THE 18TH EUROPEAN CONFERENCE ON MACHINE LEARNING
AND
THE 11TH EUROPEAN CONFERENCE ON PRINCIPLES AND PRACTICE
OF KNOWLEDGE DISCOVERY IN DATABASES

PROCEEDINGS OF THE INTERNATIONAL WORKSHOP ON

ROUGH SETS IN KNOWLEDGE DISCOVERY: FOUNDATIONS AND APPLICATIONS

RSKD'07

September 21, 2007

Warsaw, Poland

Editors:
Piotr Synak
Polish-Japanese Institute of Information Technology
Jakub Wróblewski
Polish-Japanese Institute of Information Technology

Workshop Organization

RSKD Workshop Chairs

Piotr Synak (Polish-Japanese Institute of Information Technology)
Jakub Wróblewski (Polish-Japanese Institute of Information Technology)

ECML/PKDD Workshop Chair

Marzena Kryszkiewicz (Warsaw University of Technology)

Workshop Program Committee

Aijun An
Jan Bazan
Shoji Hirano
Jan Komorowski
Lech Polkowski
Andrzej Skowron
Guoyin Wang
Hui Wang

Table of Contents

Classifiers from Granulated Data Sets: Concept Dependent and Layered Granulation

Piotr Artiemjew

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

Abstract. Granulation of data and the idea of a granular data set were proposed by L.A. Zadeh on the basis of the assumption that is at heart of all data mining techniques, i.e., that given a plausible similarity measure on objects in a data set, objects which are similar in a satisfactory degree would have also similar or even equal decision (class) values. This assumption underlies reasoning by analogy, nearest neighbors methodology, case based reasoning and rough set methods as well.
This assumption taken to an extreme implies that once a granular data system has been induced from a real data set, and decision/classification rules have been computed from it, these rules when applied to the original data should produce classification results close satisfactorily to classification results obtained on the real data with rules induced from the original non–granulated data. A number of tests performed borne out this hypothesis.
We present in this work results of experiments with real data sets and we work with the two extensions of granulation techniques presented by us in the literature so far, i.e., concept dependent granulation and layered granulation.
Keywords Rough sets, granulation of knowledge, rough inclusions, granular data sets and classifiers

1 Basic notions of rough set theory and granular theory

We choose information systems as the framework for defining knowledge understood as the ability to classify objects into various classes on the basis of indiscernibility relations, see [6].

1.1 Rough set notions

An information system [6] is a pair $IS = (U, A)$, where U is a set of objects (the universe of IS) and A is a set of attributes; we assume that both sets U, A are finite and nonempty. Attributes are represented as mappings on objects, i.e., each attribute $a \in A$ is a mapping $a : U \to V_a$, where the set V_a is the set of values of the attribute a. Adding an attribute $d \notin A$, one produces a *decision*

system $DS = (U, A, d)$ with the *decision* d reflecting a partition of objects into classes.

Objects $u, v \in U$ are *B–indiscernible* whenever $a(u) = a(v)$ for every $a \in B$, for each attribute set B. Formally, this fact is rendered by the *indiscernibility relation* $IND(B)$

$$(u, v) \in IND(B) \text{ iff } a(u) = a(v) \text{ for each } a \in B.$$

The relation $IND(B)$ does partition the universe U into classes,

$$[u]_B = \{v \in U : (u, v) \in IND(B)\},$$

where $u \in U$. Those classes form the *B-primitive granule* collection over the information system IS.
Unions of primitive granules are called *elementary granules.*

The language for object description in information systems is the descriptor language see [6]; its primitive formulas are *descriptors* of the form (a, v) with $a \in A, v \in V_a$. Formulas are constructed from descriptors by means of sentential connectives $\vee, \wedge, \neg, \Rightarrow$.

Semantics of formulas is defined as follows, where $[[p]]$ denotes the meaning of a formula p:

$$\begin{array}{c} 1.\ [[(a, v)]] = \{u \in U : a(u) = v\} \\ 2.\ [[p \vee q]] = [[p]] \cup [[q]] \\ 3.\ [[p \wedge q]] = [[p]] \cap [[q]] \\ 4.\ [[\neg p]] = U \setminus [[p]]. \end{array} \tag{1}$$

Each object $u \in U$ is described with respect to a set B of attributes by means of its information vector $inf_B(u) = \{(a = a(u)) : a \in B\}$.

1.2 Granulation and granular systems

Granulation of knowledge as a paradigm for approximate computing was proposed in Zadeh [18] and was adopted in a natural way by rough set community, see, e.g., [5], [11], [12], [15]. Granules were initially produced as unions of indiscernibility classes, and subsequently extensions in various directions were proposed. An important direction here have been granulation based on similarity relations. One method proposed in this area was usage of templates [4], i.e., generalized descriptors of the form $(a \in W_a)$ with semantics analogous to that of descriptors. Some classification results obtained by means of templates in [4] are shown here in Table 1.2 for Australian credit data set see [16].

Reasoning based on metric–induced similarity is k–nearest neighbors methods, see, e.g., [3], [17]. In this method, classification of a test object v is effected on the basis of majority voting among k nearest in the chosen metric to v training objects $t_1, ..., t_k$; one can call the set $\{t_1, .., t_n\}$ an *nn–granule(k).*

Another direction was usage of specialized similarity measures called *rough inclusions* formally derived within the paradigm of rough mereology, see [11], [12], [7], [8]. We will make use here of only one rough inclusion formally derived from

Table 1. Accuracy of classification by template and similarity methods [4]

paradigm	*system/method*	*Austr.credit*
Rough Sets	*Simple.templ./Hamming*	0.8217
Rough Sets	*Gen.templ./Hamming*	0.855
Rough Sets	*Simple.templ./Euclidean*	0.8753
Rough Sets	*Gen.templ./Euclidean*	0.8753
Rough Sets	*Match.tolerance*	0.8747
Rough Sets	*Clos.tolerance*	0.8246

the Łukasiewicz t–norm but otherwise well–known to data mining community: for an information system (U, A), the rough inclusion $\mu_L(u, v, r)$ is defined by means of the formula,

$$\mu_L(u, v, r) \Leftrightarrow \frac{|IND(u, v)|}{|A|} \geq r, \tag{2}$$

where $IND(u, v) = \{a \in A : a(u) = a(v)\}$; in plain words, $\mu(u, v, r)$ if and only if at least $r \cdot 100$ percent of attributes agree on u and v. Similarity defined by the predicate $\mu(u, v, r)$ is understood from the formula (2): in case $\mu(u, v, 1)$ holds, objects u and v are most similar, in fact, indiscernible; when the factor r tends to 0, u and v become less and less similar. Thus, $\mu(u, v, r)$ can be read as: u is similar to v to a degree r.

The idea of forming a granular counterpart to a given information system was proposed in [7] as follows. For an information system (U, A), and a granulation scheme $\mathcal{G}$, that yields a granule collection $Gr = \mathcal{G}(U)$, a covering of the universe U, $Cov(U)$, is chosen by a selected strategy $\mathcal{C}$; adopting a strategy $\mathcal{S}$, for each granule $g \in Cov(U)$, and each attribute $a \in A$, a value $\overline{a}(g) = \mathcal{S}\{a(u) : u \in g\}$ is determined. The new information system $(Cov(U), \overline{A})$, where $\overline{A} = \{\overline{a} : a \in A\}$, is a granular approximation to the original information system (U, A). Clearly, the same concerns any decision system (U, A, d) with the reduced decision $\overline{d}$.

As a strategy $\mathcal{S}$ the majority voting, see [3], was adopted, and as a strategy $\mathcal{C}$ for covering generation, the random selection was applied. The strategy $\mathcal{G}$ of granule formation was selected as in [7], i.e., a granule $g_r(u)$ of the radius r about an object u was formed as a set of objects v such that $\mu(v, u, r)$. In [9], [10], the hypothesis formulated in [7] that granulated data sets should preserve to a satisfactory degree the classification ability of the original data was borne out with many real data sets.

In this work, we report our results obtained with real data sets in case of extended granulation methods: concept–dependent granulation and layered granulation as well as with the hybrid approach consisting in a combination of those two methods.

2 Classification

Data are organized into decision systems. Classification of objects into decision classes in rough set paradigm consists in finding decision rules of the form $inf_B(u) \Rightarrow (d = v)$. The decision rule is certain when its meaning is U; otherwise the rule is possible.

A number of algorithms for rule induction were proposed by researchers in rough set theory; they are divided into the class of minimal algorithms aimed at description of classes with a minimal number of descriptors and exhaustive algorithms aimed at finding all possible rules.

Minimum size algorithms include LEM2 algorithm due to Grzymala–Busse [19] and covering algorithm in RSES package [14]; exhaustive algorithms include, e.g., LERS system due to Grzymala–Busse [21], systems based on discernibility matrices and Boolean reasoning by Skowron [13] implemented in the RSES package [14].

The rough set based rule induction systems as compared to other methods give on Australian credit data set the following results shown in Table 2. The results are from [1], see also [2].

Table 2. A comparison of errors in classification by rough set and other paradigms on Australian credit data set [1]

paradigm	*system/method*	*Austr.credit*
Stat.Methods	*Logdisc*	0.141
Stat.Methods	*SMART*	0.158
Neural Nets	*Backpropagation2*	0.154
Neural Networks	*RBF*	0.145
Decision Trees	*CART*	0.145
Decision Trees	*C4.5*	0.155
Decision Trees	*ITrule*	0.137
Decision Rules	*CN2*	0.204
Rough Sets	*NNANR*	0.140
Rough Sets	*DNANR*	0.165
Rough Sets	*best result*	0.130($SNAPM$)

3 Results for granular systems

Our experiments, see [9], [10], with many real data sets, proved that granular systems preserve knowledge to a high degree. We include in Table 3 results for Australian credit data set obtained from granular data set in comparison with other best results obtained by various rough set based methods.

Table 3. Best results for Australian credit by rough set based algorithms; in case $*$, reduction in object size is 49.9 percent, reduction in rule number is 54.6 percent; in case $**$, resp., 19.7, 18.2; in case $***$, resp., 3.6, 1.9

source	*method*	*accuracy*	*coverage*
([1])	$SNAPM(0.9)$	$error = 0.130$	–
([4])	*simple.templates*	0.929	0.623
([4])	*general.templates*	0.886	0.905
([4])	*closest.simple.templates*	0.821	1.0
([4])	*closest.gen.templates*	0.855	1.0
([4])	*tolerance.simple.templ.*	0.842	1.0
([4])	*tolerance.gen.templ.*	0.875	1.0
([22])	*adaptive.classifier*	0.863	–
([9])	$granular^{*}.r = 0.642$	0.8990	1.0
([9])	$granular^{**}.r = 0.714$	0.964	1.0
([10])	$granular^{***}.concept.r = 0.785$	0.9970	0.9995

The procedure for obtaining our results in granular case was as follows:

1. the data table (U, A) has been input;

2. classification rules have been found by means of RSES exhaustive algorithm;
3. classification of data set objects in U has been found by means of classification rules found at point 2;
4. given the granule radius, granules of that radius have been found;
5. a granular covering of the universe U has been chosen by a random choice;
6. the corresponding granular decision system has been determined;
7. a granular classifier has been induced from the granular system in point 6 by means of the algorithm of point 2;
8. classification of objects in U has been found by means of the classifier of point 7;
9. classifications from points 3,8 have been compared with respect to adopted global measures of quality: total accuracy and total covering.

Results quoted in Table 3 show that rules extracted from granulated data by RSES exhaustive algorithm perform on original Australian credit data even better than other methods on the original data.

Result in the last row of Table 3 – the best among rough set based results – was obtained by *concept–dependent granulation*. We are going to present some details of this approach.

4 Concept–dependent granulation

A modification of the approach presented for results shown above is the *concept dependent* granulation; a *concept* in the narrow sense is a decision/classification class. Granulation in this sense consists in computing granules for objects in the universe U and for all distinct granulation radii as previously, with the only restriction that given any object $u \in U$ and $r \in [0, 1]$, the new concept dependent granule $g^{cd}(u, r)$ is computed with taking into account only objects $v \in U$ with $d(v) = d(u)$, i.e., $g^{cd}(u, r) = g(u, r) \cap \{v \in U : d(v) = d(u)\}$. This method increases the number of granules in coverings but it is also expected to increase quality of classification, as expressed by accuracy and coverage.

Table 4 shows results (ten-fold test) for Australian credit data set obtained by the standard granular approach as compared to the concept dependent granular system. Granular coverings were selected at random, the strategy for determining attribute values on granules was the majority voting.

Table 4. Standard and concept dependent granular systems for Australian credit data set; exhaustive RSES algorithm:r=granule radius, macc=mean accuracy, mcov=mean coverage, mrules=mean number of rules, mtrn=mean training sample size; in each column first value is for standard, second for concept dependent

r	$macc$	$mcov$	$mrules$	$mtrn$
nil	1.0; 1.0	1.0; 1.0	12025; 12025	690; 690
0.0	0.0; 0.8068	0.0; 1.0	0; 8	1; 2
0.0714286	0.0; 0.7959	0.0; 1.0	0; 8.2	1.2; 2.4
0.142857	0.0; 0.8067	0.0; 1.0	0; 8.9	2.4; 3.6
0.214286	0.1409; 0.8151	0.2; 1.0	1.3; 11.4	2.6; 5.8
0.285714	0.7049; 0.8353	0.9; 1.0	8.1; 14.8	5.2; 9.6
0.357143	0.7872; 0.8297	1.0; 0.9848	22.6; 32.9	10.1; 17
0.428571	0.8099; 0.8512	1.0; 0.9986	79.6; 134	22.9; 35.4
0.5	0.8319; 0.8466	1.0; 0.9984	407.6; 598.7	59.7; 77.1
0.571429	0.8607; 0.8865	0.9999; 0.9997	1541.6; 2024.4	149.8; 175.5
0.642857	0.8988; 0.9466	1.0; 0.9998	5462.5; 6255.2	345.7; 374.9
0.714286	0.9641; 0.9880	1.0; 0.9988	9956.4; 10344.0	554.1; 572.5
0.785714	0.9900; 0.9970	1.0; 0.9995	11755.5; 11802.7	662.7; 665.7
0.857143	0.9940; 0.9970	1.0; 0.9985	11992.7; 11990.2	682; 683
0.928571	0.9970; 1.0	1.0; 0.9993	12023.5; 12002.4	684; 685
1.0	1.0; 1.0	1.0; 1.0	12025.0; 12025.0	690; 690

From Table 4 we infer that concept–dependent granulation enhances the classification results which for Australian credit are the best among obtained

by various rough set approaches and also better than those obtained by other paradigms shown in Table 1.

5 Layered granulation

A stability of granular systems can be checked by performing the granulation procedure in an iterative manner until no further change is noticed. We call this approach *a layered granulation*. For each radius r, the granular system obtained in the i–th iteration is granulated again at that radius to yield the $(i+1)$–st iteration and so on until no new iterate is obtained. At each iteration, classification rules were induced from the granulated data by RSES exhaustive algorithm and applied to test data. Results are presented in Tables 5 and 6.

Table 5. Normal layered–granulation; Australian; exhaustive algorithm. part1; r_gran=radius, tst=testing sample size, trn=training sample size, rul_no.=rule number, acc=accuracy, cov=coverage

normal granulation															
granulation layer volume		1				2				3					
r_gran	*tst*	*trn*	*no._rul*	*acc*	*cov*	*trn*	*no._rul*	*acc*	*cov*	*trn*	*no._rul*	*acc*	*cov*		
nill	690	690													
0.000000	690	1	0	0.000	0.000										
0.071428	690	1	0	0.000	0.000										
0.142857	690	2	0	0.000	0.000	1	0	0.000	0.000						
0.214286	690	2	2	0.733	1.000	1	0	0.000	0.000						
0.285714	690	4	13	0.764	1.000	1	0	0.000	0.000						
0.357143	690	8	21	0.793	1.000	1	0	0.000	0.000						
0.428571	690	26	92	0.835	1.000	2	0	0.000	0.000	1	0	0.000	0.000		
0.500000	690	60	486	0.852	1.000	10	88	0.789	0.948	3	32	0.436	1.000		
0.571429	690	154	1645	0.875	1.000	54	538	0.835	1.000	32	342	0.783	1.000		
0.642857	690	335	5295	0.901	1.000	253	3967	0.893	1.000	226	3452	0.894	1.000		
0.714286	690	557	9965	0.965	1.000	516	9501	0.959	1.000	499	9400	0.957	1.000		
0.785714	690	664	11766	0.990	1.000	659	11733	0.990	1.000						
0.857143	690	682	12007	0.994	1.000										
0.928571	690	684	12001	0.997	1.000										
1.000000	690	690	12025	1.000	1.000										

Table 6. Normal layered–granulation; Australian; exhaustive algorithm. part2; r_gran=radius, tst=testing sample size, trn=training sample size, rul_no.=rule number, acc=accuracy, cov=coverage

normal granulation													
granulation layer volume		4				5				*total*			
r_gran	*tst*	*trn*	*no._rul*	*acc*	*cov*	*trn*	*no._rul*	*acc*	*cov*	*trn*	*no._rul*	*acc*	*cov*
nill	690	690								690	12025	1.000	1.000
0.000000	690									1	0	0.000	0.000
0.071428	690									1	0	0.000	0.000
0.142857	690									1	0	0.000	0.000
0.214286	690									1	0	0.000	0.000
0.285714	690									1	0	0.000	0.000
0.357143	690									1	0	0.000	0.000
0.428571	690									1	0	0.000	0.000
0.500000	690									3	32	0.436	1.000
0.571429	690	25	316	0.764	1.000	24	316	0.771	1.000	24	316	0.771	1.000
0.642857	690	214	3275	0.890	1.000					214	3275	0.890	1.000
0.714286	690	492	9340	0.957	1.000	490	9328	0.954	1.000	490	9328	0.954	1.000
0.785714	690									659	11733	0.990	1.000
0.857143	690									682	12007	0.994	1.000
0.928571	690									684	12001	0.997	1.000
1.000000	690									690	12025	1.000	1.000

Stable granules are obtained in the 5–th iteration and final results are shown in column *total*. The stability of granular classifiers is preserved at high degree;

for instance, at $r = .714$, the final number of granules is 490, i.e., 71 percent of the original training objects number and drop in accuracy is only 4.6 percent to .954 from 1.0 in non–granular case.

6 A hybrid approach

In this approach two preceding methods: concept–dependent and layered are combined, i.e., concept–dependent granulation is performed iteratively in the layered manner. Results are reported in Tables 7 and 8.

Table 7. Concept–dependent layered–granulation; Australian; exhaustive algorithm. part1; r_gran=radius, tst=testing sample size, trn=training sample size, rul_no.=rule number, acc=accuracy,cov=coverage

concept − dependent granulation													
granulation layer volume		1				2				3			
r_gran	*tst*	*trn*	*no._rul*	*acc*	*cov*	*trn*	*no._rul*	*acc*	*cov*	*trn*	*no._rul*	*acc*	*cov*
nill	690	690											
0.000000	690	2	8	0.825	1.000								
0.071428	690	2	8	0.825	1.000								
0.142857	690	3	10	0.826	1.000	2	8	0.777	1.000				
0.214286	690	8	10	0.777	1.000	2	8	0.823	1.000				
0.285714	690	10	15	0.807	1.000	2	10	0.826	1.000				
0.357143	690	16	38	0.857	1.000	2	10	0.790	1.000				
0.428571	690	31	113	0.839	1.000	4	12	0.793	1.000	2	10	0.788	1.000
0.500000	690	79	546	0.827	0.999	15	110	0.810	0.970	5	90	0.587	1.000
0.571429	690	184	2117	0.896	0.999	77	789	0.835	1.000	47	558	0.720	1.000
0.642857	690	373	6216	0.949	1.000	286	4925	0.913	1.000	260	4416	0.906	1.000
0.714286	690	575	10295	0.987	0.999	534	9864	0.981	0.999	519	9739	0.981	0.999
0.785714	690	665	11812	0.997	0.997	664	11806	0.997	0.997				
0.857143	690	683	11977	0.977	1.000								
0.928571	690	685	11990	1.000	0.999								
1.000000	690	690	12025	1.000	1.000								

Table 8. Concept–dependent layered–granulation; Australian; exhaustive algorithm. part2; r_gran=radius, tst=testing sample size, trn=training sample size, rul_no.=rule number, acc=accuracy,cov=coverage

concept − dependent granulation													
granulation layer volume		4				5				*total*			
r_gran	*tst*	*trn*	*no._rul*	*acc*	*cov*	*trn*	*no._rul*	*acc*	*cov*	*trn*	*no._rul*	*acc*	*cov*
nill	690	690								690	12025	1.000	1.000
0.000000	690									2	8	0.825	1.000
0.071428	690									2	8	0.825	1.000
0.142857	690									2	8	0.777	1.000
0.214286	690									2	8	0.823	1.000
0.285714	690									2	10	0.826	1.000
0.357143	690									2	10	0.790	1.000
0.428571	690									2	10	0.788	1.000
0.500000	690									5	90	0.587	1.000
0.571429	690	42	515	0.648	1.000	40	501	0.613	1.000	40	501	0.613	1.000
0.642857	690	249	4243	0.877	1.000	247	4217	0.858	1.000	247	4217	0.858	1.000
0.714286	690	517	9723	0.983	0.999					517	9723	0.983	0.999
0.785714	690									664	11806	0.997	0.997
0.857143	690									683	11977	0.977	1.000
0.928571	690									685	11990	1.000	0.999
1.000000	690									690	12025	1.000	1.000

As expected, results in this case are yet better: at $r = .714$, drop in accuracy is 1.7 percent only to .983 from 1.0 in non–granular case.

