM method. The second row provides a "mask" vector of method KNN [4]. Descriptor with a value 1 of "mask" vector means its significance, descriptors with values equal to 0 are labeled as insignificant.

By comparison of the two methods we arrived at two descriptors selected as significant by both methods; MCM selected altogether 3 insignificant descriptors and KNN selected 5 insignificant descriptors. It is necessary to mention that a KNN method indicates sharp significance or insignificance only. The MCM method indicates weights of significance, while by comparing results we can say that d is selected

as insignificant by both methods. The results of both methods do not fit totally, but they are very similar.

8 TESTING OF THE MCM

The results presented below method were obtained from 30 test runs with the following set of parameter values:

- 16 randomly generated mapping chromosomes
- No. 1 was generated with probability 0.24
- the crossover process was applied to 20 % of the population
- the mutation process was carried out with probability 0.021

where $\bar{w}^{(i)}$ is the average of weights $w^{(i)}$. The average number of cycles necessary to obtain results was 549,56 and the average value of the optimized function is 5.333566434. It is apparent that in the meaning of Definition 5 the descriptors $d^{(5)}, d^{(9)}, d^{(13)}$ are non-significant for all the runs, $d^{(10)}, d^{(11)}$ were non-significant in 8 tests. Every test resulted in at least 4 nonsignificant descriptors. The best obtained result is shown in Figure 6.

i	1	2	3	4	5	6	7	8	9	10	11	12	13
$\bar{w}^{(i)}$	1	0.45	0.64	0.30	0	0.75	0.5	0.23	0	0.19	0.04	0.36	0

Fig. 8

9 CONCLUSIONS

The main result of the presented paper is a new method of selection of classifiers of elements, or in other words, exclusion of non-significant features from a set of descriptors, which enter as input values into a neural network classifier. The method solves significance of a property, and provides a possibility of reduction of the input vector of descriptors with a suitable setting of parameters. Moreover, the method solves correctness of data and selection of elements into the testing set.

The advantages of the proposed MCM method as compared with the well established KNN method are as follows:

1. The design of the object function is proposed in such a way that it shows the number of non-correct elements in the data set, which the KNN method cannot provide. If $f(\omega) < 1$, data are regular. When for example $f(\omega) = 5.333566434$ then there exist 5 copies of the vector of values of descriptors $\vec{\boldsymbol{d}}$, which differ in their functional values $\boldsymbol{y}$.
2. It follows from the principles of the MCM method (Figures 3 a), 3 b)) that there exists a function $\boldsymbol{F} : \mathbf{A} \to \mathbf{B}$, in an equivalent form $\boldsymbol{G}(\omega) : \boldsymbol{R}^{n-p} \to \boldsymbol{R}^{r}$,

where p is the number of non-significant features from a set of descriptors. The function $\boldsymbol{G}(\omega)$, which is able to hold the mapping $\boldsymbol{F} : \mathbf{A} \to \mathbf{B}$ while droping out non-significant descriptors. This means that the neural network shall work with a smaller number of descriptors leaving out 38 % of non-significant descriptors. This reduction will allow faster learning of the neural network with the same resulting quality of prediction.

3. The MCM method not only allows us to reduce the number of descriptors (unlike the KNN [4] method), but also, unlike KNN, enables the study of neural network optimisation for weighted input data.

Acknowledgement

This work was supported by Scientific Grant Agency Vega of Slovak Republic under grants 1/0071/09 and by Slovak Research and Development Agency APVV-0446-07.

REFERENCES

[1] Kvasnička, V.—Pospíchal, J.—Hesek, D.: Augmented Simulated Annealing Algorithm for the TSP. Central European Journal for Operations Research and Economics, Vol. 2, 1993, pp. 307–317.

[2] Mitchel, M.: An Introduction to Genetic Algorithms. MIT Press, Boston, MA 1996.

[3] Svozil, D.—Pospíchal, J.—Kvasnička, V.: Neural-Network Prediction of Carbon-13 NMR Chemical Shifts of Alkanes. J. Chem. Inf. Comp. Sci., Vol. 35, 1995, pp. 924–928.

[4] Boor, Š.—Kvasnička, V.—Pospíchal, J.: Feature Subset Selection by GA Combined with KNN Method. Mendel 2000, PC-DIR, Brno, 2000, ISBN 80-214-1609-2, pp. 407–412.

Štefan Boor graduated in mathematic and descriptive geometry from Faculty of Natural Sciences, Comenius University in Bratislava in 1980, and received the Ph. D. in theoretical and computational chemistry from Faculty of Chemical and Food Technologies, Slovak University of Technology, Bratislava in 2010. Now he is with the Department of Mathematics at the same Faculty as Assistant Professor. He has co-authored journal and conference papers related to fuzzy logic, evolutionary algorithms and neural networks.