FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 37 (2012) No. 4

DOI: 10.2478/v10209-011-0013-x

REVIEW OF THE EXTRACTION METHODS OF DNA MICROARRAY FEATURES BASED ON CENTRAL DECISION CLASS SEPARATION VS ROUGH SET CLASSIFIER

Piotr ARTIEMJEW *

Abstract. The study of DNA microarray gene extraction methods is an important and current area of research. Many researchers study gene ontological character, which contain significant information about symptoms of illnesses in tissues, types of organisms, and the distinguishing of some organisms' features. DNA microarray gene extraction methods allow us to choose the most significant genes for a given problem and some ways of their extraction. In this article, we aim to compare three methods of gene extraction. The first and second types are based on, respectively, the modified Fisher and F statistics methods. The last one is based on the novel experimental statistics called A. A common element of those three methods is the way in which we choose genes after the calculation of decision classes' separation ratio. Additionally, all three algorithms are based on the idea of central class separation from other decision concepts. We use our best 8v1.4 granular weighted voting classier as the basic element of comparison of our gene selection methods. The results of the research show that A statistics are better than other methods in all cases. In this article the best one is the SAM10 method, which works well for a small number of genes - less than one hundred. For a higher number of separated genes the SAM5 method is better - its effectiveness has been proven in recent published works.

Keywords: rough mereology, granular computing, rough sets, DNA microarrays, features extraction

*Department of Mathematics and Computer Science, University of Warmia and Mazury, Olsztyn, Poland, email:artem@matman.uwm.edu.pl

1 Introduction

We start with an introduction of the basic facts of DNA microarrays, the rough sets, and the weighted voting classifier - see [20], [4] - based on granular computing methods - see [14, 15, 19].

1.1 DNA Microarrays - Basic Information

The DNA microarray is a really useful molecular biology tool. We can place a large number of genes on a small plate, and check the gene expression profiles, among other things. The complementary DNA microarray used in this article is one of the most popular types of DNA microarrays, due to its low price compared with oligonucleotide DNA arrays. An interesting application, and the wider description of the DNA microarrays, can be found in [9], [10], [11], and [22]. The complementary DNA microarray technique is widely applied in genome sequencing - the recognition of the genes responsible for specific illnesses, etc. During the classification process each gene is treated as an *attribute*, and its value is the intensity of bond of DNA array. Due to the large number of attributes in the DNA arrays it is quite difficult to apply the most effective classification algorithms. In this article we apply the weighted voting classifier based on residual rough inclusions, proposed in [4], and [20] in order to compare the best gene extraction methods based on the modified Fisher method, F statistics and experimental A statistics.

1.2 Rough Sets Background - In a Nutshell

In the rough set theory, data are stored in the form of information, or decision systems, where the first one is defined as a pair (U, A), for U as a set of *objects*, and A as an attribute set. The decision system is defined as a triple (U, A, d); where d fulfils the condition $d \notin A$ of the decision attribute. An 'information set of object x' of the decision system is defined as follows:

$$Inf_A(x) = \{(a = a(x)) : a \in A\}, \tag{1}$$

An individual attribute of object x with value a(x) defines a descriptor $(a = a(x))$, commonly used in the short form $(a = v)$, where $a \in A \cup \{d\}$.

In the descriptor notation, the decision rules derived from object x can be described as,

$$(a_1 = a_1(x)) \wedge (a_2 = a_2(x)) \wedge \ldots \wedge (a_k = a_k(x)) \Rightarrow (d = d(u)) \tag{2}$$

where the set of conditional attributes $\{a_1, a_2, ..., a_k\}$ is the subset of A.

In the classic meaning, the granulation of knowledge in information, or decision systems consists in partitioning the universe of objects U into the elementary granules of the form

$$[x]_A = \{y \in U : a(x) = a(y), \forall a \in A\}, \text{ where the central object } x \in U \tag{3}$$

The collection of elementary granules are called The Granules of Knowledge. The granulation in this sense consists in forming the aggregates of objects, which are indiscernible from the sets of conditional attributes.

The relation $\mu \subseteq U \times u \times [0,1]$ is the formal definition of the rough inclusion in the sense of [14, 15]. In short, it can be formulated by saying '*an object x is a part of an object y to a degree of r*'.