7 Conclusion

The results presented in this work show the validity of granular approach and usefulness of concept–dependent, layered and hybrid approaches. The usage of well–known RSES algorithms is acknowledged with gratitude. Although we have at our disposal our own algorithms for inducing classifiers yet using a respected commonly known and applied tool has seemed important for objectivity of our results.

References

1. J. G. Bazan, A comparison of dynamic and non–dynamic rough set methods for extracting laws from decision tables, In: L. Polkowski, A. Skowron (Eds.), Rough Sets in Knowledge Discovery 1. Physica Verlag, Heidelberg, 1998, pp. 321–365.
2. J. G. Bazan, P. Synak, Nguyen S.H., Nguyen H. S., J. Wróblewski, Rough set algorithms in classification problems. In: Polkowski L, Tsumoto S, Lin T. Y. (eds) Rough Set Methods and Applications. New Developments in Knowledge Discovery in Information Systems, Physica Verlag, Heidelberg, 2000, pp. 49–88.
3. R. O. Duda, P. E. Hart, D.G. Stork, Pattern Classification, Wiley Interscience, New York, 2001.
4. Nguyen Sinh Hoa, Regularity analysis and its applications in Data Mining, In: L. Polkowski, S. Tsumoto, T. Y. Lin (Eds.), Rough Set Methods and Applications. New Developments in Knowledge Discovery in Information Systems, Physica Verlag, Heidelberg, 2000, pp. 289–378.
5. T. Y. Lin, Granular computing: Examples, intuitions and modeling, in: Proceedings 2005 IEEE Int. Conference on Granular Computing GrC05, IEEE Press, 2005, pp. 40–44.
6. Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer, Dordrecht, 1991.
7. L. Polkowski, Formal granular calculi based on rough inclusions (a feature talk), In: Proceedings 2005 IEEE Int. Conference on Granular Computing GrC05, IEEE Press, 2005, pp. 57–62.
8. L. Polkowski, A model of granular computing with applications: Granules from rough inclusions in information systems, In: Proceedings 2006 IEEE Int. Conference on Granular Computing, IEEE Press, 2006, pp. 9–16.
9. L. Polkowski, P. Artemjew, A study in granular rough computing: On factorizations of classifiers through granular structures, Fundamenta Informaticae (submitted).
10. L. Polkowski, P. Artiemjew, On granular rough computing: Factoring classifiers through granulated decision systems, in: Proceedings RSEiSP07, Warsaw, June 2007; Lecture Notes in Artificial Intelligence 4585, Springer, Berlin, 2007, pp. 271–279.
11. L. Polkowski, A. Skowron, Rough mereological calculi of granules: A rough set approach to computation, Computational Intelligence. An International Journal, 17 (2001) 472–492.
12. L.Polkowski, A.Skowron, Towards an adaptive calculus of granules, In: L.A.Zadeh, J.Kacprzyk (Eds.), Computing with Words in Information/Intelligent Systems 1., Physica Verlag, Heidelberg, 1999, pp. 201–228.

13. A. Skowron, Boolean reasoning for decision rules generation, In: J. Komorowski, Z. Ras (Eds.), Proceedings of ISMIS'93. Lecture Notes in Artificial Intelligence, Vol. 689 , Springer Verlag, Berlin, 1993, pp. 295–305.
14. A. Skowron, J.G. Bazan, P. Synak, J. Wróblewski, Nguyen Hung Son, Nguyen Sinh Hoa, A. Wojna, RSES: A system for data analysis; available at http://logic.mimuw.edu.pl/ rses
15. A. Skowron, J. Stepaniuk, Information granules: towards foundations of granular computing, International Journal for Intelligent Systems 16 (2001) 57–85.
16. UCI Repository: available at http://www.ics.uci.edu./ mlearn/databases/
17. A. Wojna, Analogy–based reasoning in classifier construction, Transactions on Rough Sets Vol. IV. Lecture Notes in Computer Science, Volume 3700, Springer Verlag, Berlin, 2005, pp. 277–374.
18. L. A. Zadeh, Fuzzy sets and information granularity, In: M. Gupta, R. Ragade, R. Yager(Eds.), Advances in Fuzzy Set Theory and Applications, North–Holland, Amsterdam, 1979, pp. 3–18.
19. J.W. Grzymala–Busse, Data with missing attribute values: Generalization of rule indiscernibility relation and rule induction, *Transactions on Rough Sets I*, subseries of *Lecture Notes in Computer Science*, Springer Verlag, Berlin, 2004, 78–95.
20. J.W.Grzymala–Busse, Ming Hu, A comparison of several approaches to missing attribute values in Data Mining, *Lecture Notes in Artificial Intelligence* 2005, Springer Verlag, Berlin, 2000, 378–385.
21. J.W.Grzymala–Busse, Ming Hu, A comparison of several approaches to missing attribute values in Data Mining, *Lecture Notes in Artificial Intelligence* 2005, Springer Verlag, Berlin, 2000, 378–385.
22. J. Wróblewski, Adaptive aspects of combining approximation spaces, In: S.K.Pal, L.Polkowski, A.Skowron, *Rough–neural Computing. Techniques for Computing for Words*, Springer Verlag, 2004, 139–156.

Indiscernibility-based Clustering of Non-Euclidean Relational Data

Shoji Hirano and Shusaku Tsumoto

Department of Medical Informatics
Shimane University, School of Medicine
89-1 Enya-cho, Izumo, Shimane 693-8501, Japan
E-mails: hirano@ieee.org, tsumoto@computer.org

Abstract. In this paper, we present a new method for clustering non-Euclidean relational data based on the combination of indiscernibility level and linkage algorithm. The indiscernibility level quantifies the level of global agreement for classifying two objects into the same category as indiscernible objects. Single-linkage grouping is then used to merge objects according to the indiscernibility level from bottom to top and construct the dendrogram. This scheme enables users to examine the hierarchy of data granularity and obtain the set of indiscernible objects that meets the given level of granularity. Additionally, since indiscernibility level is derived based on the binary classifications assigned independently to each object, it can be applied to non-Euclidean, asymmetric relational data. Through the clustering experiments on a synthetic dataset we demonstrate the property and usefulness of our method.

1 Introduction

Non-Euclidean relational data is a special type of data that have the following properties: First, objects are not represented in a usual feature vector space; instead, their relationships (usually similarity or dissimilarity) are measured and stored in a relational data matrix such as a proximity matrix. Second, the dissimilarity can be non-Euclidean; that means the dissimilarity may not satisfy the triangular inequality nor symmetry. This type of data mainly appears in the application areas such as social sciences where the relationships between objects are main concern; examples include subjectively judged relations between students, and input/output of the persons between countries [1]. Clustering would provide a way of discovering interesting groups of objects like groups of students with good friendship from such a non-Euclidean relational data. However, the above-mentioned property of the data would make it difficult to apply conventional clustering methods [2] [3] such as hierarchical clustering, k-means and EM algorithms. In order to find the best partition of objects that maximizes both inter-cluster homogeneity and between-clusters isolation, clustering methods usually employ geometric measures such as the variance of dissimilarities. However, it becomes difficult to form appropriate clusters if only a proximity matrix is available as intrinsic information for analysis and the raw attribute values

of data are unavailable or inaccessible. This is because the lack of attribute-value information may bring a difficulty in computing the global properties of groups such as centroids. Additionally, the choice of global coherence/isolation measures is limited if the dissimilarity is defined as a subjective or relative measure, because such a measure may not satisfy the triangular inequality for triplets of objects. Although conventional hierarchical clusterings are known to be able to deal with relative or subjective measures, they involve other problems such as erosion or expansion of data space by intermediate objects between large clusters and the results are dependent on the orders of object handling [2]. The NERF c-means proposed by Hathaway et al. [4] is an extension of fuzzy c-means and capable of handling the non-Euclidean relational data. However, as it is a sort of partitional clustering method, it is still difficult to examine the structure of the data, namely, the hierarchy of data groups.

In this paper we present a novel clustering method that represents the hierarchy of data granularity using a dendrogram. Instead of using (dis-)similarity of objects, we use indiscernibility of objects as proximity. The indiscernibility represents the level of global agreement for classifying a pair of objects as indiscernible objects, and is calculated based on the binary classifications determined independently to each object. Then the simple nearest neighbor hierarchical clustering is used to construct a dendrogram of objects, which represents the hierarchy of indiscernibility. This scheme allows us to control the granularity of resultant object groups, by interactively selecting the threshold level of indiscernibility. The benefits of this method also include that the dissimilarity of objects for forming the binary classifications does not need to satisfy symmetry nor triangular inequality; thus it could be applied to various kind of datasets including relational data.

The remainder of this paper is organized as follows. Section 2 provides some definitions related to the concept of indiscernibility used in this paper. Section 3 explains procedures of the proposed method in detail. Section 4 provides experimental results on a small synthetic data that demonstrate the behavior and usefulness of the proposed method. Finally, Section 5 conclude this paper.

2 Preliminaries

This section briefly introduces some fundamental definitions about indiscernibility [5] used in this paper. Let $U \neq \phi$ be a universe of discourse and X be a subset of U. An equivalence relation R classifies U into a set of subsets $U/R = \{X_1, X_2, ...X_N\}$ that satisfies the following conditions: (1) $X_i \subseteq U$, $X_i \neq \phi$ for any i, (2)$X_i \cap X_j = \phi$ for any $i, j, i \neq j$, (3) $\cup_{i=1,2,...N} X_i = U$. Any subset X_i is called a category and represents an equivalence class of R. A category in R containing an object $x \in U$ is denoted by $[x]_R$. Objects x_i and x_j in U are *indiscernible on* R if $(x_i, x_j) \in P$ where $P \in U/R$. For a family of equivalence relations $\mathbf{P} \subseteq \mathbf{R}$, an indiscernibility relation over $\mathbf{P}$ is defined as the intersection of individual relations $Q \in \mathbf{P}$.

3 Method

The proposed method consists of three steps:

1. Assign a binary classification to each object.
2. Compute the indiscernibility level for each pair of objects according to the binary classifications. Then construct a symmetric square matrix of indiscernibility level.
3. Construct a dendrogram from the indiscernibility matrix using the single linkage (nearest-neighbor) hierarchical clustering.

Before introducing the details of each step, we here describe the overall concept of our method. We aim at representing *hierarchy of object indiscernibility* using a dendrogram and enabling the users to interactively construct clusters of objects that meet the given level of granularity. The granularity of data in this paper puts its basis on the *indiscernibility level* of objects. The *indiscernibility level*, defined for a pair of objects, is a new measure of object similarity that represents the level of global agreement for regarding the two objects as indiscernible. Binary classifications play a key role in evaluating the indiscernibility of objects. According to the proximity to other objects, each of the N objects independently classifies U into two disjoint sets of objects; one containing objects that are indiscernible to that object, and another containing objects discernible to that object. Then, indiscernibility level of any two objects is assessed based on the ratio of binary classifications that commonly classify them as indiscernible.

The indiscernibility level can be associated with data granularity. Control of data granularity can be done by setting a threshold value on the indiscernibility level and regarding objects as indiscernible if their indiscernibility level is higher than the threshold. This process requires iterative hierarchical merging and can be nicely handled with nearest-neighbor agglomerative hierarchical grouping [2]. Note that mergence of objects is done according to the indiscernibility level, not to a conventional proximity measure such as (dis-)similarity; the global agreement for merging two objects has already been embedded within this measure.

3.1 Binary Classifications

Our method first let each object independently classify the entire set of objects U into two disjoint sets P and $U - P$. This binary classification is formalized using an equivalence relation. Let $U = \{x_1, x_2, ..., x_N\}$ be the set of objects we are interested in and let R_i be an equivalence relation defined for object x_i. Then binary classification for x_i is defined by

$$U/R_i = \{P_i,\ U - P_i\}, \tag{1}$$

where P_i contains objects that are indiscernible to x_i, and $U - P_i$ contains objects that are discernible to x_i. U/R_i can be alternatively written as $U/R_i = \{[x_i]_{R_i}, \overline{[x_i]_{R_i}}\}$, where $[x_i]_{R_i} \cap \overline{[x_i]_{R_i}} = \phi$ and $[x_i]_{R_i} \cup \overline{[x_i]_{R_i}} = U$ hold.

Table 1. An example of asymmetric, non-Euclidean dissimilarity matrix.

x_i \ x_j	x_1	x_2	x_3	x_4	x_5
x_1	0.0	0.1	0.1	0.7	0.9
x_2	0.2	0.0	0.1	0.6	0.8
x_3	0.7	0.1	0.0	0.2	0.8
x_4	0.2	0.3	0.2	0.0	0.6
x_5	0.7	0.6	0.9	0.1	0.0

Method for determining the binary classification is arbitrary. R_i should provide some criteria to form P_i; however, it may not necessarily be defined explicitly. For example, one may simply form P_i according to the proximity between objects as

$$P_i = \{x_j |\ d(x_i, x_j) \leq Th_{di}\}, \quad \forall x_j \in U. \tag{2}$$

where $d(x_i, x_j)$ denotes dissimilarity between objects x_i and x_j, and Th_{di} denotes a threshold value of dissimilarity for object x_i. Other methods, such as k-means with cluster number 2, can be used as alternatives if they are appropriate with respect to the property of the data. We have introduced a method for constructing binary grouping based on the denseness of the objects in [6]; however, one may use any method, including the choice of proximity measure, under the condition that it has the ability of performing binary classification on U.

An important point to note is that U/R_i can be defined locally and independently to each object x_i, $i = 1, 2, ..., N$. For example, U/R_1 can be defined according only to the relationships between x_1 and other $N-1$ objects, without taking into account other information such as relationships between x_2 and x_3. Similarly, U/R_i can be defined according only to the relationships between x_i and other $N-1$ objects, without taking into account the relationships between x_j and x_k, where $j, k \neq i$. This property enables us to employ an asymmetric, non-Euclidean proximity matrix as input data.

[Example 1]: Binary Classification

Let us consider an asymmetric, non-Euclidean dissimilarity matrix shown in Table 1. Here $U = \{x_1, x_2, x_3, x_4, x_5\}$. Suppose we define binary classifications U/R_i as

$$\begin{aligned} U/R_i &= \{P_i,\ U - P_i\}, \\ P_i &= \{x_j |\ d(x_i, x_j) \leq 0.5\}, \quad \forall x_j \in U. \end{aligned} \tag{3}$$

Then we obtain the following five binary classifications.

$$\begin{aligned} U/R_1 &= \{\{x_1, x_2, x_3\}, \{x_4, x_5\}\}, \\ U/R_2 &= \{\{x_1, x_2, x_3\}, \{x_4, x_5\}\}, \\ U/R_3 &= \{\{x_2, x_3, x_4\}, \{x_1, x_5\}\}, \\ U/R_4 &= \{\{x_1, x_2, x_3, x_4\}, \{x_5\}\}, \\ U/R_5 &= \{\{x_4, x_5\}, \{x_1, x_2, x_3\}\}. \end{aligned} \tag{4}$$

Note that these five classifications are derived independently. Objects such as x_1 and x_2 are classified as indiscernible in U/R_1 and U/R_2, but classified as discernible in U/R_3.

[End of example]

3.2 Indiscernibility Level

The family of binary classifications $U/\mathbf{R}$, where $\mathbf{R} = \{R_1, R_2, \ldots, R_N\}$, produces the finest sets of objects as it takes intersection of all binary classifications. In this scheme, objects fall into the same category in $U/\mathbf{R}$ only when all of the N relations agree to classify them as indiscernible. If there is at least one relation that discriminate them, they are discernible in $U/\mathbf{R}$ even when other $N-1$ relations agree to classify them as indiscernible.

Now recall the example shown previously in Eq. (4). This example contains three types of binary classifications: U/R_1 ($= U/R_2 = U/R_5$), U/R_3 and U/R_4. Since they are slightly different, classification of U by the family of binary classifications $\mathbf{R}$, $U/\mathbf{R}$, results in producing four very small, almost independent categories.

$$U/\mathbf{R} = \{\{x_1\}, \{x_2, x_3\}, \{x_4\}, \{x_5\}\}. \tag{5}$$

This simple example reveals the following problems that prevent us from observing data with appropriate granularity.

1. Binary classifications are defined independently; thus global relationships between each classification is not taken into account.
2. Binary representation of indiscernible/discernible makes it difficult to reflect the global agreement for classifying objects.

We here introduce *indiscernibility level*, a novel measure that solves the above problems and makes it possible to represent the granularity of objects while keeping the use of independently defined binary classifications. The indiscernibility level, $\gamma(x_i, x_j)$, defined for a pair of objects x_i and x_j, quantifies the ratio of binary classifications that agree to classify x_i and x_j as indiscernible. The higher level of indiscernibility implies that although there is small number of counterview, they are likely to be treated as indiscernible, and vise versa.

The *indiscernibility level* $\gamma(x_i, x_j)$ for objects x_i and x_j is defined as follows.

$$\gamma(x_i, x_j) = \frac{\sum_{k=1}^{|U|} \delta_k^{indis}(x_i, x_j)}{\sum_{k=1}^{|U|} \delta_k^{indis}(x_i, x_j) + \sum_{k=1}^{|U|} \delta_k^{dis}(x_i, x_j)}, \tag{6}$$

where

$$\delta_k^{indis}(x_i, x_j) = \begin{cases} 1, \text{ if } \ (x_i \in [x_k]_{R_k} \wedge x_j \in [x_k]_{R_k}) \\ 0, \text{ otherwise.} \end{cases} \tag{7}$$

and

$$\delta_k^{dis}(x_i, x_j) = \begin{cases} 1, \text{ if } \ (x_i \in [x_k]_{R_k} \wedge x_j \notin [x_k]_{R_k}) \ \text{ or} \\ \quad \text{if } \ (x_i \notin [x_k]_{R_k} \wedge x_j \in [x_k]_{R_k}) \\ 0, \text{ otherwise.} \end{cases} \tag{8}$$

Table 2. Indiscernibility level γ for objects in Eq. (4)

	x_1	x_2	x_3	x_4	x_5
x_1	3/3	3/4	3/4	1/5	0/4
x_2		4/4	4/4	2/5	0/5
x_3			4/4	2/5	0/5
x_4				3/3	1/3
x_5					1/1

Equation (7) means that $\delta_k^{indis}(x_i, x_j) = 1$ holds only when x_i and x_j are *indiscernible* on U/R_k under the condition that they are also indiscernible with x_k. Equation (8) means that $\delta_k^{dis}(x_i, x_j) = 1$ holds only when x_i and x_j are *discernible* on U/R_k, under the condition that either of them is indiscernible with x_k. By taking the sum of $\delta_k^{indis}(x_i, x_j)$ and $\delta_k^{dis}(x_i, x_j)$ for all $k(1 \leq k \leq |U|)$ as in Equation (6), we obtain the ratio of binary classifications that agree to treat x_i and x_j as indiscernible objects. Note that in Equation (7), we excluded the case when x_i and x_j are indiscernible but not indiscernible with x_k. This is to exclude the case where R_k does not significantly put weight on discerning x_i and x_j. P_k for U/R_k is often determined by focusing on similar objects rather than dissimilar objects. This means that when both of x_i and x_j are highly dissimilar to x_k, their dissimilarity is not significant for x_k. Thus we only count the number of binary classifications that certainly evaluate the dissimilarity of x_i and x_j.

