The Extraction Method of DNA Microarray Features Based on Modified F Statistics vs. Classifier Based on Rough Mereology

Piotr Artiemjew

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

Abstract. The paradigm of Granular Computing has emerged quite recently as an area of research on its own; in particular, it is pursued within the rough set theory initiated by Zdzisław Pawlak. Granules of knowledge can be used for the approximation of knowledge. Another natural application of granular structures is using them in the classification process. In this work we apply the granular classifier based on rough mereology, recently studied by Polkowski and Artiemjew 8_v1_w4 algorithm in exploration of DNA Microarrays. An indispensable element of the analysis of DNA microarray are the gene extraction methods, because of their high number of attributes and a relatively small number of objects, which in turn results in overfitting during the classification. In this paper we present one of our approaches to gene separation based on modified F statistics. The modification of F statistics, widely used in binary decision systems, consists in an extension to multiple decision classes and the application of a particular method to choose the best genes after their calculation for particular pairs of decision classes. The results of our research, obtained for modified F statistics, are comparable to, or even better than, the results obtained in other methods with data from the Advanced Track of the recent DNA Microarray data mining competition.

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

1 Introduction

The last few years have seen a growing interest in the exploration of DNA microarrays; the more so due to some meaningful competitions, see [17]. A number of researchers have attempted to find effective gene extraction methods and classifiers in order to predict particular scientific problems. An exemplary application can be the ability to detect some illnesses, or predict vulnerability to some diseases, and distinguish some organisms' features or types. The main motivation to

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 33–42, 2011.

use our granular methods in DNA microarray exploration was our participation in the discovery challenge, see [17] at TunedIt platform. Our algorithm, based on the modified Fisher method [7] with our weighted voting classifier 8_v1_w4 see [14], reached eighteenth place on the basic track of this competition, and was worse only by 3.491 per cent balanced accuracy than the winner. Since that time we have been carrying out intensive research on new methods of gene extraction, and we have created more than 20 new methods of gene extraction. One of the effects of our work was the idea of DNA gene extraction methods based on modified F statistics.

This work is devoted to the classification problems of DNA arrays, with the use of the methods of granular computing presented in [12], [13]. In the Introduction we briefly show the idea of DNA microarrays, and define the basic concepts of granular computing in the sense of Polkowski , op.cit., and recall the idea of granular classification ([12], [13], [14]).

1.1 DNA Microarrays

The complementary DNA microarray is the most commonly used type of DNA microarrays, which is cheaper than other types of medical microarrays. The basic information about DNA microarray can be found in [4], [5], [6] and [15].

The DNA microarray is a tool which provides the means of measuring the gene expression on a mass scale, by simultaneously examining up to 40000 DNA strands with respect to their hybridization with complementary DNA (cDNA).

This analysis technique is widely applied in genome sequencing, for example the recognition of genes responsible for specific illnesses, etc. From the classification point of view, each gene can be regarded as an *attribute* and its value is the intensity of the bond with cDNA. A large number of attributes calls for new methods of data analysis, and in this paper we apply methods of granular classification, especially the method of weight incrementation in weighted voting by residual classifiers, as proposed in [2] and [14].

1.2 Basic Notions of Rough Set Theory and Granular Theory

In the light of rough set theory, knowledge can be represented by means of information or decision systems. An *information system* is defined as a pair (U, A) where U is a universe of *objects*, and A is a set of *attributes*; a *decision system* is defined as triple (U, A, d) where $d \notin A$ is a *decision*. Objects in u are represented by means of information sets: $Inf_A(u) = \{(a = a(u)) : a \in A\}$ is the *information set* of the object u; the formula $(a = a(u))$ is a particular case of a *descriptor* of the form $(a = v)$ where v is a value of the attribute $a \in A \cup \{d\}$.

Decision rules are expressions of the form

$$\bigwedge_{a \in A} (a = a(u)) \Rightarrow (d = d(u)) \tag{1}$$

In the classic meaning, the granulation of knowledge in information/decision systems consists of partitioning the set of objects U into classes of the *indiscernibility relation*

$$IND(A) = \{(u, v) : a(u) = a(v) \text{ for each } a \in A\} \tag{2}$$

Each class $[u]_A = \{v \in U : (u, v) \in IND(A)\}$ is interpreted as an elementary granule, and unions of elementary granules are granules of knowledge. Thus granulation, in this case, means forming aggregates of objects which are indiscernible over sets of attributes.