The father of granular computing, Professor Zadeh [24], proposed to replace individual objects by 'clumps of objects'. The objects were collected together by type, and as aggregates were used for computing.

In our approach, the granules are defined based on the rough inclusions in the form proposed in [15, 19].

For the rough inclusion μ, an object u, and a granulation radius $r \in [0,1]$, a granule $g(u,r)$ is defined as $g(u,r) = \{v \in U : \mu(v,u,r)\}$, in detail

$$g(u,r) = \{v \in U : \frac{card\{IND(u,v)\}}{card\{A\}|} \geq r\} \tag{4}$$

$$\text{where, } IND(u,v) = \{a \in A : a(u) = a(v)\} \tag{5}$$

The described granules, $g(u,r)$, are computed for all objects $u \in U$, and applied in the classification process with radius r equal 1.

In the next step we define basic t-norms and the residual rough inclusion based on residuum of t-norm in the terms of decision systems. The function

$$T : [0,1] \times [0,1] \to [0,1], \tag{6}$$

which is symmetrical, associative, increasing in each coordinate, and subject to boundary conditions: $T(x,0) = 0, T(x,1) = x$, e.g, [12] is a t-norm.

The best known t–norms are the Łukasiewicz, Product, and minimum t–norm, defined as $L(x,y) = max\{0, x+y-1\}$, $Prod(x,y) = x \cdot y$, and $min(x,y) = min\{x,y\}$ respectively.

The equivalence,

$$x \Rightarrow_T y \geq r \text{ if and only if } T(x,r) \leq y \tag{7}$$

describes a *residuum* $x \Rightarrow_T y$ of a t–norm T.

For continuous t–norms $L, Prod$, and min, the residual implication is given by the formula,

$$x \Rightarrow_T y = max\{r : T(x,r) \leq y\} \tag{8}$$

normalized at the interval $[0,1]$ - see the survey [19] - looks as

$$\mu_T(x,y,r) \text{ if and only if } x \Rightarrow_T y \geq r \tag{9}$$

In the next subsection we introduce the idea of our decision assignment algorithm in terms of rough set theory.

1.3 A Voting Scheme by Residual Rough Inclusion

In our approach the rough inclusion induced by the Łukasiewicz t–norm L, is applied in the classifier synthesis.

For an object u from the test decision system, and objects v from the training base of knowledge, we assign the decision class to the test object based on the following weights: $w(v,u,\varepsilon) = dis_{\varepsilon}(u,v) \Rightarrow_T ind_{\varepsilon}(u,v)$, where ε-discernibility, and ε-indiscernibility parameters are defined as follows,

$$dis_{\varepsilon}(u,v) = \frac{|\{a \in A : ||a(u) - a(v)|| \geq \varepsilon\}|}{|A|} \tag{10}$$

$$ind_{\varepsilon}(u,v) = \frac{|\{a \in A : ||a(u) - a(v)|| < \varepsilon\}|}{|A|} \tag{11}$$

The decision class v_d with the minimal value of parameter,

$$Param(v_d) = \sum_{\{v \in U_{trn} : d(v) = v_d\}} w(v,u,\varepsilon) \tag{12}$$

is assigned into the the classified test object u.

Having described the decision value assignment method we can apply this algorithm along the lines of [20], [4].

2 Rough Set Weighted Voting Classifier Based on Granules of Training Objects

The general idea of our weighted voting classifiers consists of dynamic changes of weights during classification depending on the distance between descriptors of training and test objects. This kind of classification, especially one used in this work 8v1.4 classifier seems to reduce the overfitting phenomena during classification by a slight disturbance of proper classification - [20], [1, 2, 4].

General way of classification by 8_v1.4 method - [20], [1, 2, 4] - appears as follows,

Step 1. We choose the training decision system (U_{trn}, A, d), and the test decision system (U_{tst}, A, d),

Step 2. We search for the maximal and the minimal values of attributes a on the training set, and mark them as max_attr_a, and min_attr_a respectively.

Step 3. We fix an attribute similarity ratio as ε.

Step 4. The test objects are classified in the following way.