[Example 2]: Indiscernibility Level

The indiscernibility level $\gamma(x_1, x_2)$ of objects x_1 and x_2 in Example 1 is calculated as follows.

$$\begin{aligned}\gamma(x_1, x_2) &= \frac{\sum_{k=1}^{5} \delta_k^{indis}(x_1, x_2)}{\sum_{k=1}^{5} \delta_k^{indis}(x_1, x_2) + \sum_{k=1}^{5} \delta_k^{dis}(x_1, x_2)} \\ &= \frac{1+1+0+1+0}{(1+1+0+1+0)+(0+0+1+0+0)} \\ &= \frac{3}{4}. \end{aligned} \tag{9}$$

Let us explain this example with the calculation of the numerator (1+1+0+1+0). The first value 1 is for $\delta_1^{indis}(x_1, x_2)$. Since x_1 and x_2 are indiscernible on U/R_1 and obviously they are in the same class to x_1, $\delta_1^{indis}(x_1, x_2) = 1$ holds. The second value is for $\delta_2^{indis}(x_1, x_2)$, and analogously, it equals 1. The third value is for $\delta_3^{indis}(x_1, x_2)$. Since x_1 and x_2 are discernible on U/R_3, it becomes 0. The fourth value is for $\delta_4^{indis}(x_1, x_2)$ and it obviously, becomes 1. The last value is for $\delta_5^{indis}(x_1, x_2)$. Although x_1 and x_2 are indiscernible on U/R_5, their class is different to that of x_5. Thus $\delta_5^{indis}(x_1, x_2)$ becomes 0.

Indiscernibility levels for all of the other pairs in U are tabulated in Table 2. Note that the indiscernibility level of object x_i to itself, $\gamma(x_i, x_i)$, will always be 1.

[End of example]

3.3 Hierarchy of Indiscernibility Level

The indiscernibility level can be used with thresholding to change an observation scale of data. The threshold value Th_γ determines the indiscernibility level under which objects are considered to be indiscernible. Since treating two objects as indiscernible is equal to merge the two objects, and it is a stepwise abstraction process that goes hierarchically from bottom to top according to the indiscernibility level, conventional single-linkage hierarchical grouping/clustering method can be applied.

Agglomerative hierarchical clustering (AHC) initially assigns an independent cluster to each object. Then it seeks the most similar pair of objects and merges it into one cluster. This process is repeated until all of the initial clusters are merged into single cluster. AHC has several criteria such as single-linkage and complete-linkage for selecting objects to merge. Single-linkage selects the pairs according to the intergroup dissimilarity of the closest pairs:

$$d_{SL}(G, H) = \min_{x_i \in G, x_j \in H} d(x_i, x_j),$$

where G and H are clusters to be merged in the next step. The clustering is also called nearest-neighbor technique. Complete-linkage selects the pairs according to the intergroup dissimilarity of the furthest pairs:

$$d_{CL}(G, H) = \max_{x_i \in G, x_j \in H} d(x_i, x_j)$$

According to the definition of indiscernibility level, any objects (x_i, x_j) whose indiscernibility level exceeds the threshold value, namely, if $\gamma(x_i, x_j) \geq Th_\gamma$ holds, they should be treated as *indiscernible*. We chose the nearest-neighbor criterion to ensure this very property of γ.

This hierarchical merge/abstraction process produces a dendrogram that represents hierarchy of indiscernibility level over all objects. By setting an appropriate thresholding level on the dendrogram, one can obtain abstracted groups of objects that meet the given level of indiscernibility. Namely, one can interactively change the granularity of data. The lowest threshold produces the finest groups of objects (granules) and the highest threshold produces the coarsest groups of objects.

[Example 3]: Hierarchy of Indiscernibility Level

Let us recall the case in Example 2. The matrix of indiscernibility levels is provided in Table 2. For the sake of easy understandings, we provide in Table 3 recalculated values.

Here we treat the indiscernibility level as similarity because the mergence should proceed in decreasing order of $\gamma(x_i, x_j)$. If one prefers to treat it as dissimilarity, simply use $1 - \gamma(x_i, x_j)$ instead of $\gamma(x_i, x_j)$.

Table 4 and Figure 1 provide the detail of merging process and the dendrogram respectively. Since $\gamma(x_2, x_3) = 1.0$, these objects are indiscernible at the lowest level; thus $\{x_1\}, \{x_2, x_3\}, \{x_4\}, \{x_5\}$ constitute the finest sets of objects (granules). At $\gamma = 0.75$, x_1 becomes indiscernible with x_2. Since x_2 and x_3

Table 3. Indiscernibility level γ for objects in Eq. (4)(recalculated)

	x_1	x_2	x_3	x_4	x_5
x_1	1.0	0.75	0.75	0.2	0.0
x_2		1.0	1.0	0.4	0.0
x_3			1.0	0.4	0.0
x_4				1.0	0.33
x_5					1.0

Table 4. Indiscernibility level γ for objects in Eq. (4)(recalculated)

Step	pairs	γ	clusters
1	x_2, x_3	1.0	$\{x_1\}, \{x_2, x_3\}, \{x_4\}, \{x_5\}$
2	x_1, x_2	0.75	$\{x_1, x_2, x_3\}, \{x_4\}, \{x_5\}$
3	x_2, x_4	0.4	$\{x_1, x_2, x_3, x_4\}, \{x_5\}$
4	x_4, x_5	0.33	$\{x_1, x_2, x_3, x_4, x_5\}$

are also indiscernible, $\{x_1, x_2, x_3\}, \{x_4\}, \{x_5\}$ constitute an abstracted sets of objects. Similarly, at $\gamma = 0.40$, x_4 becomes indiscernible with x_2 and $\{x_1, x_2, x_3, x_4\}, \{x_5\}$ constitute the more abstracted sets of objects. Finally, at $\gamma = 0.33$, all objects are considered to be indiscernible and the most abstracted set is obtained.

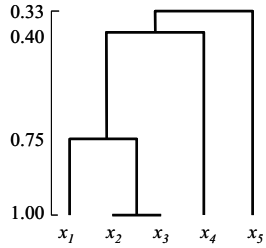

Fig. 1. Dendrogram for Example 3.

The level of abstraction can be interactively set by changing the threshold value Th_γ on the dendrogram. For example, in Figure 1, one can set $Th_\gamma = 0.5$ as a reasonable level since the difference of γ between steps is relatively large.

[End of example]

4 Experimental Results

We applied our method to a synthetic data and examined its usefulness. The dataset contained 19 objects in two-dimensional Cartesian space as shown in

Figure 2. The dataset was generated by Neyman-Scott method [7] with cluster number = 3. The label 'cls 1' to 'cls 3' shows the original class that each object belongs to.

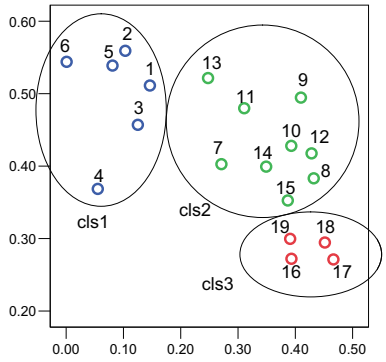

Fig. 2. 2D plot of the test data.

The proposed method starts with determining a binary classification, U/R_i, for each object x_i, $i = 1, 2, \ldots, 19$. In order to seclude the inference of methods/parameters for determining U/R_i, we used the perfect binary classifications, which were generated based on the class labels of data. Then, in order to simulate the non-Euclidean properties, we applied random disturbance to the perfect binary classifications. Taking the randomly disturbed perfect classifications as input, we calculated the indiscernibility levels and constructed a dendrogram.

Perfect binary classifications We prepared a set of binary classifications called perfect binary classifications, which can classify the data into correct groups. A perfect binary classification U/R_i for object x_i was defined as follows.

$$U/R_i = \{P_i,\ U - P_i\}, \tag{10}$$

where

$$P_i = \{x_j |\ c[x_i] = c[x_j]\}, \quad \forall x_j \in U. \tag{11}$$

where $c[x_i]$ denotes the class label of x_i assigned when creating the dataset. Obviously, the variety of perfect binary classifications in $U/\mathbf{R}$ was equal to the number of classes in the dataset, because if objects x_i and x_j belonged to the same class, R_i and R_j became identical.

After assigning a perfect binary classification to each object, we applied random disturbance to each of them. We designed the following three disturbance operations.

Table 5. Binary classifications for the test data.

x_i	P_i of U/R_i	x_i	P_i of U/R_i
x_1	1,2,4,5,6,15	x_{11}	7,8,9,10,11,12,13,14,15,12
x_2	1,2,3,4,5,4	x_{12}	7,8,9,10,11,13,14,15
x_3	1,2,3,4,5,6,6	x_{13}	7,8,9,10,11,12,13,14,15,6
x_4	1,2,3,4,5,6,12	x_{14}	7,8,9,10,12,13,14,15,15
x_5	1,2,3,4,5,6,19	x_{15}	7,8,9,10,11,12,13,14,15,6
x_6	1,2,3,5,6,14	x_{16}	16,17,18,19
x_7	7,8,9,10,11,13,14,15	x_{17}	16,17,18,19
x_8	7,9,10,11,12,13,14,15	x_{18}	16,17,18,19
x_9	7,8,9,11,12,13,14,15	x_{19}	16,17,18,19
x_{10}	7,8,9,10,11,12,13,14		

Table 6. Indiscernibility level γ for the test data.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	1.00	1.00	0.83	0.83	1.00	0.63	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.07	0.07	0.00	0.00	0.00	0.10
2	1.00	1.00	0.83	0.83	1.00	0.63	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.07	0.07	0.00	0.00	0.00	0.10
3	0.83	0.83	1.00	0.67	0.83	0.50	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.07	0.00	0.00	0.00	0.00	0.11
4	0.83	0.83	0.67	1.00	0.83	0.50	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.00	0.08	0.00	0.00	0.00	0.11
5	1.00	1.00	0.83	0.83	1.00	0.63	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.07	0.07	0.00	0.00	0.00	0.10
6	0.63	0.63	0.50	0.50	0.63	1.00	0.14	0.15	0.14	0.15	0.15	0.25	0.14	0.21	0.23	0.00	0.00	0.00	0.09
7	0.00	0.00	0.00	0.00	0.00	0.14	1.00	0.89	1.00	0.89	0.89	0.70	1.00	0.90	0.80	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.15	0.89	1.00	0.89	0.78	0.78	0.60	0.89	0.80	0.70	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.14	1.00	0.89	1.00	0.89	0.89	0.70	1.00	0.90	0.80	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.15	0.89	0.78	0.89	1.00	0.78	0.60	0.89	0.80	0.70	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.15	0.89	0.78	0.89	0.78	1.00	0.60	0.89	0.80	0.70	0.00	0.00	0.00	0.00
12	0.08	0.08	0.08	0.08	0.08	0.25	0.70	0.60	0.70	0.60	0.60	1.00	0.70	0.64	0.55	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.14	1.00	0.89	1.00	0.89	0.89	0.70	1.00	0.90	0.80	0.00	0.00	0.00	0.00
14	0.07	0.07	0.07	0.00	0.07	0.21	0.90	0.80	0.90	0.80	0.80	0.64	0.90	1.00	0.73	0.00	0.00	0.00	0.00
15	0.07	0.07	0.00	0.08	0.07	0.23	0.80	0.70	0.80	0.70	0.70	0.55	0.80	0.73	1.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	0.80
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	0.80
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	0.80
19	0.10	0.10	0.11	0.11	0.10	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.80	0.80	0.80	1.00

1. Delete: Randomly select one element in P_i and remove it from P_i.
2. Add: Randomly select one element from U and add it to P_i.
3. Replace: Randomly select one element from P_i and replace it with randomly selected element in U.

The operation was randomly selected each time. Disturbance operation was repeated $card(P_i) \times \rho$ times for each binary classification, where ρ denotes disturbance ratio. We choose $\rho = 0.2$, which meant that about 1-2 objects in each prefect binary classification were subject for random disturbance. Table 5 provides all the disturbed binary classifications ($U - P_i$ omitted for simplicity. x of x_i also omitted in P_i for simplicity).

Using the binary classifications in Table 5, we calculated indiscernibility level for each pair of objects and obtained a matrix as shown in Table 6. Then we generated the dendrogram shown in Figure 3.

At the lowest level of indiscernibility, 13 sets of objects were generated as the finest granules because the randomly disturbed binary classifications were

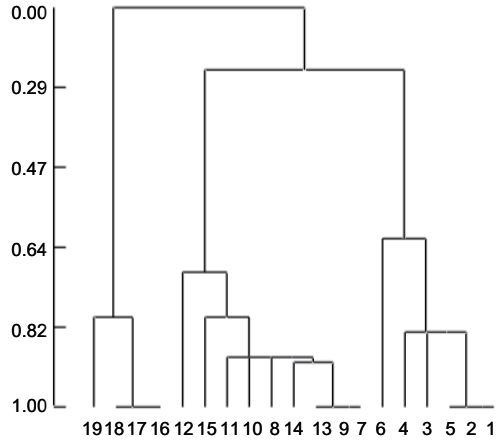

Fig. 3. Dendrogram for the Test data.

slightly different each other. However, their disturbance affected locally; therefore the binary classifications retrained the ability of classifying most of the object pairs correctly. In other words, although there exist a few counterviews, object pairs in the same class retained higher level of agreement among binary classifications to be classified as indiscernible objects. Therefore, if we changed the threshold of indiscernibility level to a slightly lower value, for example to 0.8, most of the object pairs could recover their original indiscernibility. The shapes of dendrogram around the bottom part visualizes this characteristics. As the threshold level decreases (goes toward upper direction on the dendrogram), the granularity of the data quickly became coarser, and then became stable for $Th_\gamma = 0.63$ to 0.25. For these values, the method generated correct clusters, which corresponded to the appropriate level of object granularity. If we further set Th_γ to lower value, objects with very low indiscernibility became merged and excessively abstracted sets would be obtained.

The above results demonstrated that (1)the proposed method could visualize the hierarchy of indiscernibility using dendrogram, (2) by changing the threshold level on the dendrogram, users could interactively change the granularity of objects defined based on the indiscernibility level, and (3) the method could handle non-Euclidean relational data in which asymmetry and local disturbance of the triangular inequality could occur.

5 Conclusions

In this paper we have presented a new approach of representing hierarchy of data granularity by the combination of the indiscernibility level and single-linkage

AHC. The indiscernibility level quantified the level of global agreement for classifying two objects into the same category as indiscernible objects. Single-linkage grouping merged objects according to the indiscernibility level from bottom to top and constructed the dendrogram, which enabled users to examine the hierarchy of data granularity and obtain the set of indiscernible objects that meets the given level of granularity. Additionally, it put its basis on binary classification assigned independently to each object; therefore, it can be applied to non-Euclidean, asymmetric relational data. Experimental results on a small synthetic data demonstrated the above characteristics. It remains as a future work to apply this method to other real-world data, and to compare the performance with other methods such as NERFCM [4].

References

1. H. C. Romesburg (1989): Cluster Analysis for Researchers. Krieger Publishing Inc.
2. B. S. Everitt, S. Landau, and M. Leese (2001): Cluster Analysis Fourth Edition. Arnold Publishers.
3. P. Berkhin (2002): Survey of Clustering Data Mining Techniques. Accrue Software Research Paper. URL: http://www.accrue.com/products/researchpapers.html.
4. R. J. Hathaway and J. C. Bezdek (1994), "NERF c-means: Non-Euclidean relational fuzzy clustering," *Pattern Recognition*, 27(3): 429–437.
5. Z. Pawlak (1991): Rough Sets, Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht.
6. S. Hirano and S. Tsumoto: An indiscernibility-based clustering method with iterative refinement of equivalence relations - rough clustering -. Journal of Advanced Computational Intelligence and Intelligent Informatics **7** (2003) 169–177
7. J. Neyman and E. L. Scott (1958): "Statistical Approach to Problems of Cosmology," *Journal of the Royal Statistical Society*, Series B20: 1–43.

An Integrated Web-Query Classification Approach Based on Rough Sets

Ernestina Menasalvas*[1], Santiago Eibe**[1], Maria Valencia[1], Pedro Sousa[2]

[1] Facultad de Informatica, Universidad Politecnica, Madrid, Spain
emenasalvas@fi.upm.es, seibe@fi.upm.es, mvalencia@zipi.fi.upm.es
[2] Universidad Nova de Lisboa, Portugal
pas@uninova.pt

Abstract. Categorization of web search queries is of increasing interest not only due to increasing the effectiveness and efficiency of returned results but also for the potential revenue in coupled applications such us targeted advertising or reformulation of the query for better results.
Nevertheless, categorization of the queries only based on the features of the query so far is challenging due to the small number of words in queries and the dynamism of sites and users requests.
We define features of the queries based on visibility and decay of the terms they contain to be integrated with a taxonomy of concepts based on the site structure. The proposed method is basically composed of two steps: firstly some queries are categorized according to the results they return and a fast classifier is built based on these results. In a second stage, the method exploits properties of the visibility of the terms in the queries and in the front page along a period, to obtain a set of attributes that are used to further cluster queries and enrich classification. Altogether they integrate an online categorization of queries to help the site decision for targeted advertising. The classifier builder is based on Rough Set techniques integrated with traditional K-means and decision trees. Results of the classification on a site serving news and using GSA as search engine is shown.
Keywords. Query Categorization, Rough Sets, search engines, ubiquitous data mining

1 Introduction

Many search engine companies are interested in providing commercial services in response to user queries, including targeted advertisement, product reviews or any other value added services such as banking and transportation [7]. In [24] the authors remark that the length of Web queries coupled with the constantly changing distribution and vocabulary of queries hinders traditional text classification posing a challenge on successful query classification. In fact is difficult to determine the user's desired task and information needed from the very short

* Project partially financed by Project TIN2004-05873
** Project partially financed by Project CCG06-UPM/ESP-0259

Web queries. A lot of attention has been put in Web search, to organize the large number of Web pages in the search result after the user issues a query, according to the potential categories of the results.

Successfully mapping incoming user queries to topical categories, particularly those for which the search engine has domain-specific knowledge, can bring improvements in both the efficiency and the effectiveness of Web search [2]. Several studies have attempted to classify user queries in terms of the informational actions of users. In early work [5], manually classified a small set of queries into transactional, navigational, and informational tasks.

The other challenge in query classification is deciding the categories towards the classification will be performed. When dealing with queries submitted to a web site, the site structure itself can be taken as a possible categorization. The problem is specially of interest in web sites dedicated to news. In this case the level of dynamism in users and request due to inherent changing nature of pieces of news is even higher hindering the process of query categorization. If the site wants to enrich the result of the search with added values, any information related to the user context is valuable to better respond to the user needs. According to [25] making the web truly responsive to human needs means understanding the user needs. The author mentions the problem of interpreting multiple domains, contents and purposes that can lead to different applications interpreting the same meta data in different ways.

We concentrate on on-line classification of queries into categories that are of potential interest to the site sponsor so to be used later for marketing purposes. We attack that problem presenting a double stage method. First, we present a fast on line classification scheme that labels queries into the site structure based taxonomy categories taking into account the words available in the query. In a second stage, information of events on the front page together with measures of the words visibility and decay are used to build an improved classifier. Rough sets are known to be a good tool for mining although efficiency problem has to be tackled. Consequently in this paper we combine rough set techniques with decision trees and traditional k-means and two-step algorithm.