Rough inclusions, due to [11,12], are relations which in natural language can be expressed by saying that '*an object x is a part of an object y to a degree of r*'. A formal description of a rough inclusion is as a relation,

$$\mu \subseteq U \times u \times [0, 1] \tag{3}$$

In [12], [13], some methods for inducing rough inclusions in information/decision systems were introduced, from which we apply in this paper methods based on voting by test objects by means of weights computed with the help of residual rough inclusions, which we will now discuss.

Granular computing, introduced by Zadeh [18], consists in replacing objects with 'clumps of objects' collected together by means of a similarity relation, and in computing using these aggregates. In our setting, granules are formed by means of rough inclusions in the way pointed to in [12], see a survey in [13]. In formal terms, for a rough inclusion μ, an object u, and a real number $r \in [0, 1]$, a *granule about u of radius r*, $g(u, r)$ is defined as follows,

$$g(u, r) = \{v \in U : \mu(v, u, r)\} \tag{4}$$

1.3 Granular Classification of Knowledge

This type of granules may be applied in synthesis of a classifier in the way first proposed in [12]. The idea consists of fixing a radius of granulation r, and computing granules $g(u, r)$ for all objects $u \in U$. From the set of all granules a covering $C(U, r)$ is chosen, usually by means of a random choice. For each granule $g \in C(U, r)$, factored values $\overline{a}(g)$ of attributes a on g are computed, usually by means of majority voting, with random resolution of ties. The decision system $(U(C, r), \{\overline{a} : a \in A\}, \overline{d})$ is called a *granular resolution* of the initial decision system (U, A, d). For the granular resolution, various methods known in rough sets or elsewhere for classifier synthesis can be applied. The main features of this approach, see [2,3], [14], are: noise reduction - resulting in higher accuracy of classification - and classifier size reduction, resulting in much smaller number of classifying rules.

In the next section we describe in more detail rough inclusions used in this work along with their usage in analysis of DNA microarrays.

2 Application of Residual Rough Inclusions

For the decision system (U, A, d), we outline a rough inclusion based on the notion of a residuum of a t–norm.

2.1 Residua of T–Norms, Residual Rough Inclusions

The function $T : [0,1] \times [0,1] \rightarrow [0,1]$ which is symmetric, associative, increasing in each coordinate, and subject to boundary conditions: $T(x,0) = 0, T(x,1) = x$, is a t–norm, see, e.g, [8]. Examples of t–norms are,

1. (the Łukasiewicz t–norm) $L(x,y) = max\{0, x+y-1\}$.
2. (the Product t–norm) $P(x,y) = x \cdot y$.
3. (the Minimum t–norm) $M(x,y) = min\{x,y\}$.

By a *residuum* $x \Rightarrow_T y$ of a t–norm T, a function is meant, defined by means of,

$$x \Rightarrow_T y \geq r \ if \ and \ only \ if \ T(x,r) \leq y \tag{5}$$

As all t–norms L, P, M are continuous, in their cases, the residual implication is given by the formula,

$$x \Rightarrow_T y = max\{r : T(x,r) \leq y\} \tag{6}$$

Residual rough inclusions on the interval $[0,1]$ are defined, see, eg, [13] as,

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

2.2 A Voting Scheme for Decision Value Assignment

In classifier synthesis, this rough inclusion, e.g., induced by the Łukasiewicz t–norm L, is applied in the following way. As usual, the data set is split into *training set* and the *test set*. For a test object u, training objects v vote for decision value at u by means of weights

$$w(v,u,\varepsilon) = dis_\varepsilon(u,v) \Rightarrow_T ind_\varepsilon(u,v) \tag{8}$$

For each decision value v_d, a parameter,

$$Param(v_d) = \sum_{\{v \in U_{trn} : d(v) = v_d\}} w(v,u,\varepsilon) \tag{9}$$

is computed and the decision value assigned to u is $v_d(u)$ with the property that

$$v_d(u) = min\{Param(v_d) : v_d\} \tag{10}$$

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

3 DNA Microarray Features Extraction Method

The main purpose of this work is to present a selected gene extraction method based on modified F statistics - by using a classifier based on mereological granules. The data presented in this paper can be interpreted without context, because all the decision classes of the examined DNA microarrays are classified in a general sense, as one big decision system.