For $\forall\ a \in A$, training objects $v_p \in U_{trn}$, for $p \in \{1, ..., card\{U_{trn}\}$, and the test objects $u_q \in U_{tst}$, where $q \in \{1, ..., card\{U_{tst}\}$ we compute

(i) If $\frac{|a(u_q)-a(v_p)|}{max\_attr_a - min\_attr_a} \geq \varepsilon$, then

$$w(u_q, v_p) = w(u_q, v_p) + \frac{|a(u_q) - a(v_p)|}{(max\_attr_a - min\_attr_a) * (\varepsilon + \frac{|a(u_q)-a(v_p)|}{max\_attr_a - min\_attr_a})} \text{ i. e.,} \tag{13}$$

$$w(u_q, v_p) = w(u_q, v_p) + \frac{|a(u_q) - a(v_p)|}{(max\_attr_a - min\_attr_a) * \varepsilon + |a(u_q) - a(v_p)|} \tag{14}$$

(ii) If $\frac{|a(u_q)-a(v_p)|}{max\_attr_a - min\_attr_a} < \varepsilon$, then

$$w(u_q, v_p) = w(u_q, v_p) + \frac{|a(u_q) - a(v_p)|}{(max\_attr_a - min\_attr_a) * \varepsilon}$$

If the weights between u_q test object and all v_p training objects are computed then we start the voting procedure by means of the following parameters,

$$Param(v_d) = \sum_{\{v_p \in U_{trn} : d(v_p) = v_d\}} w(u_q, v_p), \tag{15}$$

Lastly, the v_d concept with minimal value of the parameter $Param(v_d)$ is assigned to u_q test object.

2.1 The Result Validation Method

To validate results in this article we have used a resampling method called Leave One Out (LOO). The motivation to use the LOO method is to be found, among other places in [13]. This article proves the effectiveness and almost unbiased character of this method.

We have introduced basic facts about our approach and now we return to our analysis of DNA microarrays.

3 DNA Microarray Features Extraction Methods

3.1 The Main Motivation

Our general goal is the comparison of gene extraction methods based on Fisher distance and F, A statistics - by using a classifier based on mereological granules. We would like to find the best method among those studied and identify the numbers of the genes separating the decision classes with the highest rate, and give the best

classification results. The genes which we have found can be used for ontological analysis, but our methods of gene extraction do not take into account the ontological sense of separated genes. The context of data doesn't matter either.

The high number of genes in comparison with the number of objects can cause a problem with overfitting. For this reason, we need some extraction methods which can point us towards smaller groups of genes which, as a decision system, can effectively classify samples of data. It is time to show our propositions of gene extraction methods based on the central decision class separation.

In the first algorithm (MFM1), we have chosen the most characteristic genes which best differentiate decision classes. An application of modified Fisher method is the basic element of this algorithm.

3.2 Feature Extraction Method Based on Modified Fisher Method - Case 1 (MFM1)

For the decision system (U, A, d), where $U = \{u_1, u_2, ..., u_n\}, A = \{a_1, a_2, ..., a_m\}$, $d \notin A$, classes of d: $c_1, c_2, ..., c_k$, we propose to obtain the rate of separation of the gene $a \in A$ for decision class c_i, $i = 1, 2, ..., k$ in the following way. We let,

$$S^{c_i}(a) = \frac{(\overline{C}_i^a - \hat{C}_i^a)^2}{Z_{\overline{C}_i^{a^2}} + Z_{\hat{C}_i^{a^2}}}, a \in A. \tag{16}$$

where,

$$C_i^a = \{a(u) : u \in U \ and \ d(u) = c_i\}. \tag{17}$$

$$\overline{C}_i^a = \frac{\{\sum a(u) : u \in U \ and \ d(u) = c_i\}}{card\{C_i^a\}}, \hat{C}_i^a = \frac{\{\sum a(v) : v \in U \ and \ d(v) \neq c_i\}}{card\{U\} - card\{C_i^a\}}. \tag{18}$$

$$Z_{\overline{C}_i^{a^2}} = \frac{\sum_{a(u) \in C_i^a} (a(u) - \overline{C}_i^a)^2}{card\{C_i^a\}}, Z_{\hat{C}_i^{a^2}} = \frac{\sum_{a(v) \in U \setminus C_i^a} (a(v) - \hat{C}_i^a)^2}{card\{U\} - card\{C_i^a\}} \tag{19}$$