The rest of the paper is organized as follows. In section 2 a review of previous studies on categorization of queries as well as the application of rough sets in web mining problems is found. In section 3 the presented approach is presented, firstly we define the basics of the method and later the stages of the process. In section 4 results of the proposed method in a real web site that uses GSA [8] as search engine is shown. To conclude, section 5 presents the results obtained so far in mentioned news site.

2 Related Work

2.1 Mining News Sites

The structure of the most visited regions of the Web is altered at the timescale from hours to days. In [9] the authors analyze the dynamics of visits of a major

news portal, representing the prototype for such a rapidly evolving network. In [6] a topic mining framework that supports the identification of meaningful topics from news stream data is proposed. The approach presented is based on retrieving News articles and applying data mining to produce patterns that are stored in a data base. The clustering technique proposed is incremental so to deal with the high rate of documents update. In [26] author propose a two-way clustering technique based on probabilistic theory to address the problem of analyzing web log data collected at a typical on-line newspaper site. More general, in [23] the problem of reasoning about information changes is considered in the context of complex concepts approximated hierarchically and actions that can be triggered to change properties of investigated objects.

2.2 Rough Sets and Web Mining

The rough set concept [20, 21] is a mathematical tool proposed by Prof. Pawlak in the earlies 80's to reason about vagueness and uncertainty. The Rough Set theory bears on the assumption that in order to define a set we need initially some knowledge about elements of the universe. In Rough Set theory a concept is described by its *lower* and *upper* approximations defined with respect to some indiscernibility relation.

A detailed review explaining the state of the art and the future directions for web mining research in soft computing framework is provided by Pal et al. [19] and Mitra [15]. The use of Rough Sets theory for knowledge discovery and web mining is widely acknowledged [1] with special interest on Information Retrieval and Association/Clustering of queries. In [13] how rough set theory can be used to develop clustering schemes for web mining is presented. The unsupervised classification described in the paper uses properties of rough sets along with Genetic Algorithms to represent clusters as interval sets. Later the same author in [14] presents a variation of the k-means algorithm in which clusters are represented as rough sets as well as the promising results of applying the approach to cluster web visitors. The approach we present for classification also uses rough set techniques and clustering methods but in our case we do not modify the original algorithms but we integrate the results so rough sets are used to find the most discriminant attributes required for classification.

In [16] a method of snippet representation enrichment using Tolerance Rough Set model is presented. The proposed method is used to construct a rough set based search result clustering algorithm that is compared it with other methods.

In [18, 17] Carrot2 and Lingo algorithms for clustering web search results emphasizing cluster description quality are presented. The algorithms are based on algebraic transformations of the term-document matrix and frequent phrase extraction using suffix arrays.

To provide an effective solution to avoid disambiguities and duplication of information (terms in queries), in [12] the relationship between association rules and rough set based decision rules is defined proving that a decision pattern is a kind of closed pattern. It then presents a novel concept of rough association rules in order to improve the effectiveness of association rule mining.

2.3 Categorization of queries

In [3] an approach to topical categorization of general web queries in order to maintain sufficient categorization recall when web queries are short is presented. The approach is based on 2 principles: using the query logs alone this means not using external sources and integration of classification techniques. This works extends that of [2] that introduces the use of a web search engine log as a source of unlabelled data to aid in automatic classification. The process presented by this authors is robust to changes in the stream of data as the only source of information for classification are the queries themselves that is the very thing that changes the most. Our approach is similar to this one in the sense that the fast classifier is also only based on the queries themselves. Nevertheless, in our approach we also use information related to the front page so to help the categorization of queries.

In [4] a method that exploits page counts and text snippets returned by the search engine are used to define different scores to compute similarity of words. These scores are integrated later with support vector machines leading a robust similarity measurement that improves significantly the F-measure in entity disambiguation tasks and in community mining task. In [10] an approach to add useful meta data to search results by fast-feature techniques is explored. The main motivation of the paper is to stress the importance of lightweight rapid techniques for categorizing search results into meaningful and stable categories. The approach presented in this paper also deals with a fast-feature categorization method but instead of being based on features of the results we based our approach on features of the submitted queries.

In [22] an approach called *query enrichment* to deal with the problem categorization of short and ambiguous queries is presented. The proposed method takes a short query and maps it to intermediate objects and from the intermediate object to the target category. For the query enrichment the method integrates information of different search engines. As in this approach the method that we present trust the search engine for a first categorization of queries, but in our approach we only submit queries to one search engine for manual categorization and the enrichment is made later by means of visibility measures.

3 The proposed method

3.1 Outline

We present a method to categorize the queries submitted to a search engine searching for results across the corporate content of a company. This is the case for example of the GSA [8] (Google Search Appliance) that delivers results to users across virtually all the information in a company. In this case, the results of the search is determined by the contents the company has indexed and should be related to the site structure. This is the first assumption under the approach we present: the taxonomy of concepts can be extracted from the site structure.

The second assumption as in [3] is that no external sources are used for categorization but the queries themselves and the information of the site structure. As in [10] we also used fast-feature techniques for the classification but in our case based on features of the query itself and we do not analyze the result set. We concentrate on classifying queries into categories that are of potential interest to the site sponsor so to be used later for marketing purposes. This is to say the categorization will be used to choose the best ads to display together with the results of the query. We partly inspire this study in the one found in [22].

3.2 Taxonomies of concepts

We propose to deal with two kind of categories:

- The one inherent in the site structure: this is very stable taxonomy as the concepts and structure of the site does not change very often
- The one extracted from the front page: this is a very changing taxonomy but it reveals interesting information regarding the dynamics of the users and their interests

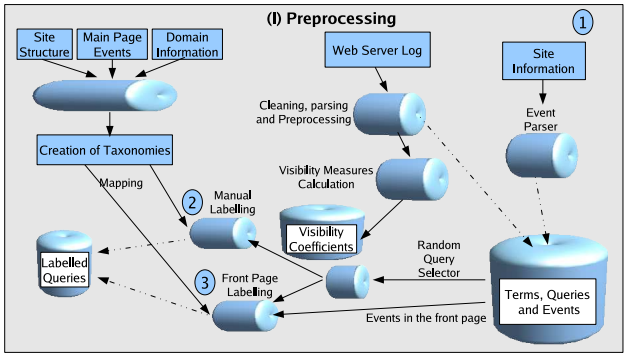

Fig. 1. Preprocessing Activities

In order to categorize queries according to these taxonomies we propose to randomly choose a set of queries and label them manually and from those extract a fast classifier. Due to dynamism of sites and users we assume a set of words to appear in a continuous way that will be present in the queries prior to their appearance in the site or in the front page. On the other hand, words referring to current and fashionable topics will have decaying interest. To deal with this and other facts we propose a set of measures to score the visibility of terms that will help to improve the fast classifier efficiency.

3.3 Word appearance-count-based visibility

In [11] the author defines the visibility of a story based on the number of votes it receives and based on it, interestingness measure coefficient is calculated. In our approach we try to measure the interest of queries based on the words it carries. From this coefficient one can calculate the potential value the query has for the site and accordingly act.

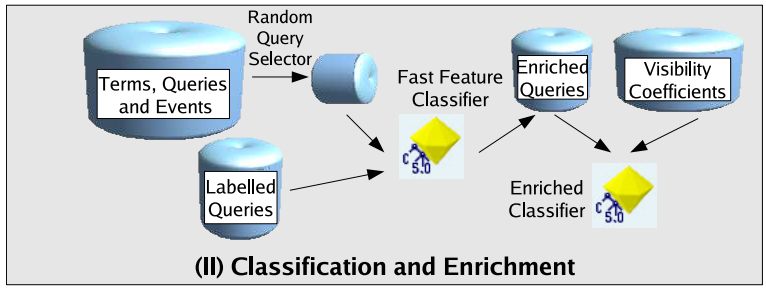

Fig. 2. Enrichment Process

Interestingness of the query highly depends on features of the terms contained in the query. Consequently we firstly categorize terms and then according to them we label queries. Two criteria are taken into account:

- visibility of the term on the queries
- visibility of the term on the front page

Consequently we define the *visibility* of a term in a period of time $V(t, p)$ based on its appearance frequency (number of times it appears) in the period. According to this coefficient we define:

- stable terms in a period $s_t(p)$: those terms t that appear along the period p under study in uniform way. Uniformity is a function of the average number of appearance of the terms in the period. It also depends on the decay function of the term $d(t, p)$ that reflects the ratio of appearance along time. Values of decay over 1 means increasing ratio of appearance, values close to 0 means decreasing ratios and values around 1 means stable term
- new term $n_t(p)$: a term t belongs to this set if it appears for the first time in the period p
- top-ten: one term t belongs this set if in the period p appears over the average ratio of appearance

All these sets are calculated for different period length: 24 hours, 7 days, 31 days, ... and both for the appearance of words in heading in the front page and in queries. Queries and later sessions are enriched with information regarding the terms they contain: 24 hours top ten, preceding day, week before, ... This attributes will be used for the enriched categorization to deal with dynamism of words and users.

3.4 Approach

The method to classify queries in a fixed site makes the following assumptions:

- The site structure inherently hides domain information that can be used to build a taxonomy of concepts
- Given that the search engine is indexing the documents to be retrieved, categorization process can be based on the results the search engine returns in a query. Results of the engine can be mapped into concepts of the taxonomy. In other words, we trust that results of the engine can be mapped in the site structure. By this assumption, our approach first categorizes a query by covering its potential meanings in the site structure, and then classify the search results to the domain defined categories
- Contents located in the main page correspond to current events or/and fashionable topics. The structure of the main page can also be associated with a taxonomy of concepts. This taxonomy can be used to be able to categorize queries according to front page visibility
- The dynamism of the web makes content and words change meaning. News like "Killer dogs" makes reference one day to an accident in which a child was death by a dog and following month can refer special treatment for dogs. The result of evolution due to time should be combined into a coherent whole for better coverage and higher categorization robustness

According to these assumptions the method we propose is composed of the following stages (see figures 1 and 2 for detail):

1. Off-line pre-processing:
   - Taxonomies creator
   - Web Log cleaning: this process has to be repeated due to the stream nature of the log data
   - Visibility sets calculation: as in the previous step, sets has to be updated as new data arrived and are cleaned
2. Manual labelling of queries
   - Search sets of results for a selected set of queries: Randomly some queries are chosen and are sent to the search engine to analyze the set of results.
   - Mapping of the results into domain taxonomy. We propose to use only the urls of the result set for the mapping in the taxonomy as in [3].
3. Front-page labelling of queries. For those queries containing terms of the headings in front page, randomly some queries are chosen and manually labelled
4. Fast-feature classifier: We call it fast-feature as only the words contained in the query are used to build the classifier
   - Reduct calculation. Sets of words that appear in the categories chosen are used so to decide which of them are discriminant for each category. We use rough set based techniques for this step
   - Rough Sets based classifier. Based on the results we propose to build the classifier

– Build final model
5. Enrichment of the queries regarding to visibility functions
6. Enriched classifier
7. Evaluation

4 Case study: a News site

4.1 Data Set description

To test the proposed method, we conducted experiment of a real dataset extracted from a News Site that uses GSA [8] as search engine. We test with its logs as they have all the features of dynamism, shortness and ambiguity that we discuss when proposing the method.

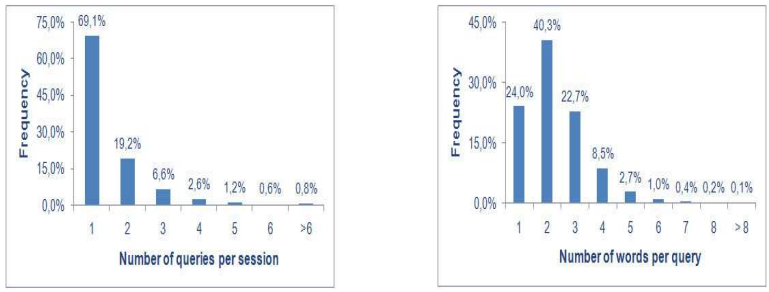

Fig. 3. Frequency of queries and sessions with different lengths

The weblog used corresponds to the activity of the site along 4 months containing a total of 442909 records. After cleaning and preprocessing the log, 106443 different queries have been identified in which a total of 43742 different terms are used. In figure 3 information regarding the number of words contained in each query and their associated frequency is shown, together with the number of queries per session.

The figure 3 shows that queries containing two words are the most frequent (40,3%). Furthermore, 96% of the queries have no more than four words. the queries contains different combinations of terms that include names of TV programs, URLs, dates, TV series actors, ... From the preprocessing was also observed that there are very frequent terms that appear misspelled many times and this had to be marked and corrected prior to categorization. On the other hand, we could also observed the incidence of the evolution of news in some submitted queries and on the different categorization the site makes as they evolve (for example prior of launching a new tv programme it appears to be as Top Hit in the front page but as soon as it is running it goes under TV series category.

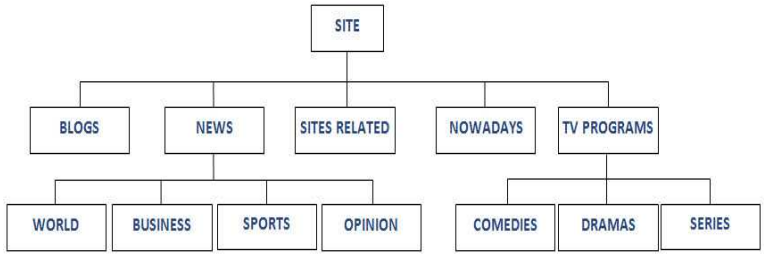

Fig. 4. Site Structure

4.2 The method

Preprocessing of logs The site structure was analyzed to extract the taxonomy of concepts in a first stage and the first level of the taxonomy is depicted in figure 4. It is important to note at this point that analyzing other News site such as (http://www.telecinco.es/, http://www.canalrcn.com/, http://www.bbc.co.uk/, www.canalcaracol.com.co) they all have in common a very similar structure of the site at least for the first level of the hierarchy.

On the other hand, the front page was also analyzed to obtain the way in which headlines are located on it and the meaning of this location. The concepts extracted will be later used to label queries.

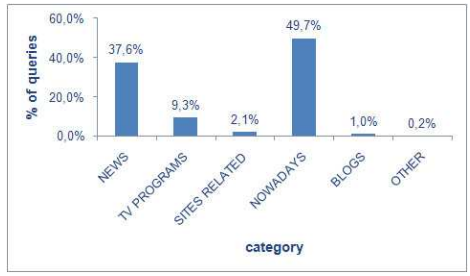

Fig. 5. Categories Distribution

The web log was cleaned eliminating for example duplicated records and applying basic lexicographical correction. Once the logs was considered to be cleaned, the visibility functions were applied obtaining the lists of stable words, top-ten and disappearing terms to name some for different length periods. It is worthnoting that terms related to names of TV series that seems to be top ten in the country such as "The ugly Betty" belongs to the more stable words along the period of time under study. On the other hand, those terms related to some breaking news such as "Hussein" are the ones in which the decay function

changes more abruptly in a short period of time. To end with, the list of words appearing one day makes with a high degree of confidence, makes reference to some headline that appears later on the day in the front page or the following day the latest.

To end the preprocessing stage the set of records was splitted into two sets: $D1$ and $D2$ of approximately the same length (around 130.000 records each). The first one $D1$ will be used to generate the categorized records and the fast feature classifier and the second one $D2$ will be used for testing and improving the first classifier. On order to properly test results the two sets corresponds to different periods of time, this is to say the first set corresponds to the first two months of activity gathered and the second datasets corresponds to the next two months. This way, we will be able to test how the dynamism and evolution of terms affects the results.

Categorization of queries and fast feature classifier From the first dataset $D1$ we randomly select different datasets that were submitted to the GSA and taking into account the urls given as result they were manually categorized. On the other hand, records were also categorized as having terms related to the front page or not and in the first case, the corresponding category was assigned. The distribution of categories for the categorized set can be seen in figure 4.

Once the categories are obtained the next step is to build the classifier. The main problem was the large number of terms belonging to each category. Consequently, at this point visibility coefficients were used so to choose significantly smaller number of terms for each category and they were grouped according to the associations that frequently were together. With this information reducts were calculated to extract the sets of words that best discriminate each category. The rule set obtained categorized properly $73,5$ % of the records observing that records that could not be categorized correspond to new terms appearing the day when the record is submitted in most of the cases. Consequently, the enriched classifier is obtained to avoid this problem.

Enriched classifier To increase the accuracy of the model obtained only using terms, information related to visibility coefficient is used to enrich the classification power. The process we performed is as follows:

1. We enrich the records with information related to visibility coefficients and we add a flag to establish where such a record was successfully categorized by the classifier.
2. We obtain a classification model that given a record predicts if the record will be or not classified
3. In case it will be classified we run the classification
4. For records not classified we build a classifier taking into account the information on the front page

We have run the proposed method on the dataset obtaining the following results: The reduct: Visibility-front-page-24, new-term-24, frequent-top-ten,

frequent-30 is enough to discriminate possibility of categorization with confidence of $0,89$. The classification power of the fast-feature classifier increases then to obtain an accuracy of $0,875\%$. The classifier obtained for not categorized records is highly dependant on terms on the front page and, consequently, the more often it gets updated the better results. In our case, we made experiments updating the classifier for a month, week, and day. Experiments on a week basis show to be almost equally efficient as the day one, $0,84$ % of the not categorized records in the first case where now classified.

5 Conclusion and Outline

Query categorization has increased value in those sites using internal web search as a service for users as the company can deliver together with the result some added value service to the user. Examples of the services can be reformulation of the queries so to obtain better results or marketing services related to the topic of the submitted query. Nevertheless, due to the shortness and ambiguity of the queries, categorization is a challenging problem. In this paper we have proposed a method for query categorization based on the words the query contains and about terms visibility coefficients to improve the classification performance of terms alone. We have tested the method on the logs of a News site containing 442.909 records, and we have tried different techniques for the classification step. In particular C4.5, K-means, Apriori and Rough Set based methods where used. Rough sets based methods shown to be more time consuming but the accuracy and error rates make them a good candidate for model generator where time is not a problem.

References

1. Dragos Arotaritei and Sushmita Mitra. Web mining: a survey in the fuzzy framework. *Fuzzy Sets and Systems*, 148(1):5–19, 2004.
2. Steven M. Beitzel, Eric C. Jensen, Ophir Frieder, David Grossman, David D. Lewis, Abdur Chowdhury, and Aleksandr Kolcz. Automatic web query classification using labeled and unlabeled training data. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 581–582, New York, NY, USA, 2005. ACM Press.
3. Steven M. Beitzel, Eric C. Jensen, David D. Lewis, Abdur Chowdhury, and Ophir Frieder. Automatic classification of web queries using very large unlabeled query logs. *ACM Trans. Inf. Syst.*, 25(2):9, 2007.
4. Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Measuring semantic similarity between words using web search engines. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 757–766, New York, NY, USA, 2007. ACM Press.
5. Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.
6. Seokkyung Chung and Dennis McLeod. Dynamic topic mining from news stream data. In *CoopIS/DOA/ODBASE*, pages 653–670, 2003.