The huge amount of information obtained from DNA microarrays, due to the large number of gene–attributes, needs some preparatory methods in order to reduce this amount of information. We attempt to choose the genes that best separate the decision classes. Our approach to the separation of classes in this paper is as follows.

We have applied here F statistics, extended over multiple decision classes, which are well-known for the separation of the two decision classes.

Features Extraction Method Based on Modified F Statistics Method: Case6 (MSF6). 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 pairs of decision classes c_i, c_j, where $i, j = 1, 2, ..., k$ and $i \neq j$ in the following way. We let,

$$F_{c_i,c_j}(a) = \frac{MSTR_{c_i,c_j}(a)}{MSE_{c_i,c_j}(a)} \tag{11}$$

$$C_i^a = \{a(u) : u \in U \text{ and } d(u) = c_i\}, C_j^a = \{a(v) : v \in U \text{ and } d(v) = c_j\}.$$

$$\overline{C}_i^a = \frac{\{\sum a(u) : u \in U \text{ and } d(u) = c_i\}}{card\{C_i^a\}}, \overline{C}_j^a = \frac{\{\sum a(v) : v \in U \text{ and } d(v) = c_j\}}{card\{C_j^a\}}$$

$$\overline{C}_{i,j}^a = \frac{\{\sum a(u) : u \in U \text{ and } (d(u) = c_i \text{ or } d(u) = c_j)\}}{card\{C_i^a\} + card\{C_j^a\}},$$

$$MSTR_{c_i,c_j}(a) = card\{C_i^a\} * (\bar{C}_i^a - \bar{C}_{i,j}^a)^2 + card\{C_j^a\} * (\bar{C}_j^a - \bar{C}_{i,j}^a)^2$$

$$MSE_{c_i,c_j}(a) = \frac{\sum_{l=1}^{card\{C_i^a\}}(a(u_l) - \bar{C}_i^a) + \sum_{m=1}^{card\{C_j^a\}}(a(v_m) - \bar{C}_j^a)}{card\{C_i^a\} + card\{C_j^a\} - 2}$$

$$\text{where } u_l \in C_i^a,\ l = 1, 2, ..., card\{C_i^a\},\ v_m \in C_j^a, m = 1, 2, ..., card\{C_j^a\}$$

After the rate of the separation $F_{c_i,c_j}(a)$, are computed for all genes $a \in A$ and all pairs of decision classes c_i, c_j, where $i \neq j$ and $i < j$ genes are sorted in decreasing order of , $F_{c_i,c_j}(a)$

$F^1_{c_{i_1},c_{i_2}} > F^2_{c_{i_1},c_{i_2}} > ... > F^{card\{A\}}_{c_{i_1},c_{i_2}}$, where $i_1 \in \{1,2,...,k-1\}$ and $i_2 \in \{i_1+1,...,k\}$

Finally, for experiments we have chosen 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 $j_1 = 1,2,...,k-1$ **do**
    for $j_2 = j_1+1,...,k$ **do**
      if $F_{c_{j_1},c_{j_2}}(a) = F^i_{c_{j_1},c_{j_2}}(a)$ and $a \notin A'$ **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
  if $iter = fixed\ number\ of\ the\ best\ genes$ **then**
    BREAK
  end if
end for
return A'

4 Augmented Weighted Voting by Granules of Training Objects

The voting scheme proposed in sect. 2.2 is here augmented along the lines of [2]. The idea is to increase or decrease weights depending on the case, as shown in five variants (as Algorithms 8_v1.1, v1.2, v1.3, v1.4, v1.5 of [2]). These variants are described in [1], [14], but in this work we use only the best algorithm among those studied, variant 8_v1.4.

The procedure of chosen algorithm is as follows:

Step 1. The training decision system (U_{trn}, A, d) and the test decision system (U_{tst}, A, d) have been input, where U_{tst}, U_{trn} are, respectively, the test set and the training set, A is a set of attributes, and d is a decision.

Step 2. max_attr_a and min_attr_a have been found from the training data set, where max_attr_a, min_attr_a are, respectively, the maximal and the minimal value of attribute a on the training set.

Step 3. A chosen value of ε (determining attribute similarity degree) has been input.