After the rate of the separation, $S^{c_i}(a)$ is computed for all genes $a \in A$ and all decision classes c_i; genes are sorted in the increasing order of $S^{c_i}(a)$,

$$S_1^{c_1}(a) > S_2^{c_1}(a) > ... > S_{card\{A\}}^{c_1}(a)$$

$$S_1^{c_2}(a) > S_2^{c_2}(a) > ... > S_{card\{A\}}^{c_2}(a)$$

$$\vdots$$

$$S_1^{c_k}(a) > S_2^{c_k}(a) > ... > S_{card\{A\}}^{c_k}(a)$$

Finally, we choose for experiments the fixed number of genes from the sorted list by means of the procedure,

Procedure
Input data
$A' \leftarrow \emptyset$
$iter \leftarrow 0$
for i=1,2,...,card{A} **do**
  for j=1,2,...,k **do**
    $S^{c_j}(a) = S_i^{c_j}(a)$
    if $a \notin A'$ **then**
      $A' \leftarrow a$
      $iter \leftarrow iter + 1$
      if $iter = fixed\ number\ of\ the\ best\ genes$ **then**
        BREAK
      end if
    end if
  end for
  if $iter = fixed\ number\ of\ the\ best\ genes$ **then**
    BREAK
  end if
end for
return A'

The next algorithm (MSF4) has similar motivation to the previous one. However, we applied here F statistics, extended on multiple decision classes, well known for separation of the two decision classes.

3.3 Feature Extraction Method Based on Modified F Statistics Method - Case4 (MSF4)

In this case the rate of separation at the anologous assumptions as MFM1 is defined as follows,

$$F_{c_i}(a) = \frac{MSTR_{c_i}(a)}{MSE_{c_i}(a)} \tag{20}$$

$$C_i^a = \{a(u) : u \in U \ and \ d(u) = c_i\}$$

$$\overline{C}_i^a = \frac{\{\sum a(u) : u \in U \ and \ d(u) = c_i\}}{card\{C_i^a\}},\ \hat{C}_i^a = \frac{\{\sum a(v) : v \in U \ and \ d(v) \neq c_i\}}{card\{U\} - card\{c_i\}}$$

$$MSTR_{c_i}(a) = card\{C_i^a\} * (\bar{C}_i^a - \hat{C}_i^a)^2$$

$$MSE_{c_i}(a) = \frac{\sum_{j=1}^{card\{C_i^a\}}(a(u_j) - \bar{C_i^a})^2}{card\{C_i^a\}}, \text{ where } u_j \in C_i^a, i = 1, 2, ..., card\{C_i^a\}$$

After the rate of the separation, $F^{c_i}(a)$ is computed for all genes $a \in A$ and all decision classes c_i; genes are sorted in the decreasing order of $F^{c_i}(a)$,

$$F_1^{c_1}(a) > F_2^{c_1}(a) > ... > F_{card\{A\}}^{c_1}(a)$$

$$F_1^{c_2}(a) > F_2^{c_2}(a) > ... > F_{card\{A\}}^{c_2}(a)$$

$$\vdots$$

$$F_1^{c_k}(a) > F_2^{c_k}(a) > ... > F_{card\{A\}}^{c_k}(a)$$

Finally, we choose for experiments the fixed number of genes from the sorted list by means of the procedure,

Procedure
Input data
$A' \leftarrow \emptyset$
$iter \leftarrow 0$
for i=1,2,...,card{A} **do**
  for j=1,2,...,k **do**
    $F^{c_j}(a) = F_i^{c_j}(a)$
    if $a \notin A'$ **then**
      $A' \leftarrow a$
      $iter \leftarrow iter + 1$
      if $iter = fixed\ number\ of\ the\ best\ genes$ **then**
        BREAK
      end if
    end if
  end for
  if $iter = fixed\ number\ of\ the\ best\ genes$ **then**
    BREAK
  end if
end for
return A'

An idea for modifying the above approach was suggested by Professor Polkowski. We thought how to extract genes by means of the distance between gene attribute values and the distance between gene values and an average value for a considered decision class or the rest of decision classes.