7. Susan T. Dumais and Hao Chen. Hierarchical classification of Web content. In Nicholas J. Belkin, Peter Ingwersen, and Mun-Kew Leong, editors, *Proc. of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval*, pages 256–263, Athens, GR, 2000. ACM Press, New York, US.
8. Google. Google search appliance frequently asked questions, accessed 2006. Available online at http://www.google.com/enterprise/gsa/features.html.
9. Yukiko Kawai, Tadahiko Kumamoto, and Katsumi Tanaka. User preference modeling based on interest and impressions for news portal site systems. In *DEXA*, pages 549–559, 2006.
10. Bill Kules, Jack Kustanowitz, and Ben Shneiderman. Categorizing web search results into meaningful and stable categories using fast-feature techniques. In *JCDL*, pages 210–219, 2006.
11. Kristina Lerman. Social information processing in social news aggregation. *ArXiv Computer Science e-prints*, March 2007.
12. Yuefeng Li and Ning Zhong. Rough association rule mining in text documents for acquiring web user information needs. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 226–232, Washington, DC, USA, 2006. IEEE Computer Society.
13. P. Lingras. Rough set clustering for web mining. In *Proc. of 2002 World Congress on Computational Intelligence, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE(02) Special Session on Computational Web Intelligence (CWI), Honolulu, Hawaii, USA 2002*.
14. Pawan Lingras and Chad West. Interval set clustering of web users with rough k-means. *J. Intell. Inf. Syst.*, 23(1):5–16, 2004.
15. S. Mitra, S. Pal, and P. Mitra. Data mining in soft computing framework: A survey. *IEEE Transactions on Neural Networks*, 2001.
16. Chi Lang Ngo and Hung Son Nguyen. A method of web search result clustering based on rough sets. *wi*, 0:673–679, 2005.
17. Stanislaw Osinski, Jerzy Stefanowski, and Dawid Weiss. Lingo: Search results clustering algorithm based on singular value decomposition. In *Intelligent Information Systems*, pages 359–368, 2004.
18. Stanislaw Osinski and Dawid Weiss. A concept-driven algorithm for clustering search results. *IEEE Intelligent Systems*, 20(3):48–54, 2005.
19. Sankar Pal, Varun Talwar, and Pabitra Mitra. Web mining in soft computing framework: Relevance, state of the art and future directions, 2002.
20. Z. Pawlak. Rough sets present state and further prospects, 1994.
21. Zdzislaw Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Norwell, MA, USA, 1992.
22. Dou Shen, Rong Pan, Jian-Tao Sun, Jeffrey Junfeng Pan, Kangheng Wu, Jie Yin, and Qiang Yang. Query enrichment for web-query classification. *ACM Trans. Inf. Syst.*, 24(3):320–352, 2006.
23. Andrzej Skowron and Piotr Synak. Planning based on reasoning about information changes. In *RSCTC*, pages 165–173, 2006.
24. Isak Taksa, Sarah Zelikovitz, and Amanda Spink. Using web search logs to identify query classification terms. In *ITNG '07: Proceedings of the International Conference on Information Technology*, pages 469–474, Washington, DC, USA, 2007. IEEE Computer Society.
25. Tiffany Ya Tang and Gordon I. McCalla. Student modeling for a web-based learning environment: A data mining approach. In *AAAI/IAAI*, pages 967–968, 2002.
26. H. Wettig, J. Lahtinen, T. Lepola, P. Myllymaki, and H. Tirri. Bayesian analysis of online newspaper log data, 2003.

A Hierarchy Concept in Modeling of Concurrent Systems Described by Information Systems

Krzysztof Pancerz [1,2], Zofia Matusiewicz [3]

[1] Chair of Computer Science Foundations
University of Information Technology and Management
Sucharskiego Str. 2, 35-225 Rzeszów, Poland
kpancerz@wsiz.rzeszow.pl
[2] Chair of Computer Science and Knowledge Engineering
College of Management and Public Administration
Akademicka Str. 4, 22-400 Zamość, Poland
[3] Chair of Mathematics
University of Information Technology and Management
Sucharskiego Str. 2, 35-225 Rzeszów, Poland
zmatusiewicz@wsiz.rzeszow.pl

Abstract. The paper provides a brief outline of the methodology for building suitable hierarchical models of concurrent systems described by information systems. The models have the form of hierarchical colored Petri nets. An introduction of a hierarchy concept in modeling of large systems seems to be necessary for simplification of legibility of the obtained models. Therefore, the main purpose of the hierarchy constructs is to break down the complexity of the large nets by dividing them into a number of subnets. In the proposed approach, the starting point for building a hierarchical net is an information system describing a given concurrent system. The hierarchy construct starts as early as on the level of description by building the so-called generalized information system. Such a system arises from combining selected real processes of a concurrent system into some generalized processes.

Keywords: rough sets, information systems, colored Petri nets, hierarchical models.

1 Introduction

The idea of describing concurrent systems using information systems has been initiated by Z. Pawlak in [8]. In this approach, an information system represented by a data table includes the knowledge of global states of a given concurrent system CS. The columns of a table are labeled with names of attributes (treated as the processes of CS). Each row labeled with an object (treated as a global state of CS) includes a record of attribute values (treated as local states of processes). The next step in development of that idea was modeling concurrent systems described by information systems using different kinds of Petri nets. Until now,

concurrent systems described by information systems have been modeled using non-hierarchical place transition nets (see [10]) as well as non-hierarchical colored Petri nets (see [4], [5], [6], [12]). The use of colored Petri nets enables us to obtain succinct and coherent models. In this case, the complexity of models is divided among net structures, declarations and net inscriptions. However, large data tables cause the models to become more complicated. Especially, their declarations and inscriptions are less readable because of their sizes. A natural way to solve this problem is building hierarchical models. A given hierarchical model can be constructed by suitable combination of a number of smaller models. Looking at the history, an analogous approach has been used in constructing the large software systems. The existence of subroutines and modules in all modern programming languages makes building such systems possible. Moreover, the legibility of their codes is greater. Therefore, colored Petri nets have also a hierarchical version. Hierarchical colored Petri nets have been proposed in [1]. One can also find there five different hierarchy constructs. In this paper we make the first attempt to model concurrent systems described by information systems using some kind of hierarchical colored Petri nets different from those proposed in [1]. In order to do it, first, we propose to build the so-called generalized information systems describing concurrent systems on the more abstract level by combining specific (real) processes of concurrent systems into some generalized processes. On the basis of generalized information systems, the net models on the first level are built. However, subnets on the second level represent just real processes. Constructing single subnets on the individual levels is based on algorithms given in [4], [6].

2 Preliminaries

First, we recall basic concepts concerning rough set theory [7] used in the paper as well as give some informal description of colored Petri nets. For a formal description of them we refer the reader to [2].

2.1 Information Systems, Reducts and Rules

In the rough set theory, information systems are used to represent some knowledge of elements of the universe of discourse. An *information system* is a pair $S = (U, A)$, where U is a nonempty, finite set of objects, called the universe, A is a nonempty, finite set of attributes, i.e., $a : U \to V_a$ for $a \in A$, where V_a is called the value set of a. Any information system can be represented as a data table whose columns are labeled with attributes, rows are labeled with objects, and entries of the table are attribute values. Each minimal (with respect to inclusion) subset $B \subseteq A$ which preserves discernibility of objects determined by the set A of all attributes in the information system $S = (U, A)$ is called a *reduct* of S. The set of all reducts in a given information system S will be denoted as $Red(S)$. For a given reduct $R \in Red(S)$, there exists the functional dependency between values of attributes from R and values of attributes from $A - R$ marked

with $R \Rightarrow A - R$, i.e., the values of attributes from R uniquely determine all values of attributes from $A - R$. Especially, we have that $R \Rightarrow \{a\}$ for each $a \in A - R$. Atomic formulas over A and $V = \bigcup_{a \in A} V_a$ are expressions of the form (a, v), where $a \in A$ and $v \in V_a$. In this paper, a *rule* r in the information system S is a formula of the form $(a_{i_1}, v_{i_1}) \wedge (a_{i_2}, v_{i_2}) \wedge \ldots \wedge (a_{i_l}, v_{i_l}) \Rightarrow (a_d, v_d)$, where $a_d \in A$, $v_d \in V_{a_d}$, $a_{i_j} \in A - \{a_d\}$, and $v_{i_j} \in V_{a_{i_j}}$ for $j = 1, \ldots, l$. A formula on the left side of $\Rightarrow$ is called a predecessor of a rule, whereas a formula on the right side of $\Rightarrow$ is called a successor of a rule. A rule r is *true* in the information system $S = (U, A)$ if and only if for each object $u \in U$ we have that if $a_{i_1}(u) = v_{i_1}$ and $a_{i_2}(u) = v_{i_2}$ and ... and $a_{i_l}(u) = v_{i_l}$, then $a_d(u) = v_d$. A rule is called *minimal* in S if and only if the removal of any atomic formula from the predecessor of a rule causes that a rule is not true in S. The set of all minimal rules true in S will be denoted as $Rul(S)$.

2.2 Subsystems

Let $S = (U, A)$ be an information system. An information system $S' = (U', A')$ such that $U' \subseteq U$, $A' = \{a' : a \in B \subseteq A\}$, $a'(u) = a(u)$ for $u \in U'$ and $V_{a'} = V_a$ for $a \in A$ is called a *subsystem* of S. The set A' will be identified with a corresponding subset B of A. It is easy to see that each reduct $R \in Red(S)$ determines a subsystem $S_R = (U_R, R)$ of S. If we have a family $Sub(S) = \{S_1 = (U_1, A_1), S_2 = (U_2, A_2), \ldots S_q = (U_q, A_q)\}$ of subsystems of S, where $U_1 = U_2 = \ldots = U_q = U$ and $A_i \cap A_j = \emptyset$ for any $i, j = 1, 2, \ldots, q$ and $i \neq j$, then subsystems from $Sub(S)$ are called *attribute-separable subsystems*.

2.3 Colored Petri Nets

In general, Petri nets are a graphical and mathematical tool for modeling processes acting concurrently. The states of colored Petri nets are represented by places (drawn as circles) whereas the actions are represented by transitions (drawn as rectangles). The places are connected with transitions by arcs (and also inversely, transitions with places). For each place p, a nonempty set of tokens called a *color set* is associated with p. In a given place p, only tokens belonging to the color set associated with p can appear. A marking of the net determines distribution of tokens on all places. A marking of a single place p is a multiset over the color set associated with p. Expressions on arcs determine which tokens are removed from input places and which tokens are placed in output places of transitions connected with arcs. For each transition t, a boolean expression called a guard expression is associated with t. If this expression is fulfilled, then a transition can be fired. Firing a given transition t causes removing suitable tokens from input places of t and placing suitable tokens in output places of t. So, transitions are responsible for state (marking) changes. Arc expressions and guard expressions may include, among others, variables which are replaced with suitable values during firing transitions.

3 First Step to Modeling Concurrent Systems Described by Information Systems Using Hierarchical Colored Petri Nets

In this section, we give an introduction to modeling concurrent systems described by information systems using hierarchical colored Petri nets. In order to build a hierarchical net model $HCPN(S)$ of a given concurrent system described by the information system $S = (U, A)$ we may perform Algorithm 1. In the presented approach, the information system $S = (U, A)$ describing a given concurrent system must satisfy the following requirement: S has at least one reduct R such that $R \subset A$. Otherwise, we can create only subsystems of S representing single processes and what follows we cannot combine any processes together to obtain some generalized processes. The exponential time complexity of Algorithm 1 is the main inconvenience of the proposed approach.

Algorithm 1 for building a hierarchical model of a given concurrent system described by an information system.

Input: An information system $S = (U, A)$ such that $\underset{R \in Red(S)}{\exists} R \subset A$.

Output: A hierarchical net model $HCPN(S)$ of a given concurrent system described by the information system S.

Start

Step 1: Compute a set $Red(S)$ of all reducts of S.

Step 2: Split the information system S into attribute-separable subsystems with respect to reducts of S as follows.

- **Step 2.1:** Create empty families $\mathcal{S}_{Red}$ and $\mathcal{S}_{Attr}$ of subsystems.
- **Step 2.2:** From the set $Red(S)$ of all reducts of S choose a maximal (with respect to the number of elements) subset $Red^*(S) = \{R_1, R_2, \ldots, R_k\}$ of pairwise disjoint reducts (i.e., $R_i \cap R_j = \emptyset$ for any $i, j = 1, 2, \ldots, k$ and $i \neq j$) of S.
- **Step 2.3:** For each reduct $R \in Red^*(S)$ do:
  - **Step 2.3.1:** Create a subsystem $S_R = (U, R)$ of S.
  - **Step 2.3.2:** Add S_R to the family $\mathcal{S}_{Red}$.
- **Step 2.4:** For each attribute $a \in A - (R_1 \cup R_2 \cup \ldots \cup R_k)$ do:
  - **Step 2.4.1:** Create a subsystem $S_a = (U, \{a\})$ of S.
  - **Step 2.4.2:** Add S_a to the family $\mathcal{S}_{Attr}$.
- **Step 2.5:** Create a family $Sub(S)$ of all attribute-separable subsystems of S by adding all subsystems from $\mathcal{S}_{Red}$ and all subsystems from $\mathcal{S}_{Attr}$ to $Sub(S)$.

Step 3: Build a generalized information system $\mathbf{S} = (\mathbf{U}, \mathbf{A})$ for S as follows. Each attribute $\mathbf{a} \in \mathbf{A}$ corresponds exactly to one subsystem $S' = (U', A')$ from $Sub(S)$. Each object $\mathbf{u} \in \mathbf{U}$ corresponds exactly to one object u from the original information system S. For each object $\mathbf{u} \in \mathbf{U}$ and each attribute $\mathbf{a} \in \mathbf{A}$:

- if the attribute **a** corresponds to a subsystem S_R from $\mathcal{S}_{Red}$, then $\mathbf{a}(\mathbf{u})$ is a tuple of the form $(a_{i_1}(u), a_{i_2}(u), \ldots, a_{i_k}(u))$, where $a_{i_1}, a_{i_2}, \ldots, a_{i_k} \in R$,
- if the attribute **a** corresponds to a subsystem S_a from $\mathcal{S}_{Attr}$, then $\mathbf{a}(\mathbf{u})$ has the form $a(u)$.

Obviously, each tuple $(a_{i_1}(u), a_{i_2}(u), \ldots, a_{i_k}(u))$ can be additionally encoded by a single value for simplification of the description.

Step 4: For each subsystem $S' = (U', A')$ obtained in Step 2, where $card(A) > 1$, compute a set $Rul(S')$ of all minimal rules true in S'.

Step 5: Compute a set $Rul(\mathbf{S})$ of all minimal rules true in **S**.

Step 6: For each subsystem $S' = (U', A')$ obtained in Step 2, build a submodel $CPN(S')$ in the form of a colored Petri net. A guard expression for a transition of $CPN(S')$ is constructed on the basis of the set $Rul(S')$ of all minimal rules true in S'. Note that for each subsystem $S' = (U', A')$, where $card(A') = 1$, we have that $Rul(S') = \emptyset$.

Step 7: For the generalized information system **S**, build a model $CPN(\mathbf{S})$ in the form of a colored Petri net. A guard expression for a transition of $CPN(\mathbf{S})$ is constructed on the basis of the set $Rul(\mathbf{S})$ of all minimal rules true in **S**.

Step 8: Create a hierarchical net model $HCPN(S)$ on the basis of net models $CPN(\mathbf{S})$ and $\{CPN(S_i)\}_{S_i \in Sub(S)}$.

Step 9: Return a hierarchical net model $HCPN(S)$.

Stop

4 An Illustration

In this section, we depict an example explaining the presented algorithm.

We now apply the proposed methodology for building a hierarchical model of a concurrent system of financial processes like exchange rates between the Polish zloty and important currencies like the US dollar (marked with *usd*), the euro (marked with *eur*), the Japanese yen (marked with *jpy*), the Canadian dollar (marked with *cad*), the Swiss franc (marked with *chf*), and the Pound sterling (marked with *gbp*). The behavior of a financial system observed by us is described in a data table representing an information system S (see Table 1). Attributes correspond to currencies, whereas objects correspond to consecutive business days. The meaning of attribute values is the following: -1 denotes decreasing a given exchange rate in relation to the previous exchange rate, 0 denotes remaining a given exchange rate on the same level in relation to the previous exchange rate, 1 denotes increasing a given exchange rate in relation to the previous exchange rate. Formally, for the information system S, we have: the set of objects $U = \{u_1, u_2, \ldots, u_7\}$, the set of attributes $A = \{usd, eur, jpy, cad, chf, gbp\}$, the sets of attribute values $V_{usd} = V_{eur} = V_{jpy} = V_{cad} = V_{gbp} = \{-1, 0, 1\}$ and $V_{chf} = \{-1, 0\}$.

We obtain the following set of reducts of S: $Red(S) = \{R_1, R_2, R_3, R_4\}$, where $R_1 = \{usd, cad\}$, $R_2 = \{jpy, cad\}$, $R_3 = \{eur, jpy, gbp\}$, and $R_4 = \{usd, eur, gbp\}$. For the further computation, we can choose the disjoint reducts R_1 and R_3. Choosing these reducts we have three attribute-separable subsystems $S_{R_1} = (U, R_1)$, $S_{R_3} = (U, R_3)$ and $S_{chf} = (U, \{chf\})$ shown in Table 2.

Table 1. An information system S describing a financial system

U/A	usd	eur	jpy	cad	chf	gbp
u_1	-1	-1	-1	-1	-1	-1
u_2	-1	0	-1	0	0	-1
u_3	1	0	1	1	0	1
u_4	0	0	0	1	0	0
u_5	0	1	0	0	0	0
u_6	0	0	0	1	0	0
u_7	1	0	1	0	0	-1

For each subsystem, there are also presented encoded values of attributes for the generalized information system.

Table 2. Subsystems S_{R_1}, S_{R_3} and S_{chf}, respectively.

U/A	usd	cad	$curr_1$
u_1	-1	-1	**1**
u_2	-1	0	**2**
u_3	1	1	**3**
u_4	0	1	**4**
u_5	0	0	**5**
u_6	0	1	**4**
u_7	1	0	**6**

U/A	eur	jpy	gbp	$curr_2$
u_1	-1	-1	-1	**1**
u_2	0	-1	-1	**2**
u_3	0	1	1	**3**
u_4	0	0	0	**4**
u_5	1	0	0	**5**
u_6	0	0	0	**4**
u_7	0	1	-1	**6**

U/A	chf	$curr_3$
u_1	-1	$\mathbf{-1}$
u_2	0	**0**
u_3	0	**0**
u_4	0	**0**
u_5	0	**0**
u_6	0	**0**
u_7	0	**0**

A generalized information system **S** is shown in Table 3. The attribute $curr_1$ corresponds to a generalized process representing two real processes (currencies) usd and cad, whereas the attribute $curr_2$ corresponds to a generalized process representing three real processes (currencies) eur, jpy and gbp.

Table 3. A generalized information system **S** for S

U/A	$curr_1$	$curr_2$	$curr_3$
u_1	**1**	**1**	$\mathbf{-1}$
u_2	**2**	**2**	**0**
u_3	**3**	**3**	**0**
u_4	**4**	**4**	**0**
u_5	**5**	**5**	**0**
u_6	**4**	**4**	**0**
u_7	**6**	**6**	**0**

Minimal rules true in the subsystems have a standard form. For the subsystem S_{R_1}, we obtain only one rule: $(cad, -1) \Rightarrow (usd, -1)$. For the subsystem

S_{R_3}, we obtain ten minimal rules true in S_{R_3}. For example, we have the following rules: $(gbp, 0) \Rightarrow (jpy, 0)$ and $(jpy, 1) \Rightarrow (eur, 0)$. Obviously, there is the lack of rules within the subsystem S_{chf}. On the first level (i.e., for the information system $\mathbf{S}$), we have, for example, the following rules: $(curr_1, \mathbf{1}) \Rightarrow (curr_2, \mathbf{1})$, $(curr_3, -\mathbf{1}) \Rightarrow (curr_2, \mathbf{1})$.

An outline of the hierarchical net model $HCPN(S)$ is shown in Figure 1. It is worth noting that there exist some connections (symbolically shown in the net) representing generalization between the net on the first level and the nets on the second level. Each place of the net $CPN(\mathbf{S})$ represents places of one of the subnets. For example, the place p_{curr1} of $CPN(\mathbf{S})$ is a generalized place for places p_{usd} and p_{cad}. The foregoing implies that markings of places of $CPN(\mathbf{S})$ and markings of places in subnets are in the strict relationship. It is also significant throughout firing transitions on the first level and the second level. Each place (drawn as a circle) in the net on the first level represents one generalized process. A transition (drawn as a rectangle) represents the change of states of generalized processes. Each place in nets on the second level represents a real process. Each transition of these nets represents the change of states of processes belonging to a given subsystem. In Figure 1, some significant elements of a net description have been deliberately omitted (such as global declarations of color sets and variables, arc expressions, guard expressions for transitions, types and markings of places). First of all, markings in places of nets represent states of processes of the modeled system. For more detailed descriptions of net models of concurrent systems described by information systems we refer readers to [4], [5].