Step 4. Classification of testing objects by means of weighted granules of training objects is done as follows:

For all conditional attributes $a \in A$, training objects $v_p \in U_{trn}$, where $p \in \{1, ..., card\{U_{trn}\}$ and test objects $u_q \in U_{tst}$, where $q \in \{1, ..., card\{U_{tst}\}$, for $train_a = max\_attr_a - min\_attr_a$ and $||a(u_q) - a(v_p)|| = \frac{|a(u_q)-a(v_p)|}{train_a}$ we compute

Subcase a) *If* $||a(u_q) - a(v_p)|| \geq \varepsilon$, *then*

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

i. e.,

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

Subcase b) *If* $||a(u_q) - a(v_p)|| < \varepsilon$, *then*

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

After weights in either Case are computed - for a given test object u_q and each training objects v_p - the voting procedure comprises computing values of parameters,

$$Param(c) = \sum_{\{v_p \in U_{trn} : d(v_p) = c\}} w(u_q, v_p), \tag{12}$$

for $\forall c$, decision classes.

Finally, the test object u_q is classified to the class c^* with a minimal value of Param(c).

After all test objects u_q are classified, quality parameters Total accuracy and Total coverage are computed.

The results for our algorithms with real DNA microarrays (see Table 1 from Advanced Track of Discovery Challenge see [16] and [17]) are reported in the next section.

5 The Results of Experiments with Leave One Out Method for Sample DNA Microarray Data

As we have studied, DNA microarrays contain unequal and small significant decision classes - see Table 1 - which is why we are evaluating results by a balanced accuracy parameter,

$$B.acc = \frac{acc_{c_1} + acc_{c_2} + ... + acc_{c_k}}{k} \tag{13}$$

Due to considerations of space, only an exemplary test can be discussed here. We apply our best classification algorithm 8_v1.4 among those studied [1] based on weighed voting with fixed parameter $\varepsilon = 0.01$, and our feature extraction method with Leave One Out method (LOO). For Leave One Out method a confusion matrix is built, in which the tested objects from all folds are treated as one test decision system. The motivation to use the Leave One Out method can be found, among other places in [9] and [17]. These papers prove the effectiveness and almost unbiased character of this method. Another argument proving its effectiveness it that FS+LOO model was successfully used for microarray data by the winners of the Advanced Track competition [17].

Table 1. An information table of the examined data sets (see [16]); data1 = anthracyclineTaxaneChemotherapy, data2 = BurkittLymphoma, data3 = HepatitisC, data4 = mouseType, data5 = ovarianTumour, data6 = variousCancers_final

Data.name	*No.of.attr*	*No.of.obj*	*No.of.dec.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%)

5.1 The Results for Our Gene Extraction Method

DNA microarray gene separation method MSF6 based on modified statistic F produces one of the best average results in a global sense from among all the methods that we have studied. On the basis of average results for our best method - see Table 2 - we can conclude that the best balanced accuracy 0.789 for all examined data has been obtained with only 50 genes. Table 4 presents the comparison of our best results and the results of the winners of Advanced Track discovery challenge - see [17]. It is evident that our methods are comparable to, or even better than, other methods. Balanced accuracy computed in all 28 decision classes of examined data is about 3 percent better than the best from Advanced Track [17].

Table 2. Leave One Out; The average balanced accuracy of classification for MSF6 algorithm; Examined data sets: all from Table 1; No.of.genes = number of classified genes, method = method's name

*method**No.of.genes*	10	20	50	100	200	500	1000
MSF6	**0.718**	**0.759**	**0.789**	**0.782**	**0.781**	**0.777**	**0.783**

Table 3. Leave One Out; 50 genes; The balanced accuracy of classification for all 28 decision classes with MSF6 algorithm; Examined data sets: all from Table 1, acc_b = Balanced Accuracy

data.class	acc_b	*data.class*	acc_b	*data.class*	acc_b	*data.class*	acc_b
*data*1.1	0.568	*data*3.1	0.927	*data*4.4	0.81	*data*6.4	0.538
*data*1.2	0.703	*data*3.3	0.87	*data*4.1	1	*data*6.7	1
*data*2.3	0.969	*data*4.3	0.952	*data*5.3	0.963	*data*6.5	0.809
*data*2.2	0.977	*data*4.2	0.536	*data*5.1	1	*data*6.6	0.714
*data*2.1	0.688	*data*4.7	0.438	*data*5.2	0.4	*data*6.8	0.938
*data*3.2	0.941	*data*4.6	0.359	*data*6.3	0.958	*data*6.9	0.833
*data*3.4	1	*data*4.5	0.629	*data*6.2	0.665	*data*6.10	0.909