Table 1: An information table of the examined data sets - see [23]; data1 = anthracyclineTaxaneChemotherapy, data2 = BurkittLymphoma, data3 = HepatitisC, data4 = mouseType, data5 = ovarianTumour, data6 = variousCancers_final

Data	*No.attr*	*No.obj*	*No.class*	*The.dec.class.details*
*data*1	61359	159	2	1(59.7%), 2(40.2%)
*data*2	22283	220	3	3(58.1%), 2(20%), 1(21.8%)
*data*3	22277	123	4	2(13.8%), 4(15.4%), 1(33.3%), 3(37.3%)
data4	45101	214	7	3(9.8%), 2(32.2%), 7(7.4%), 6(18.2%), 5(16.3%), 4(9.8%), 1(6%)
*data*5	54621	283	3	3(86.5%), 1(6.3%), 2(7%)
data6	54675	383	9	3(6.2%), 2(40.4%), 4(10.1%), 7(5.2%), 5(12.2%), 6(10.9%), 8(4.1%), 9(4.6%), 10(5.7%)

3.4 Feature Extraction Method Based on A Statistics - Case10 (SAM10)

For this method we defined the decision system as (U, B, d), where $U = \{u_1, u_2, ..., u_n\}$, $B = \{a_1, a_2, ..., a_m\}$, and $d \notin B$, $d \in \{c_1, c_2, ..., c_k\}$, we propose to obtain the rate of separation of the gene $a \in A$ for decision class c_i, where $i = 1, 2, ..., k$ in the following way. We let,

$$A_{c_i}(a) = C_i^a \wedge_\varepsilon \{U \backslash C_i^a\}, \tag{21}$$

$$C_i^a = \{a(u) : u \in U \text{ and } d(u) = c_i\}, \hat{C}_i^a = \frac{\{a(v) : v \in U \text{ and } d(v) \neq c_i\}}{card\{U\} - card\{C_i^a\}}, \tag{22}$$

where,

$$C_i^a \wedge_\varepsilon \{U \backslash C_i^a\} = \frac{card\{a(u) \in C_i^a : \exists a(v) \in \{U \backslash C_i^a\}; \frac{|a(u)-a(v)|}{train_a} \leq \varepsilon\} + card\{a(v) \in \{U \backslash C_i^a\} : \exists a(u) \in C_i^a; \frac{|a(u)-a(v)|}{train_a} \leq \varepsilon\}}{card\{U\}} -$$

$$\frac{card\{a(u) \in C_i^a : \frac{|a(u)-\hat{C}_i^a|}{train_a} > \varepsilon\}}{card\{C_i^a\}},$$

where, $train_a = max\_attr_a - min\_attr_a$, $a \in B$.

After the rate of the separation $A^{c_i}(a)$ are computed for all genes $a \in B$ and all decision classes c_i, genes are sorted in the increasing order of $A^{c_i}(a)$,

$$A_1^{c_1}(a) < A_2^{c_1}(a) < ... < A_{card\{B\}}^{c_1}(a)$$

$$A_1^{c_2}(a) < A_2^{c_2}(a) < ... < A_{card\{B\}}^{c_2}(a)$$

$$\vdots$$

$$A_1^{c_k}(a) < A_2^{c_k}(a) < ... < A_{card\{B\}}^{c_k}(a)$$

Finally, we choose for experiments the fixed number of genes from the sorted list by means of the procedure,

Procedure
Input data
$B' \leftarrow \emptyset$
$iter \leftarrow 0$
for i=1,2,...,card{B} **do**
  for j=1,2,...,k **do**
    $A^{c_j}(a) = A_i^{c_j}(a)$
    if $a \notin B'$ **then**
      $B' \leftarrow a$
      $iter \leftarrow iter + 1$
      if $iter = fixed\ number\ of\ the\ best\ genes$ **then**
        BREAK
      end if
    end if
  end for
  if $iter = fixed\ number\ of\ the\ best\ genes$ **then**
    BREAK
  end if
end for
return B'

The results of our algorithm with real data sets, see [23], are reported in the next section.

4 The results of our research on real data sets

One of the most common parameters which are used for evaluation of data with unbalanced decision classes in the cardinality sense (see examined data sets in Tab. 1) are balanced accuracy and balanced coverage, whose definitions appear in the equation 23 and 24 respectively.

$$Balanced.acc = \frac{acc_{c_1} + acc_{c_2} + ... + acc_{c_k}}{k}, \tag{23}$$

$$Balanced.cov = \frac{cov_{c_1} + cov_{c_2} + ... + cov_{c_k}}{k}. \tag{24}$$

For clarity, the average results of classification presented in Tables 2 are the average values of balanced accuracy from all examined data sets.