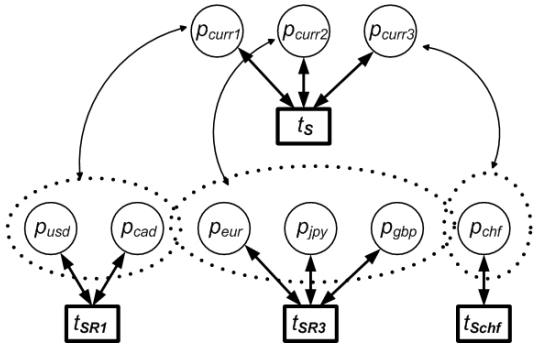

Fig. 1. An outline of the hierarchical net model $HCPN(S)$ for the information system S

The main property of the created net models of concurrent systems described by information systems (also these hierarchical) is as follows. Such models generate all global states of modeled systems which are consistent with all rules extracted from information systems and used to build models (see [4], [10], [12]). In the approach presented in this paper, on the first level, a hierarchical net model $HCPN(S)$ for a given information system S enables us to obtain a maximal consistent extension [11], [13] of a generalized information system $\mathbf{S}$ created for S. Individual models on the second level enable us to obtain maximal consistent extensions of appropriate subsystems of S. It is worth mentioning that models can generate not only global states observed earlier (and collected in information systems), but also some new global states which have not been observed yet. They are consistent with all extracted rules. Such global states can deliver new knowledge about modeled systems.

It is possible to add other levels to the hierarchical net models. We can consider subsystems representing the so-called components of information systems (see [4], [9]) and also single processes. In that case, for a given information system S, we have the four level hierarchical net model. Then, the first level consists of the net built for a generalized information system $\mathbf{S}$ of S, the second level consists of the nets built for subsystems created by taking into consideration reducts of S, the third level consists of the nets built for subsystems created by taking into consideration components [9], [10] of S, and finally, the fourth level consists of the nets built for single processes (attributes) of S.

5 Conclusions

The paper gives a rather informal description of the idea of building hierarchical models (in the form of colored Petri nets) of concurrent systems described by information systems. It is, however, significant to give a formal description of such models especially from the point of view of Petri nets, and that will be done in the future papers. Moreover, the approach to the hierarchy constructs presented in the paper is one of the possible approaches. This paper constitutes a continuation of research on discovering concurrent models from data tables started by Z. Pawlak in [7].

Acknowledgments

This paper has been partially supported by the grant from the University of Information Technology and Management in Rzeszów, Poland.

References

1. Huber, P., Jensen, K., Shapiro, R.M.: Hierarchies in Coloured Petri Nets. In: G. Rozenberg (ed.), *Advances in Petri Nets 1990*, LNCS **483**, Springer-Verlag, Berlin Heidelberg (1991) 313-341.

2. Jensen, K.: *Coloured Petri Nets. Vol. 1: Basic Concepts.* Springer-Verlag, Berlin Heidelberg (1997).
3. Jensen, K., Rozenberg, G. (eds.): *High-level Petri Nets. Theory and Application.* Springer-Verlag (1991).
4. Pancerz, K., Suraj, Z.: Synthesis of Petri Net Models: A Rough Set Approach. *Fundamenta Informaticae* **55**(2), IOS Press, Amsterdam (2003) 149-165.
5. Pancerz, K., Suraj, Z.: Discovering Concurrent Models from Data Tables with the ROSECON System. *Fundamenta Informaticae* **60**(1-4), IOS Press, Amsterdam (2004) 251-268.
6. Pancerz, K., Suraj, Z.: Automated Discovering Concurrent Models from Data Tables - an Overview of Algorithms. In: *Proc. of the AICCSA'2005*, Cairo, Egypt, IEEE Computer Society (2005) [on CD].
7. Pawlak, Z.: *Rough Sets - Theoretical Aspects of Reasoning About Data.* Kluwer Academic Publishers, Dordrecht (1991).
8. Pawlak, Z.: Concurrent Versus Sequential the Rough Sets Perspective. *Bulletin of the EATCS* **48** (1992) 178-190.
9. Suraj, Z.: Discovery of Concurrent Data Models from Experimental Tables: A Rough Set Approach. *Fundamenta Informaticae* **28**(3-4), IOS Press, Amsterdam (1996) 353-376.
10. Suraj, Z.: Rough Set Methods for the Synthesis and Analysis of Concurrent Processes. In: L. Polkowski, S. Tsumoto, T.Y. Lin (eds.), *Rough Set Methods and Applications*, Physica-Verlag, Berlin (2000) 379-488.
11. Suraj, Z.: Some Remarks on Extensions and Restrictions of Information Systems. In: W. Ziarko, Y. Yao (eds.), *Rough Sets and Current Trends in Computing*, LNAI **2005**, Springer, Berlin (2001) 204-211.
12. Suraj, Z., Pancerz, K.: A Synthesis of Concurrent Systems: A Rough Set Approach. In: G. Wang et al., *Proc. of the RSFDGrC'2003*, Chongqing, China, LNAI **2639**, Springer-Verlag, Berlin Heidelberg (2003) 299-302.
13. Suraj, Z., Pancerz, K., Owsiany, G.: On Consistent and Partially Consistent Extensions of Information Systems. In: D. Ślęzak, G. Wang, M. Szczuka, I. Düntsch, Y. Yao (eds.), *Proc. of the RSFDGrC'2005, Part I*, Regina, Canada, LNAI **3641**, Springer-Verlag, Berlin Heidelberg (2005) 224-233.

Towards Granular Computing: Classifiers Induced From Granular Structures

Lech Polkowski[1,2] and Piotr Artiemjew[2]

[1]Polish–Japanese Institute of Information Technology
Koszykowa 86, 02008 Warszawa, Poland
[2]Department of Mathematics and Computer Science
University of Warmia and Mazury, Olsztyn, Poland
polkow@pjwstk.edu.pl;artem@matman.uwm.edu.pl

Abstract. Granular computing as a paradigm is an area frequently studied within the Approximate Reasoning paradigm. Proposed by L.A. Zadeh granular computing has been studied within fuzzy as well as rough set approaches to uncertainty. It is manifest that both theories are immanently related to granulation as fuzzy set theory begins with fuzzy membership functions whose inverse images are prototype granules whereas rough set theory starts with indiscernibility relations whose classes are prototype, or, elementary granules.

Many authors have devoted their works to analysis of granulation of knowledge, definitions of granules, methods for combining (fusing) granules into larger objects, applications of granular structures, see, quoted in references works by A. Skowron, T.Y. Lin, Y.Y. Yao, L. Polkowski and others.

In this work, the emphasis is laid on granular decision (data) systems: they are introduced, methods of their construction with examples are pointed to, and applications are exhibited; those applications are founded on the basic although often implicit principle of data mining, viz., once a plausible for given data similarity measure is found, objects satisfactorily similar should reveal sufficiently close (or, for that matter identical) class values.

In this work, this principle is applied to granules, following the idea presented by L. Polkowski at 2005, 2006 IEEE GrC conferences, that granules built on basis of a similarity relation from a given decision system should consists of objects similar to such a degree that averaging them would lead to new objects which together would constitute a new decision system preserving to a high degree knowledge represented by the original decision system. As knowledge in rough set theory is meant as the classification ability, it seems reasonable to test knowledge content with classifiers as classifier accuracy.

This informal idea is tested in this work with some specific tools for granule construction, granular system building, and some well–tested classifiers known in literature for a few data sets from the UCI repository.

In the following sections we outline: basic ideas of rough computing, granulation of knowledge, the idea of a granular decision system and we include the results of exemplary tests with real data.

Keywords rough sets, granular data sets, rough inclusions, classification accuracy

1 Basic rough computing

In this section, we outline basic facts of rough set theory, see [12], [11], necessary for a further development of ideas. Classical ideas about representations of uncertainty, expressed respectively by Gottlob Frege and Max Black, found realization respectively in rough and fuzzy concept theories. Despite their formal differences and distinct starting points, both compute with granules of objects: rough sets with indiscernibility classes of objects, fuzzy sets with inverse images of fuzzy membership functions.

Rough set theory understands knowledge as classification and thus it does represent knowledge by means of *information systems*, i.e., pairs of the form (U, A) where U is a set of *objects* and A is a set of *attributes* with each $a \in A$ a mapping $a : U \to V_a$ on U into the value set V_a. Objects are coded by their *information sets* of the form

$$inf(u) = \{(a = a(u)) : a \in A\}.$$

Objects u, v with $inf(u) = inf(v)$ are called *indiscernible* and they are regarded as identical with respect to the given set A. The *B–indiscernibility relation relative to a set $B \subseteq A$* is

$$ind(B) = \{(u, v) : \forall a \in B.a(u) = a(v)\}.$$

Classes $[u]_B = \{v : (u, v) \in ind(B)\}$ are *B–elementary granules* of knowledge. Their unions are *B–granules* of knowledge.

A subset $B \subseteq A$ of the attribute set is a *reduct* of an information system (U, A) iff B is a minimal with respect to inclusion subset of A such that $ind(B) = ind(A)$; reducts and their extensions: relative reducts, play an important role in rule induction, see [34].

The language of descriptor logic is most often in use when logical aspects of rough sets are in focus. A formula $(a = v)$ is an *elementary descriptor*; *descriptors* are formed as the smallest set containing all elementary descriptors and closed under sentential connectives $\vee, \wedge, \neg, \Rightarrow$. The meaning $[a = v]$ of an elementary descriptor is defined as the set $\{u : a(u) = v\}$ and it is recursively extended to meaning of descriptors, i.e., $[\alpha \vee \beta] = [\alpha] \cup [\beta]$, $[\alpha] \wedge [\beta] = [\alpha] \cap [\beta]$, $[\neg\alpha] = U \setminus [\alpha]$.

Another form of prototype granules is provided by *templates*, see [30], i.e., generalized descriptors of the form $(a \in W_a)$ where $W_a \subseteq V_a$.

Decision systems are information systems of the form $(U, A \cup \{d\})$ with a singled out attribute d called the *decision* that does represent a description of objects by an external informed source (say, an expert). *Decision rules* of are formulas of the form

$$\bigwedge_{a \in B}(a = v_a) \Rightarrow (d = v).$$

The rule is *true, certain* whenever $[\bigwedge_{a \in B}(a = v_a)] \subseteq [d = v]$; otherwise, it is *possible*.

2 Classification

Classification methods can be divided according to the adopted methodology, into classifiers based on reducts and decision rules, classifiers based on templates and similarity, classifiers based on descriptor search, classifiers based on granular descriptors, hybrid classifiers.

For a decision system (U, A, d), classifiers are sets of decision rules. Induction of rules was a subject of research in rough set theory since its beginning. In most general terms, building a classifier consists in searching in the pool of descriptors for their conjuncts that describe sufficiently well decision classes. As distinguished in [36], there are three main kinds of classifiers searched for: *minimal*, i.e., consisting of minimum possible number of rules describing decision classes in the universe, *exhaustive*, i.e., consisting of all possible rules, *satisfactory*, i.e., containing rules tailored to a specific use. Classifiers are evaluated globally with respect to their ability to properly classify objects, usually by *error* which is the ratio of the number of correctly classified objects to the number of test objects, *total accuracy* being the ratio of the number of correctly classified cases to the number of recognized cases, and *total coverage*, i.e, the ratio of the number of recognized test cases to the number of test cases.

Minimum size algorithms include LEM2 algorithm due to Grzymala–Busse [7] and covering algorithm in RSES package [33]; exhaustive algorithms include, e.g., LERS system due to Grzymala–Busse [5], systems based on discernibility matrices and Boolean reasoning [32], see also [2], [3], implemented in the RSES package [33].

Minimal consistent sets of rules were introduced in Skowron and Rauszer [34]. Further developments include dynamic rules, approximate rules, and relevant rules as described in [2], [3], as well as local rules (op. cit.) effective in implementations of algorithms based on minimal consistent sets of rules. Rough set based classification algorithms, especially those implemented in the RSES system [33], were discussed extensively in [3].

In [2], a number of techniques were verified in experiments with real data, based on various strategies:

`discretization` of attributes (codes:N-no discretization, S-standard discretization, D-cut selection by dynamic reducts, G-cut selection by generalized dynamic reducts);

`dynamic selection of attributes` (codes: N-no selection, D-selection by dynamic reducts, G-selection based on generalized dynamic reducts);

`decision rule choice` (codes: A-optimal decision rules, G-decision rules on basis of approximate reducts computed by Johnson's algorithm, simulated annealing and Boltzmann machines etc., N-without computing of decision rules);

`approximation of decision rules` (codes: N-consistent decision rules, P-approximate rules obtained by descriptor dropping);

`negotiations among rules` (codes:S-based on strength, M-based on maximal strength,R-based on global strength, D-based on stability).

Any choice of a strategy in particular areas yields a compound strategy denoted with the alias being concatenation of symbols of strategies chosen in consecutive areas, e.g., NNAND etc.

We record here in Table 2 an excerpt from the comparison (Table 8, 9, 10 in [2]) of best of these strategies with results based on other paradigms in classification for two sets of data: Diabetes and Australian credit from UCI Repository [37].

Table 1. A comparison of errors in classification by rough set and other paradigms

paradigm	*system/method*	*Diabetes*	*Austr.credit*
Stat.Methods	*Logdisc*	0.223	0.141
Stat.Methods	*SMART*	0.232	0.158
Neural Nets	*Backpropagation2*	0.248	0.154
Neural Networks	*RBF*	0.243	0.145
Decision Trees	*CART*	0.255	0.145
Decision Trees	*C4.5*	0.270	0.155
Decision Trees	*ITrule*	0.245	0.137
Decision Rules	*CN2*	0.289	0.204
Rough Sets	*NNANR*	0.335	0.140
Rough Sets	*DNANR*	0.280	0.165
Rough Sets	*best result*	0.255($DNAPM$)	0.130($SNAPM$)

3 Similarity

Similarity relations include metrics as prototype similarity measures, see, e.g., [4]. Starting from a metric, one can build a similarity measure in the way pointed by Henri Poincare, see *Science et Hypothesé*, (Paris,1905): He considered a relation $\tau_d(x, y)$ which holds iff $d(x, y) < r$ for some fixed r and a metric d. τ_d is a *tolerance relation*, i.e., it is reflexive and symmetric.

We generalize this idea: we let $\mu_d(x, y, r)$ iff $d(x, y) < 1-r$; then, the predicate μ_d does satisfy a number of conditions:

1. $\mu_d(x, y, 1)$ iff $x = y$;
2. if $\mu_d(x, y, 1)$ and $\mu_d(z, x, r)$ then $\mu_d(z, y, r)$;
3. if $\mu_d(x, y, r)$ and $s < r$ then $\mu_d(x, y, s)$.
4. if $\mu_d(x, y, r)$ and $\mu_d(y, z, s)$ then $\mu_d(x, z, L(r, s))$, where $L(r, s) = max\{0, r + s - 1\}$ is the well–known Łukasiewicz functor of many–valued logics, known also as the Łukasiewicz t–norm or tensor product (see [8],[13]).

Properties 1–3 are singled out by us as characteristic for *rough inclusions*; property 4. which does reflect the triangle inequality for the metric d, is the *transitivity property* of rough inclusions of the form μ_d. These properties will be established as well for rough inclusions defined in the sequel.

An abstract generalization of the similarity μ_d is a similarity measure called a rough inclusion; its definition will refer to properties 1–3 with an additional important factor, viz., property 1 will be in general referred to an *ingredient relation* of mereology, see, e.g., [16, 17].

A *rough inclusion* $\mu_\pi(x, y, r)$, where x, y are individual objects, $r \in [0, 1]$, and π is a part relation of a chosen mereology of concepts, does satisfy the following requirements, relative to a given part relation π on a set U of individual objects,

$$\begin{array}{c} 1.\ \mu_\pi(x, y, 1) \Leftrightarrow x\ ing_\pi\ y; \\ 2.\ \mu_\pi(x, y, 1) \Rightarrow [\mu_\pi(z, x, r) \Rightarrow \mu_\pi(z, y, r)]; \\ 3.\ \mu_\pi(x, y, r) \wedge s < r \Rightarrow \mu_\pi(x, y, s). \end{array} \quad (1)$$

Those requirements seem to be intuitively clear: 1. demands that the predicate μ_π is an extension to the relation ing_π of the underlying system of Mereology; 2. does express monotonicity of μ_π, and 3. assures the reading: "to degree at least r".

3.1 The rough inclusion induced by Łukasiewicz's t–norm

In this work, as a similarity measure on objects of an information system, we will apply the rough inclusion of the form,

$$\mu_L(u, v, r) \text{ iff } \frac{|IND(u,v)|}{|A|} \geq r,$$

which is clearly induced by the reduced modulo $|A|$ Hamming metric: $h(u, v) = \frac{|DIS(u,v)|}{|A|}$ where $DIS(u, v) = U \times U \setminus IND(u, v)$, i.e, $\mu_L(u, v, r)$ iff $h(u, v) \leq 1 - r$.

The relation between the rough inclusion μ_L and the Łukasiewicz t–norm $L(x, y) = max\{0, x + y - 1\}$ is deeper as shown in, e.g., [16, **?**], viz., starting from the representation $L(x, y) = g(f(x) + f(y))$, see, e.g., [13], the rough inclusion μ_L does satisfy the relation $\mu_L(u, v, r)$ iff $g(\frac{|DIS(u,v)|}{|A|}) \geq r$, see, e.g., [16, 17].

Clearly, as a particular case of 3, p.4, the rough inclusion μ_L does satisfy the transitivity property in the form of,

$$\text{if } \mu_L(u, v, r) \text{ and } \mu_L(v, w, s) \text{ then } \mu_L(u, w, L(r, s)). \quad (2)$$

4 Granulation of knowledge

The issue of granulation of knowledge as a problem on its own, has been posed by L.A.Zadeh [43], [42]. The issue of granulation has been a subject of intensive studies within rough set community, as witnessed by a number of papers, e.g., TY Lin [9], [10], Polkowski [25], [23],[24], Qing Liu [29], Skowron [35], YY Yao [39], [40], [41].

Rough set context offers a natural venue for granulation, and indiscernibility classes were recognized as *elementary granules* whereas their unions serve as *granules of knowledge*; these granules and their direct generalizations to various similarity classes were subject to a research, see, e.g., TY Lin [9], [10], YY Yao [39],[40], [41]. Granulation of knowledge by means of rough inclusions was studied in Polkowski [25, 22], [23],[24], [14],[15], [16], [17].

Granulation of knowledge and applications to knowledge discovery in the realm of approximation spaces were studied, among others, in [35].

A study of granule systems was carried out in Polkowski [14],[15], [16], [17], in order to find general properties of granules and formally describe the principal applications of calculi on granules. In definitions of granules and in proofs of granule properties, techniques of mereology were applied as more simple and elegant than those of naive set theory.

The general scheme for inducing granules in Polkowski [16], [17] is as follows.

We fix an information system $I = (U, A)$, and a *rough inclusion* μ on U.

For an object u and a real number $r \in [0, 1]$, we define the *granule* $g_\mu(u, r)$ *about* u *of the radius* r, *relative to* μ, by letting,

$$g_\mu(u, r) \text{ is } ClsF(u, r), \tag{3}$$

where the property $F(u, r)$ is satisfied with an object v if and only if $\mu(v, u, r)$ holds and Cls is the class operator of mereology, see [16, **?**].

It was shown, see,e.g., [15], Theorem 4, that in case of a transitive μ,

$$v \; ing \; g_\mu(u, r) \Leftrightarrow \mu(v, u, r). \tag{4}$$

Property (4) allows for representing the granule $g_\mu(u, r)$ as the list of those objects v for which $\mu(v, u, r)$ holds. It concerns in particular the rough inclusion μ_L by (2).

For a given granulation radius r, and a rough inclusion μ, we form the collection $U^G_{r,\mu} = \{g_\mu(u, r)\}$ of all granules of the radius r relative to μ.

5 Granular decision systems

For each $u \in U$, we select a set $N(u)$ of objects and assign a value of decision $d(u)$ on the basis of values of $d(v)$ for $v \in N(u)$. Our sets $N(u)$ for $u \in U$, are formed as granules of the form $g_\mu(u, r)$ with μ, r fixed; for each such granule g, and each attribute $a \in A \cup \{d\}$, the factored value $a^*(g)$ is defined as $\mathcal{S}(\{a(u) : u \in g\})$; contrary to the practice of using a metric that combines values of all attributes, in our approach, attributes are involved independently; similarity is driven by the rough inclusion μ.