Table 4. Average balanced accuracy; Modified F statistics vs Advanced Track results of the Discovery Challenge [17]; Examined data sets: all from Table 1; in case * Leave One Out result for 50 genes

method	*Balanced Accuracy*
*MSF6**	**0.789**
RoughBoy [17]	0.75661
ChenZe [17]	0.75180
wulala [17]	0.75168

6 Conclusions

The research results for our 8_v1_w4 classification method [2], [14] (with gene extraction MSF6 algorithm with examined data) are comparable to the best results from the Advanced Track of data mining contest see [17]. Those results have been evaluated by means of average balanced accuracy computed in all 28 decision classes of examined data. What follows from our experiments is that the essential element of gene separation methods is the way to choose the best genes after their calculation. In the case of the MSF6 method we choose genes which best separate particular pairs of decision classes one by one from all combinations, without the repetition of length 2 of decision classes.

The search is in progress for a theoretical explanation of the effectiveness of gene separation methods, based on F statistics, as well as work aimed at developing the theoretical description of these statistics, and will be reported.

Acknowledgements

The author wishes to express his thanks to Professor Lech Polkowski for his invaluable support and advice.

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

References

1. 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 2010. IEEE Press, Sergy Pontoise France (2010)
2. 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)
3. Artiemjew, P.: On Classification of Data by Means of Rough Mereological Granules of Objects and Rules. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) RSKT 2008. LNCS (LNAI), vol. 5009, pp. 221–228. Springer, Heidelberg (2008)
4. Brown, M., Grundy, W., et al.: Knowledge-based analysis of microarray gene expression data by using support vector machines. University of California, Berkeley (1999)
5. Eisen, M.B., Brown, P.O.: DNA arrays for analysis of gene expression. Methods Enzymol 303, 179–205 (1999)
6. Furey, T.S., Cristianini, D.N., Bernarski, S.M., Haussler, D.: Support Vector Machine Classification and Validation of Cancer Tissue Samples Using Microarray Expression Data. Bioinformatics 16, 906–914 (2000)
7. Gorecki, P., Artiemjew, P.: DNA microarray classification by means of weighted voting based on rough set classifier. In: Proceedings 2010 IEEE International Conference on Soft Computing and Pattern Recognition SocPar 2010, pp. 269–272. IEEE Computer Society, Sergy Pontoise (2010)
8. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (1998)
9. Molinaro, A.M., Simon, R., Pfeiffer, R.M.: Prediction error estimation: a comparison of resampling methods. Bioinformatics 21(15), 3301–3307 (2005)
10. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht (1991)
11. Polkowski, L.: Toward rough set foundations. Mereological approach (a plenary lecture). In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 8–25. Springer, Heidelberg (2004)
12. Polkowski, L.: Formal granular calculi based on rough inclusions (a feature talk). In: Proceedings 2005 IEEE Int. Confrence on Granular Computing GrC 2005, pp. 57–62. IEEE Press, Los Alamitos (2005)
13. Polkowski, L.: A Unified Approach to Granulation of Knowledge and Granular Computing Based on Rough Mereology: A Survey. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Handbook of Granular Computing, pp. 375–401. John Wiley & Sons, New York (2008)
14. Polkowski, L., Artiemjew, P.: On classifying mappings induced by granular structures. In: Peters, J.F., Skowron, A., Rybiński, H. (eds.) Transactions on Rough Sets IX. LNCS, vol. 5390, pp. 264–286. Springer, Heidelberg (2008)
15. Schena, M.: Microarray analysis. Wiley, Hoboken (2003)
16. `http://tunedit.org/repo/RSCTC/2010/A`
17. Wojnarski, M., Janusz, A., Nguyen, H.S., Bazan, J., Luo, C., Chen, Z., Hu, F., Wang, G., Guan, L., Luo, H., Gao, J., Shen, Y., Nikulin, V., Huang, T.-H., McLachlan, G.J., Bošnjak, M., Gamberger, D.: RSCTC'2010 Discovery Challenge: Mining DNA Microarray Data for Medical Diagnosis and Treatment. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 4–19. Springer, Heidelberg (2010)
18. Zadeh, L.A.: Fuzzy sets and information granularity. In: Gupta, M., Ragade, R., Yager, R.R. (eds.) Advances in Fuzzy Set Theory and Applications, pp. 3–18. North Holland, Amsterdam (1979)