In order to show our results in a more objective way, we use for our 8_v1.4 classification algorithm only one value of epsilon $\varepsilon = 0.01$. We have carried out Leave One Out experiments with real DNA microarray data from the Tuned It platform

Table 2: Leave One Out; Average balanced accuracy of classification for implemented methods; Examined data sets: anthracyclineTaxaneChemotherapy, BurkittLymphoma, HepatitisC, mouseType, ovarianTumour, variousCancers_final; No.of.genes = number of classified genes, method = method's name

$method \backslash No.of.genes$	10	20	50	100	200	500	1000
$MFM1$	0.746	0.764	0.79	0.796	0.795	0.794	0.782
$MSF4$	0.651	0.655	0.699	0.722	0.751	0.769	0.776
$SAM10$	**0.781**	**0.817**	**0.822**	0.835	0.83	0.824	0.808
$SAM5$ [1]	0.718	0.77	0.815	**0.841**	**0.84**	**0.846**	**0.833**
$MSF6$ [2]	0.718	0.759	0.789	0.782	0.781	0.777	0.783

described in details in the Table 1. To achieve a proper computation of accuracy for the LOO method it is necessary to build a confusion matrix, with the assumption that objects from all folds are treated as one decision system.

The average of balanced accuracy for all examined methods, in comparison with recently studied methods SAM5 (see [1]) and MSF6 (see [2]), is shown in Table 2. As we can see for the small number of genes (less than 100), the best method is SAM10, and starting from 100 genes to 1000 genes the best is the SAM5 method. Those two algorithms are unrivalled with other presented methods, and are from 3 to 6 percent better.

5 Conclusions

The results of our research showed beyond a shadow of a doubt the vast advantage of SAM10 and SAM5 methods over the remaining gene separation methods. Those results have been confirmed by average results of balanced accuracy. It turns out that the SAM10 method works best for a small number of genes in the range of 10, 20 and 50. Its characteristic is to decrease the product of a given class with the remaining classes by discernibility degree of central class elements from the average of the other classes. Contrary to this, the SAM5 method makes use of lowering the value of product weight of a pair of decision classes by the indiscernibility degree of elements of a given class from an average value of paired decision classes, which is a characteristic element of this method. It works best for a large number of genes in the range of 100, 200, 500, and 1000. The essential element of these methods is the way of choosing the best genes after their calculation. The main difference is the general approach to gene separation; in the SAM10 method we have the separation of the central class from all other classes, but in the SAM5 method there is a separation of pairs of decision classes.

In our future research we are going to examine to what extent the analyzed gene separation methods depend on the information content of particular DNA microarrays.

6 Acknowledgements

The research has been supported by grant 1309-802 from Ministry of Science and Higher Education of the Republic of Poland.

References

[1] Artiemjew P.: The Extraction Method of DNA Microarray Features Based on Experimental *A* Statistics. In: J.T. Yao et al. (eds.) RSKT 2011, Banff, Canada, LNCS, Springer, Heidelberg, vol. 6954, 2011, 642–648.

[2] Artiemjew P.: The Extraction Method of DNA Microarray Features Based on Modified *F* Statistics vs Classifier Based on Rough Mereology. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Ras, Z., W. (eds.) ISMIS 2011, Warsaw, Poland, LNAI, Springer, Heidelberg, vol. 6804, 2011, 33–42.

[3] Artiemjew P.: Classifiers based on rough mereology in analysis of DNA microarray data, in: *Proceedings 2010 IEEE International Conference on Soft Computing and Pattern Recognition SocPar'10*, Sergy Pontoise France, IEEE Press, 2010.

[4] Artiemjew P.: On strategies of knowledge granulation and applications to decision systems, PhD Dissertation, Polish Japanese institute of Information Technology, L. Polkowski, Supervisor, Warsaw, 2009.

[5] Artiemjew P.: Rough mereological classifiers obtained from weak rough set inclusions, in: Proceedings of Int. Conference on Rough Set and Knowledge Technology RSKT'08, Chengdu China, LNAI, Springer Verlag, Berlin, vol. 5009, 2008, 229–236.