As a result, each granule g does produce a new object g^*, with attribute values $a(g^*) = a^*(g)$ for $a \in A$, possibly not in the data set universe U.

From the set $U^G_{r,\mu}$, of all granules of the form $g_\mu(u, r)$, by means of a strategy $\mathcal{G}$, we choose a covering $Cov^G_{r,\mu}$ of the universe U. Thus, a decision system D^*=$(\{g^* : g \in Cov^G_{r,\mu}\}, A^* \cup \{d^*\})$ is formed, called the *granular counterpart relative to strategies* $\mathcal{G}, \mathcal{S}$ to the decision system $D = (U, A \cup \{d\})$; this new system is substantially smaller in size for intermediate values of r, hence, classifiers induced from it have correspondingly smaller number of rules. Clearly, in case $r = 1.0$, the granular system does coincide with the original system, as granules are in this case indiscernibility classes.

As stated above, our hypothesis is that the granular counterpart D^* at sufficiently large granulation radii r preserves knowledge encoded in the decision system D to a satisfactory degree so given an algorithm $\mathcal{A}$ for rule induction,

classifiers obtained from the training set *D(trn)* and its granular counterpart $D^*(trn)$ should agree with a small error on the test set *D(tst)*.

6 Exemplary test results

Following the line of investigation begun with Table 2, we include in Table 6 best results of other authors on the Australian credit data set obtained with rough set based methods; in the last three rows, we anticipate the results obtained by means of granular data sets and we give also best granular cases from this work. The last result is obtained by means of the concept–dependent granulation and it is discussed in details in the work by P. Artiemjew [1] in these Proceedings.

Table 2. Best results for Australian credit by some rough set based algorithms; in case $*$, reduction in object size is 40.6 percent, reduction in rule number is 43.6 percent; in case $**$, resp. 10.5, 5.9

so	*me*	*acc*	*cov*
[2]	$SNAPM(0.9)$	$error = 0.130$	–
[30]	*simple.templates*	0.929	0.623
[30]	*general.templates*	0.886	0.905
[30]	*closest.simple.templates*	0.821	.1.0
[30]	*closest.gen.templates*	0.855	1.0
[30]	*tolerance.simple.templ.*	0.842	1.0
[30]	*tolerance.gen.templ.*	0.875	1.0
[38]	*adaptive.classifier*	0.863	–
this.work	$granular^*.r = 0.642857$	0.867	1.0
this.work	$granular^{**}.r = 0.714826$	0.875	1.0
[1]	$conceptdependent.r = 0.785714$	0.9970	0.9995

6.1 An example of granulation

In Table 6.1, a random sample of 20 objects with all 9 attribute values from Pima Indians Diabetes data set [37] is given.

Table 3. A sample of 20 objects from Pima Indians Diabetes

obj	*a1*	*a2*	*a3*	*a4*	*a5*	*a6*	*a7*	*a8*	*d*
o1	11	143	94	33	146	36.6	0.254	51	1
o2	4	144	58	28	140	29.5	0.287	37	0
o3	5	124	74	0	0	34	0.22	38	1
o4	8	109	76	39	114	27.9	0.64	31	1
o5	4	122	68	0	0	35	0.394	29	0
o6	0	165	90	33	680	52.3	0.427	23	0
o7	9	152	78	34	171	34.2	0.893	33	1
o8	4	146	78	0	0	38.5	0.52	67	1
o9	1	119	88	41	170	45.3	0.507	26	0
o10	0	95	80	45	92	36.5	0.33	26	0
o11	1	71	62	0	0	21.8	0.416	26	0
o12	6	99	60	19	54	26.9	0.497	32	0
o13	2	108	64	0	0	30.8	0.158	21	0
o14	11	136	84	35	130	28.3	0.26	42	1
o15	2	120	54	0	0	26.8	0.455	27	0
o16	1	106	70	28	135	34.2	0.142	22	0
o17	0	99	0	0	0	25	0.253	22	0
o18	6	125	78	31	0	27.6	0.565	49	1
o19	5	117	86	30	105	39.1	0.*l*251	42	0
o20	2	122	70	27	0	36.8	0.34	27	0

For the granulation radius of $r = 0.25$, and the object $o13$, the granule $g(o13, 0.25)$ consists of objects $o13, o3, o5, o8, o11, o15, o17, o20$, collected in Table 6.1.

Table 4. Objects composing the granule $g(o13, 0.25)$

obj	$a1$	$a2$	$a3$	$a4$	$a5$	$a6$	$a7$	$a8$	d
$o3$	5	124	74	0	0	34	0.22	38	1
$o5$	4	122	68	0	0	35	0.394	29	0
$o8$	4	146	78	0	0	38.5	0.52	67	1
$o11$	1	71	62	0	0	21.8	0.416	26	0
$o13$	**2**	**108**	**64**	**0**	**0**	**30.8**	**0.158**	**21**	**0**
$o15$	2	120	54	0	0	26.8	0.455	27	0
$o17$	0	99	0	0	0	25	0.253	22	0
$o20$	2	122	70	27	0	36.8	0.34	27	0

The majority voting strategy returns the decision value 0 for the new object $g(o13, 0.25)^*$; values of other attributes are determined by this strategy as well; the new object representing the granule $g(o13, 0.25)$ may be, e.g., that in Table 6.1.

Table 5. New object determined by majority voting that represents the granule $g(o13, 0.25)$

$new.obj$	$a1$	$a2$	$a3$	$a4$	$a5$	$a6$	$a7$	$a8$	d
$g(o13, 0.25)^*$	2	122	64	0	0	26.8	0.22	27	0

6.2 Experiments with Australian credit data set

Experiments have been carried out in accordance with the following procedure,

1. the data table (U, A) has been input;
2. classification rules have been found on the training subtable of 50 percent of objects by means of each of the three algorithms;
3. classification of dataset objects in the test subtable of remaining 50 percent of objects has been found for each of the three classifications found at point 2;
4. given the granule radius, granules of that radius have been found on the training subtable;
5. a granular covering of the training subtable has been chosen;
6. the corresponding granular decision system has been determined;
7. granular classifiers have been induced from the granular system in point 6 by means of each of algorithms in point 2;
8. classifications of objects in the test subtable have been found by means of each of classifiers in point 7;
9. classifications from points 3,8 have been compared with respect to adopted global measures of quality: total accuracy and total covering.

Table 6.2 presents results obtained in case of Australian credit data set for random selection of a covering; the radius "nil" denotes results for the original data set without granulation. Total accuracy is the ratio of the number of correctly classified cases to the number of classified cases; total coverage is the ratio of the number of classified cases to the number of cases, see [33]. The algorithm applied is the exhaustive classifier of the RSES system.

Table 6. Australian credit dataset:r=granule radius,tst=test sample size,trn=training sample size,rulex=number of rules with exhaustive algorithm, aex=total accuracy with exhaustive algorithm,cex=total coverage with exhaustive algorithm

r	tst	trn	$rulex$	aex	cex
nil	345	345	5597	0.872	0.994
0.0	345	1	0	0.0	0.0
0.0714286	345	1	0	0.0	0.0
0.142857	345	2	0	0.0	0.0
0.214286	345	3	7	0.641	1.0
0.285714	345	4	10	0.812	1.0
0.357143	345	8	23	0.786	1.0
0.428571	345	20	96	0.791	1.0
0.5	345	51	293	0.838	1.0
0.571429	345	105	933	0.855	1.0
0.642857	345	205	3157	0.867	1.0
0.714286	345	309	5271	0.875	1.0
0.785714	345	340	5563	0.870	1.0
0.857143	345	340	5574	0.864	1.0
0.928571	345	342	5595	0.867	1.0

In Table 6.2, accuracy is within error of 10 percent in comparison to $r = nil$ from the radius of 0.285714 on, and it is better or within error of 4 percent from the radius of 0.5 where reduction in training set size is 85 percent and reduction in rule set size is 95 percent. The result of .875 at $r = .714$ is among the best at all (see Table 2). Coverage is better from $r = .214$ in the granular case, reduction in objects is 99 percent, reduction in rule size is almost 100 percent.

7 Conclusion

The presented results bear out the hypothesis put forth in [16, 17] that granular systems preserve knowledge to a sufficiently high degree; the simplest rough inclusion applied will be rectified in further investigations to produce possibly yet better results.

References

1. P.Artiemjew, Classifiers from Granulated Data Sets: Concept Dependent and Layered Granulation, In: these Proceedings.
2. J.G. Bazan, "A comparison of dynamic and non–dynamic rough set methods for extracting laws from decision tables", In: *Rough Sets in Knowledge Discovery 1*, L. Polkowski, A.Skowron Eds., Physica Verlag: Heidelberg, 1998, 321–365.
3. J.G. Bazan, Hung Son Nguyen, Sinh Hoa Nguyen, P. Synak and J. Wróblewski, "Rough set algorithms in classification problems", In: *Rough Set Methods and Applications*, L.Polkowski, S.Tsumoto and T.Y.Lin (Eds.), Physica Verlag: Heidelberg, 2000, 49–88.
4. R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification*, John Wiley and Sons, New York, 2001.
5. J.W. Grzymala–Busse, "LERS – a system for learning from examples based on rough sets", In: *Intelligent Decision Support: Handbook of Advances and Applications of the Rough Sets Theory*, R. Słowiński Ed., Kluwer: Dordrecht, 1992, 3–18.
6. J.W. Grzymala–Busse, "Data with missing attribute values: Generalization of rule indiscernibility relation and rule induction", *Transactions on Rough Sets I*, subseries of *Lecture Notes in Computer Science*, Springer Verlag: Berlin, 2004, 78–95.

7. J.W.Grzymala–Busse, Ming Hu, "A comparison of several approaches to missing attribute values in Data Mining", *Lecture Notes in Artificial Intelligence* 2005, Springer Verlag: Berlin, 2000, 378–385.
8. P. Hájek, *Metamathematics of Fuzzy Logic*, Kluwer, Dordrecht, 1998.
9. T. Y. Lin, "From rough sets and neighborhood systems to information granulation and computing with words", In: *Proceedings of the European Congress on Intelligent Techniques and Soft Computing*, 1997, 1602–06.
10. T. Y. Lin, "Granular computing: Examples, Intuitions, and Modeling", In: [27], 40–44.
11. Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer: Dordrecht, 1991.
12. Z. Pawlak, "Rough sets", *Int. J. Computer and Information Sci.* 11, 1982, 341–356.
13. L. Polkowski, *Rough Sets. Mathematical Foundations*, Physica Verlag, Heidelberg, 2002.
14. L. Polkowski, "A rough set paradigm for unifying rough set theory and fuzzy set theory" (a plenary lecture), In: *Proceedings RSFDGrC03*, Chongqing, China, 2003, LNAI vol. 2639, Springer Verlag: Berlin, 2003, 70–78; cf. also *Fundamenta Informaticae* 54, 2003, 67–88.
15. L. Polkowski, "Toward rough set foundations. Mereological approach" (a plenary lecture), In: *Proceedings RSCTC04*, Uppsala, Sweden, 2004; LNAI vol. 3066, Springer Verlag: Berlin, 2004, 8–25.
37–45.
16. L. Polkowski, "Formal granular calculi based on rough inclusions" (a feature talk), In: [27], 57–62.
17. L.Polkowski, "A model of granular computing with applications" (a feature talk), In: [28], 9–16.
18. L.Polkowski, "Granulation of knowledge in decision systems: The approach based on rough inclusions. The method and its applications", In: *Proceedings RSEISP 07*, LNAI vol. 4585, Springer Verlag, Berlin, 2007, plenary talk.
19. L.Polkowski and P. Artiemjew, "Granular computing: Granular classifiers and missing values", In: *Proceedings IEEE ICCI07*, Lake Tahoe NV, 2007.
20. L.Polkowski and P. Artiemjew, "On granular rough computing with missing values", In: *Proceedings RSEISP07*, LNAI vol. 4585, Springer Verlag, Berlin, 2007, 271-279.
21. L.Polkowski and P. Artiemjew, "On granular rough computing: Factoring clasifiers through granulated structures", In: *Proceedings RSEISP07*, LNAI vol. 4585, Springer Verlag, Berlin, 2007, 280-289.
22. L.Polkowski and A. Skowron, "Rough mereology: a new paradigm for approximate reasoning", *International Journal of Approximate Reasoning* 15(4), 1997, 333–365.
23. L. Polkowski and A. Skowron, "Grammar systems for distributed synthesis of approximate solutions extracted from experience", In: *Grammatical Models of Multi–Agent Systems*, Gh. Paun and A.Salomaa Eds., Gordon and Breach: Amsterdam, 1999, 316–333.
24. L.Polkowski and A.Skowron, "Towards an adaptive calculus of granules", In: *Computing with Words in Information/Intelligent Systems,1.*, L.A.Zadeh and J.Kacprzyk Eds., Physica Verlag: Heidelberg, 1999, 201–228.
25. L. Polkowski and A. Skowron, "Rough mereological calculi of granules: A rough set approach to computation", *Computational Intelligence. An International Journal* 17(3), 2001, 472–492.

26. L. Polkowski, A.Skowron and J.Zytkow, "Tolerance based rough sets", In: *Soft Computing:Rough Sets, Fuzzy logic, Neural Networks, Uncertainty Management*, T.Y. Lin and M.A. Wildberger Eds., Simulation Councils, Inc.: San Diego, 1995, 55–58.
27. Proceedings of IEEE 2005 Conference on Granular Computing,GrC05, Beijing, China, July 2005, IEEE Computer Society Press, 2005.
28. Proceedings of IEEE 2006 Conference on Granular Computing, GrC06, Atlanta, USA, May 2006, IEEE Computer Society Press, 2006.
29. Qing Liu and Hui Sun, "Theoretical study of granular computing", In: *Proceedings RSKT06*, Chongqing, China, 2006; *Lecture Notes in Artificial Intelligence* 4062, Springer Verlag: Berlin, 2006, 92–102.
30. Sinh Hoa Nguyen, "Regularity analysis and its applications in Data Mining", In: *Rough Set Methods and Applications*, L.Polkowski, S.Tsumoto and T.Y.Lin Eds., Physica Verlag: Heidelberg, 2000, 289–378.
31. A. Skowron and C. Rauszer, "The discernibility matrices and functions in decision systems", In:*Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory*, R. Słowiński Ed., Kluwer: Dordrecht, 1992, 311–362.
32. A. Skowron, "Boolean reasoning for decision rules generation", In: *Methodologies for Intelligent Systems*, J.Komorowski and Z. Ras Eds., LNAI 689, Springer Verlag: Berlin, 1993, 295–305.
33. A. Skowron et al., "RSES: A system for data analysis"; available at http: logic.mimuw.edu.pl/ rses/
34. A. Skowron and C. Rauszer, "The discernibility matrices and functions in decision systems", In:*Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory*, R. Słowiński Ed., Kluwer: Dordrecht, 1992, 311–362.
35. A. Skowron and J. Stepaniuk, "Information granules: towards foundations of granular computing", *International Journal for Intelligent Systems* 16, 2001, 57–85.
36. J. Stefanowski, "On rough set based approaches to induction of decision rules", In: *Rough Sets in Knowledge Discovery 1*, L. Polkowski and A.Skowron Eds., Physica Verlag: Heidelberg, 1998, 500–529.
37. http://www.ics.uci.edu./ mlearn/databases/
38. J. Wróblewski, "Adaptive aspects of combining approximation spaces", In: S.K.Pal, L.Polkowski, A.Skowron, *Rough–neural Computing. Techniques for Computing for Words*, Springer Verlag, 2004, 139–156.
39. Y.Y. Yao, "Granular computing: Basic issues and possible solutions", In: *Proceedings 5th Joint Conference Information Sciences I*, P.P. Wang Ed., Assoc. Intell. Machinery: Atlantic, NJ, 2000, 186–189.
40. Y.Y. Yao, "Information granulation and approximation in a decision–theoretic model of rough sets", In: [**?**], 491–516.
41. Y.Y. Yao, "Perspectives of granular computing", In: [27], 85–90.
42. L.A. Zadeh, "Fuzzy sets and information granularity", In: *Advances in Fuzzy Set Theory and Applications*, M. Gupta, R. Ragade and R. Yager Eds., North–Holland: Amsterdam, 1979, 3–18.
43. L.A. Zadeh, "Graduation and Granulation are keys to computation with information described in Natural Language", In: [27], p.30.

Improving Rule-Based Classifiers Induced by MODLEM by Selective Pre-processing of Imbalanced Data

Jerzy Stefanowski[1] and Szymon Wilk[1,2]

[1] Institute of Computing Science, Poznań University of Technology,
ul. Piotrowo 2, 60–965 Poznań, Poland
jerzy.stefanowski@cs.put.poznan.pl, szymon.wilk@cs.put.poznan.pl
[2] Telfer School of Management, University of Ottawa,
136 Jean-Jacques Lussier Str., K1N 6N5 Ottawa, Canada
wilk@telfer.uottawa.ca

Abstract. In the paper we discuss inducing rule-based classifiers from imbalanced data, where one class (a minority class) is under-represented in comparison to the remaining classes (majority classes). To improve the ability of a classifier to recognize this class, we propose a new selective pre-processing approach that is applied to data before inducing a rule-based classifier. The approach combines selective filtering of the majority classes with focused over-sampling of the minority class. Results of a comparative experimental study show that our approach improves sensitivity for the minority class while preserving the ability of a classifier to recognize examples from the majority classes.

1 Introduction

Many real-life knowledge discovery problems involve learning from *imbalanced data*, which means that one of the classes (further called a *minority class*) includes much smaller number of examples than the others (further referred to as *majority classes*). Moreover, examples from the minority class are usually of primary interest. Such situation is typical for medical problems, where the number of patients requiring special attention (e.g., therapy or treatment) is much smaller than the number of patients who do not need it. Similar situations occur in other domains – in [4, 14] the following problems are reported: detecting fraud/intrusion, managing risk, detecting of oil spills in satellite images, predicting technical equipment failures and information filtering.

Learning methods usually do not work properly on imbalanced data as they are "somehow biased" to focus on the majority classes while "missing" examples from the minority class. As a result created classifiers are also biased toward better recognition of the majority classes and they usually have difficulties (or even are unable) to classify correctly new objects from the minority class. This problem also affects rough set rule-based classifiers as elementary sets for the minority class are "weaker" than the ones for the majority classes and consequently rules generated on their basis have a lesser chance to contribute to the

final classification. Overall classification accuracy is not the only and the best criterion characterizing performance of a classifier induced from imbalanced data. Satisfactory recognition of the minority class may be often more preferred, thus, a classifier should be characterized rather by its *sensitivity* and *specificity* for the minority class (sensitivity is defined as the ratio of correctly recognized examples from the minority class and specificity is the ratio of correctly excluded examples from the majority classes).

Too small number of examples in the minority class is not the only one problem with creating classifiers from imbalanced data. Other problems are: overlapping of examples from the minority class with examples from the majority classes, noise, data fragmentation, inappropriate use of greedy search strategies or evaluation measures. A number of solutions have been proposed to solve them, for review see [5, 14]. The most common are pre-processing techniques that change the distribution of examples among classes by appropriate sampling. Other approaches modify either induction or classification strategy, assign weights to examples, and use boosting or other combined classifiers. Some researchers transform the problem of learning from imbalanced data to the problem of cost learning (although it is not the same and misclassification costs are unequal and unknown) and use techniques from the ROC curve analysis.

We also studied this problem in two different ways. In [7] we introduced an approach that modified the structure of a rule-based classifier to increase its sensitivity. Then in [13] we studied a rough set pre-processing approach, where examples from majority classes belonging to a boundary between rough approximations of the minority class were filtered. Although it improved sensitivity of rule-based classifiers, we also noticed that focusing only on inconsistent examples was not sufficient as other "difficult" examples from lower approximations may still have degraded classification performance. Therefore, now we focus our attention on recent selective methods that change the original class distribution. In particular we are interested in *Synthetic Minority Over-sampling Technique* (SMOTE) [4] and *Neighborhood Cleaning Rule* (NCR) [10]. SMOTE is based on a specialized random introducing artificial examples from the minority class in some regions of data [4]. NCR, on the other hand, removes these examples from the majority classes that are located on the border with the minority class or that may be treated as noise [10]. Although these methods perform well [2], some of their properties could be seen as shortcomings. NCR is focused mainly on improving sensitivity for the minority class what deteriorates recognition of the majority classes. In general, there is a kind of trade-off between sensitivity and specificity but too large drop of specificity may not be accepted. Random introduction of artificial examples by SMOTE may be questionable or quite difficult to interpret and to justify in some domains (e.g., in medicine).