[6] Artiemjew P.: On classification of data by means of rough mereological granules of objects and rules, in: Proceedings of Int. Conference on Rough Set and Knowledge Technology RSKT'08, Chengdu China, LNAI, Springer Verlag, Berlin, vol. 5009, 2008, 221–228.

[7] Artiemjew P.: Natural versus granular computing: Classifiers from granular structures, in: Proceedings of 6th International Conference on Rough Sets and Current Trends in Computing RSCTC'08, Akron, Ohio, USA, Springer Berlin / Heidelberg, vol. 5306, 2008, 150–159.

[8] Artiemjew P.: Classifiers from granulated data sets: Concept dependent and layered granulation, in: *Proceedings RSKD'07. Workshop at ECML/PKDD'07*, Warsaw Univ. Press, Warsaw, 2007, 1–9.

[9] Brown M., Grundy W., et al.: Knowledge-based analysis of microarray gene expression data by using support vector machines, University of California, 1999.

[10] Eisen MB, Brown PO: DNA arrays for analysis of gene expression. Methods Enzymol 303, 1999, 179–205.

[11] Furey T.S., Cristianini, Duffy N., Bernarski, Schummer M., Haussler D.: "Support Vector Machine Classification and Validation of Cancer Tissue Samples Using Microarray Expression Data," Bioinformatics, vol. 16, 2000, 906–914.

[12] H*á*jek P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht, 1998.

[13] Molinaro A.M., Simon R., Pfeiffer R.M.: Prediction error estimation: a comparison of resampling methods, in: *Bioinformatics, vol. 21, issue 15, Oxford University Press, Oxford, UK*, 2005, 3301–3307.

[14] Polkowski L.: Toward rough set foundations. Mereological approach (a plenary lecture), in: *Proceedings RSCTC04*, Uppsala, Sweden, 2004, LNAI, Springer Verlag, Berlin, vol. 3066, 2004, 8–25.

[15] Polkowski L.: Formal granular calculi based on rough inclusions (a feature talk), in: *Proceedings 2005 IEEE Int. Confrence on Granular Computing GrC'05*, IEEE Press, 2005, 57–62.

[16] Polkowski L.: Formal granular calculi based on rough inclusions (a feature talk), in: *Proceedings 2006 IEEE Int. Conference on Granular Computing GrC'06*, IEEE Press, 2006, 57–62.

[17] Polkowski L.: Granulation of knowledge in decision systems: The approach based on rough inclusions. The Method and its applications (plenary talk), in: Lecture Notes in Artificial Intelligence (Proceedings RSEiSP'07), Springer Verlag, Berlin, vol. 4585, 2005, 69–79.

[18] Polkowski L.: The paradigm of granular rough computing, in: *Proceedings ICCI'07. 6th IEEE Intern. Conf. on Cognitive Informatics*, IEEE Computer Society, Los Alamitos, CA, 2007, 145–163.

[19] Polkowski L.: A Unified Approach to Granulation of Knowledge and Granular Computing Based on Rough Mereology: A Survey, in: Handbook of Granular Computing, Witold Pedrycz, Andrzej Skowron, Vladik Kreinovich (Eds.), John Wiley & Sons, New York, 2008, 375–401.

[20] Polkowski L., Artiemjew P.: On classifying mappings induced by granular structures. Transactions on Rough Sets IX. Lecture Notes in Computer Science, Springer Verlag, Berlin, vol. 5390, 2008, 264–286.

[21] Polkowski L., Artiemjew P.: A study in granular computing: On classifiers induced from granular reflections of data, Transactions on Rough Sets IX. Lecture Notes in Computer Science, Springer Verlag, Berlin, vol. 5390,2008, 230–263.

[22] Schena M.: Microarray analysis. Wiley, Hoboken, NJ, USA, 2003.

[23] http://tunedit.org/repo/RSCTC/2010/A

[24] Zadeh L.A.: Fuzzy sets and information granularity. In: Gupta, M., Ragade, R., Yager, R.R.(Eds.): Advances in Fuzzy Set Theory and Applications. North Holland, Amsterdam, 1979, 3–18.

Presented at the Congress of Young IT Scientists, Międzyzdroje, Poland, 20–22.09.2012