The main goal of this paper is to introduce a new selective pre-processing approach that aims at improving sensitivity for the minority class while preserving the ability of a classifier to recognize the majority classes and keeping overall accuracy at an acceptable level. Our approach combines selective filtering of examples from the majority classes with over-sampling of the minority class,

however, it should remove less examples than NCR. Moreover, unlike SMOTE, it does not introduce any artificial examples, but replicates existing examples from the minority class that are located in "difficult regions" (i.e., they are surrounded by examples from the majority classes).

The second goal is to conduct an experimental evaluation of our pre-processing approach in combination with rule-based classifiers induced by the MODLEM algorithm. We compare it to other pre-processing methods such as SMOTE and NCR also combined with MODLEM classifiers. MODLEM has been chosen for consistency with our previous research on imbalanced data [7, 13] and its usefulness in many classification problems [12].

2 Related Works

We briefly describe only these pre-processing methods, which are related to our proposal; for more extensive reviews see [5, 14]. As the uneven distribution of examples among classes makes induction of classifiers more difficult, sampling methods are used to transform it. The simplest are random *over-sampling* which replicates examples from the minority class and random *under-sampling* which randomly eliminates examples from the majority classes until a required degree of balance between classes is reached. However, random under-sampling may potentially remove some important examples and simple over-sampling may also lead to overfitting. Thus, recent research on sampling suggests focusing on particular examples from the minority class or the majority classes.

In [9] Kubat and Matwin analyzed mutual positions of examples from the majority classes. They distinguish four categories of examples: *noisy* examples located inside the minority class region, *borderline* examples (i.e., these laying either on or very close to the border between the minority class and the majority classes), *redundant* examples (i.e., examples distant from the border between classes) and *safe* examples. They detect these categories by applying Hart's Condensed Nearest Neighbor rule and Tomek links (two closest examples from different classes). Following the ideas of example selection from pattern recognition they introduced *one-side-sampling* approach, where the majority classes are selectively reduced by removing noise, borderline and redundant examples while keeping the minority class unchanged.

Another approach to focused removal of noisy and borderline examples from the majority class is NCR introduced by Laurikkala in [10]. NCR uses the Wilson's Edited Nearest Neighbor rule [15] and it can be shortly summarized in the following way: for each example x , its 3 nearest neighbors are found; if x belongs to one of the majority classes and its nearest neighbors misclassify it, then x is removed; if x belongs to the minority class and its neighbors misclassify it, then the neighbors that belong to the majority classes are removed. Experimental studies [2, 9, 10] demonstrated that both above approaches provided better sensitivity than simple random over-sampling. According to [10] NCR performs better than one-side sampling and considers noisy examples more carefully.

Chawla et al. introduced SMOTE, which selectively over-samples the minority class by creating new synthetic (artificial) examples [4]. Its main idea is to consider each example from the minority class and randomly introduce new artificial examples along the lines joining it with some of its k nearest neighbors from the minority class. SMOTE can generate artificial examples with quantitative and qualitative attributes [4] and the number of nearest neighbors depends on how extensive over-sampling is required. SMOTE is claimed to reduce the danger of overfitting as it does not simple replicate quite specific border examples but increases the "density" between examples from the minority class. A combination of SMOTE with some elements of under-sampling may additionally improve the ability of induced classifiers to recognize the minority class [2, 4].

In our previous research on pre-processing [13] an approach based on rough sets was applied to imbalanced and inconsistent data. We studied two techniques to detect and process inconsistent examples from the majority classes in the boundary between the minority and majority classes. The first one removes these examples from the learning set while the other *relabels* them as the minority class. The idea of relabeling was partly inspired by other research on the Generalized Edited Nearest Neighbor algorithm by Koplowitz and Brown (its description is given in [3]). In experiments these techniques were combined with two rule induction algorithms – LEM2 and MODLEM. Both techniques cleaned the boundary region of the minority class, what allowed inducing less specific rules. Moreover, the relabeling technique by increasing the number of examples in the minority class resulted in stronger rules that in turn led to higher sensitivity [13].

3 New Approach to Selective Pre-processing

Our proposal to selective pre-processing of imbalanced data combines elements of focused removal of examples from the majority classes with over-sampling of the minority class. Although it is inspired by some ideas presented in section 2, we apply them differently. First, we think that one-side-sampling and NCR may remove too many examples from the majority classes. Such greedy "cleaning" should definitely lead to increased sensitivity for the minority class, however, too extensive changes in the majority classes may deteriorate the ability of an induced classifier to recognize examples from these classes. As stated in the introduction, we believe in many problems it is necessary not only to improve sensitivity for the minority class, but also to maintain an acceptable level of overall accuracy.

The other premise for our approach is criticism of over-sampling performed by SMOTE that comes from our experience in analyzing real-life, especially medical data. Namely, we claim that random introduction of artificial examples may be questionable in practice, e.g., artificial "non-existing" patients could be questionable for physicians. SMOTE may introduce quite a high number of such artificial examples as according to [4] it may use the majority of 5 neighbors to generate them. Moreover, the position of new examples is selected in the direction

of the nearest examples from the minority class without checking their relation to the nearest examples from the majority classes. To overcome these shortcomings we check alternative over-sampling for the minority class. It identifies only those examples that are likely to be misclassified and amplifies them and does not modify these examples that are possibly correctly classified.

Our approach to selective pre-preprocessing consists of two phases. In the first phase we analyze the "internal characteristics" of examples by distinguishing between their two types – *safe* and *noisy*. *Safe* examples should be correctly classified by an induced classifier, while *noisy* are very likely to be misclassified and thus require special attention in the second phase. We discover the type of an example by applying the Nearest Neighbor rule with the heterogeneous value distance metric (HVMD) [15] that handles quantitative and qualitative attributes. An example is *safe* if it is correctly classified by its k nearest neighbors, otherwise it is *noisy*. We further divide *safe* examples into *safe-certain* and *safe-possible* depending on the characteristic of their nearest neighbors. Analogously, *noisy* examples are divided into *noisy-certain* and *noisy-possible*.

In the second phase we process examples according to their type. As we want to preserve all examples from the minority class, we assume that only examples from the majority classes may be removed or relabeled (i.e., assigned to the minority class). Unlike previous methods, we want to modify the majority classes more carefully, therefore, we preserve all *safe* examples from the majority classes (let us note that NCR removes some of them if they are too close to *noisy* examples from the minority class). We propose three different techniques of processing examples: *relabeling and amplification*, *weak amplification* and *strong amplification*. They all involve modification of the minority class, however, the degree and scope of changes varies between techniques.

The *relabeling and amplification* technique is inspired by our previous good experience from [13]. It relabels *noisy* examples from the majority classes that are located in the nearest neighborhood of *noisy* examples from the minority class. Then it amplifies those *noisy-certain* examples from the minority class that have only *safe* examples from the majority classes in their nearest neighborhood. The *weak amplification* technique amplifies all *noisy* examples from the minority class. Finally, *strong amplification* also amplifies all *noisy* examples from the majority class, however it does it more extensively. It also amplifies these *safe* examples from the minority class that have *safe* examples from the majority classes in their nearest neighborhood.

Our approach is presented below in details as pseudo-code. We use C to denote the minority class and O to denote a helper class that combines all the majority classes. We also use "flags" to indicate the types of examples , e.g., examples from C are flagged as *C-safe-certain*, *C-safe-possible*, *C-noisy-certain* and *C-noisy-possible*, similar flags are used for examples from O. Moreover, for better readability we introduce "wildcard" flags, e.g., *C-noisy-** denotes both *C-noisy-certain* and *C-noisy-possible*. Finally, we assume $classify\_knn(x, k)$ classifies x using its k nearest neighbors, $knn(x, k, f)$ finds these of k nearest neighbors of example x that are flagged as f, $count\_knn(x, k, c)$ counts how many of k nearest

neighbors of x belong to class c, and $count\_knn(x, k, f)$ counts how many of k nearest neighbors of x are are flagged as f. Following [10] we set k to 3.

```
1: for each x ∈ O do
2:    if classify_knn(x, 3) is correct then
3:       if count_knn(x, 3, O) = 3 then
4:          flag x as O-safe-certain
5:       else
6:          flag x as O-safe-possible
7:    else {classify_knn(x, 3) is incorrect}
8:       if count_knn(x, 3, C) = 3 then
9:          flag x as O-noisy-certain
10:      else
11:         flag x as O-noisy-possible
12: for each x ∈ C do
13:   if classify_knn(x, 3) is correct then
14:      if count_knn(x, 3, C) = 3
         or count_knn(x, 3, O) = count_knn(x, 3, O-noisy-*) then
15:         flag x as C-safe-certain
16:      else
17:         flag x as C-safe-possible
18:   else {classify_knn(x, 3) is incorrect}
19:      if count_knn(x, 3, O) = count_knn(x, 3, O-noisy-*) then
20:         flag x as C-noisy-possible
21:      else
22:         flag x as C-noisy-certain
23: D ← all x ∈ O flagged as O-noisy-*
24: if relabeling and amplification then
25:   for each x flagged as C-noisy-* do
26:      for each y ∈ knn(x, 3, O-noisy-*) do
27:         relabel y by changing its class from O to C
28:         remove y from D
29:   for each x flagged as C-noisy-certain do
30:      amplify x by creating its count_knn(x, 3, O-safe-*) copies
31: else if weak amplification then
32:   for each x flagged as C-noisy-* do
33:      amplify x by creating its count_knn(x, 3, O-safe-*) copies
34: else {strong amplification}
35:   for each x flagged as C-safe-possible do
36:      amplify x by creating its count_knn(x, 3, O-safe-*) copies
37:   for each x flagged as C-noisy-* do
38:      if classify_knn(x, 5) is correct then
39:         amplify x by creating its count_knn(x, 3, O-safe-*) copies
40:      else
41:         amplify x by creating its count_knn(x, 5, O-safe-*) copies
42: remove all x ∈ D
```

The first phase of our approach (lines 1-22) starts with identifying the types of examples from the majority classes. If an example is correctly classified using its 3 nearest neighbors, then it is safe – if all its 3 nearest neighbors are also from the majority classes, then it is flagged as *O-safe-certain* (lines 3-4), otherwise it is flagged as *O-safe-possible* (line 6). If an example is misclassified (line 7) then it is noisy – if all its 3 nearest neighbors are from the minority class, then it is flagged as *O-noisy-certain* (lines 8-9), otherwise it is flagged as *O-noisy-possible* (line 11). In the similar way the types of examples from the minority class are checked (line 12). If an example is classified correctly with its 3 nearest neighbors, then it is safe – if all its 3 nearest neighbors are from the minority class or all examples from the majority classes in its 3-nearest neighborhood are noisy, then it is flagged as *C-safe-certain* (lines 14-15), otherwise it is flagged as *C-safe-possible* (line 17). If an example is misclassified (line 18), then it is noisy – if all examples from the majority classes in its 3-nearest neighborhood are noisy, then it is flagged as *C-noisy-possible* (lines 19-20), otherwise it is flagged as *C-noisy-certain* (line 22).

The second phase (lines 23-42) starts with selecting all *O-noisy-** examples into the removal set D (line 23). Further processing depends on the selected technique. If it is *relabeling and amplification* (line 24), then for each *C-noisy-** example all *O-noisy-** examples in its 3-nearest neighborhood are identified (line 26), relabeled (line 27), and removed from D (line 28). Then each *C-noisy-certain* example is amplified by creating as many of its copies as there are *O-safe-** examples its 3-nearest neighborhood (line 30). If the selected technique is *weak amplification* (line 31), then each *C-noisy-** example is amplified by creating as many of its copies as there are *O-safe-** examples in its 3-nearest neighborhood (line 33). If the selected technique is *strong amplification* (line 34), then each *C-safe-possible* example is amplified by creating as many of its copies as there are *O-safe-** examples its 3-nearest neighborhood (line 36). Then for each *C-noisy-** example we check its extended neighborhood and classify it using its 5 nearest neighbors. If an example is classified correctly, then it is amplified by creating as many of its copies as there are *O-safe-** examples in its 3-nearest neighborhood (lines 38-39). Otherwise if an example is still classified incorrectly, it is stronger amplified by creating as many of its copies as there are *O-safe-** examples in its 5-nearest neighborhood (line 41). Finally, all examples from D are removed from a data set.

The above approach could be combined with any learning algorithm. In this study we combine it with MODLEM – a rough set algorithm for inducing rule-based classifiers, which was introduced by Stefanowski [11]; see also [12] for its detailed description. Shortly speaking, MODLEM follows the idea of sequential covering of rough approximations of decision classes by a minimal set of rules. While creating elementary conditions it handles both qualitative and quantitative attributes, and selection of the best condition is controlled by a criterion based on entropy. A new example is classified by matching its description to all induced rules. As it may lead to ambiguous situations (e.g., multiple match), we employ a strategy described in [12], which uses the strength of matched rules to

solve conflicts (the strength of a rule is defined as the number of learning examples that satisfy the condition and the decision part of this rule). For each class the total strength of matched rules is calculated and the example is assigned to the strongest class. If no rule matches the classified example, the nearest rules are identified using HVDM and their strengths are used to find the strongest class in the same way as for matched rules – for more details see [12].

4 Experimental Study

The aim of experiments was to evaluate classification abilities of rule-based classifiers created by combining three techniques of the the selective pre-processing (relabeling and amplification, weak amplification, strong amplification) with MODLEM. We compared them to the basic approach with classifiers induced by MODLEM directly from imbalanced data (without any pre-processing), and classifiers created by combining SMOTE and NCR with MODLEM. In order to find the best over-sampling degree for SMOTE, we tested its different values and selected the one leading to the highest sensitivity of induced classifiers. Moreover, to extend the comparison we also included MODLEM with an approach that modifies the classification strategy for a rule-based classifier induced directly from a data set (without pre-processing) [7]. This approach was originally introduced by Grzymala in [6] and it is based on the idea of multiplying the strength of all minority class rules by the same real number, called a *strength multiplier*, while not changing the strength of rules from the majority classes. As a result, during such minority class rules have a better chance to classify new objects. The value of the strength multiplier is found by maximizing the measure *gain = sensitivity + specificity*. Implementations of all methods and the MODLEM algorithm were done in Java using the WEKA environment [16].

Table 1. Characteristics of evaluated data sets (N – number of examples, N_A – number of attributes, C – minority class, N_C – number of examples in the minority class, $R_C = N_C/N$ – ratio of examples in the minority class)

Data set	N	N_A	C	N_C	R_C
Acl	140	6	with knee injury	40	0.29
Breast cancer	286	9	recurrence-events	85	0.30
Bupa	345	6	sick	145	0.42
Cleveland	303	13	positive	35	0.12
Ecoli	336	7	imU	35	0.10
Glass	214	9	vehicle_windows_float_processed	17	0.08
Haberman	306	3	died	81	0.26
Hepatitis	147	19	die	31	0.21
New-thyroid	260	5	hyper	35	0.13
Pima	768	8	positive	268	0.35

The experiments were carried out on 10 data sets listed in Table 1. They come either from the UCI repository [1] or from our medical partners (acl). We selected the data sets that were characterized by varying degree of imbalance (ratio of examples in the minority class) and that were used in related works [7, 10]. Several data sets originally included more than two classes, however, to simplify calculations we decided to collapse all majority classes into one.

In the experiments we evaluated sensitivity and specificity for the minority class attained by created classifiers – see Tables 2 and 3. To control the trade-off between these two measures we also calculated their geometric mean, denoted as GM – see Table 4. According to [9] this measure relates to the point on a ROC curve and besides maximizing values of both components it allows to keep them balanced. Finally, we also evaluated overall accuracy – see Table 5. All measures were estimated in the 10-fold stratified cross validation repeated 5 times.

In order to compare the performance of evaluated approaches on all data sets we used the Wilcoxon Signed Ranks Test – a nonparametric test for significant differences between paired observations (confidence $\alpha = 0.05$). Considering sensitivity, all other approaches significantly outperformed the basic approach with no pre-processing. NCR led to the highest increase of sensitivity among all evaluated approaches – the differences between NCR and all other approaches were significant. The second best were two new selective pre-processing techniques: relabeling and amplification (relabel) and strong amplification (strong) – the difference between them was not significant. The third was SMOTE and weak amplification (weak). The approach with the strength multiplier (multiplier) led to the smallest increase of sensitivity.

In case of specificity, the basic approach was significantly better than all other approaches. The differences between the remaining approaches, except NCR, were not significant. Specificity attained by NCR was the lowest. Similar observation applies to overall accuracy – the basic approach was the best, then there were three techniques of new proposed approach, SMOTE and multiplier. All of them were significantly better than overall accuracy achieved by NCR.

NCR provided good results in terms of GM for a few data sets (cleveland, ecoli, glass), where increase of sensitivity caused only slight decrease of specificity. In general, we can conclude that very high increase of sensitivity was usually connected with decrease of specificity and consequently deteriorated overall accuracy. For other data sets, the proposed selective approach often demonstrated good trade off between sensitivity and specificity - although differences were not significant, the highest GM were obtained for the relabel technique.

When comparing the new approach to SMOTE we observed that it led to higher sensitivity allowing to "maintain" similar specificity and overall accuracy. The multiplier approach was the least efficient in improving sensitivity, however, quite good in keeping specificity close to the basic approach.

Table 2. Sensitivity for evaluated approaches and data sets (basic – no pre-processing, relabel – relabeling and amplification, weak - weak amplification, strong – strong amplification, multiplier - strength multiplier)

Data set	None	SMOTE	NCR	Relabel	Weak	Strong	Multiplier
Acl	0.7350	0.7500	0.9100	0.8950	0.8900	0.8900	0.7800
Breast cancer	0.3186	0.4681	0.6381	0.5544	0.4369	0.5386	0.4508
Bupa	0.5199	0.7529	0.8734	0.8375	0.7985	0.8047	0.5973
Cleveland	0.0717	0.1967	0.2850	0.2033	0.1600	0.1883	0.0933
Ecoli	0.4400	0.6300	0.7283	0.6367	0.6233	0.6333	0.4683
Glass	0.1700	0.2800	0.3400	0.2800	0.3200	0.3100	0.1800
Haberman	0.2397	0.3139	0.6258	0.4681	0.4039	0.4828	0.4011
Hepatitis	0.3833	0.4167	0.5300	0.5250	0.4283	0.4617	0.4950
New-thyroid	0.8067	0.8950	0.8100	0.8500	0.8467	0.8883	0.8233
Pima	0.4853	0.6147	0.7933	0.7377	0.6853	0.7377	0.7050

Table 3. Specificity for evaluated approaches and data sets (basic – no pre-processing, relabel – relabeling and amplification, weak - weak amplification, strong – strong amplification, multiplier - strength multiplier)

Data set	None	SMOTE	NCR	Relabel	Weak	Strong	Multiplier
Acl	0.9220	0.9080	0.8320	0.8860	0.8860	0.8860	0.9060
Breast cancer	0.8043	0.6570	0.5227	0.6212	0.7097	0.6061	0.6866
Bupa	0.8200	0.5720	0.3080	0.3930	0.4530	0.4590	0.7690
Cleveland	0.9553	0.9017	0.9092	0.9381	0.9418	0.9412	0.9441
Ecoli	0.9714	0.9462	0.9235	0.9514	0.9641	0.9581	0.9674
Glass	0.9818	0.9788	0.9634	0.9737	0.9758	0.9778	0.9778
Haberman	0.8155	0.7720	0.6583	0.7196	0.7455	0.7127	0.7366
Hepatitis	0.9208	0.9315	0.8570	0.9147	0.9062	0.9168	0.8867
New-thyroid	0.9900	0.9844	0.9844	0.9867	0.9878	0.9856	0.9856
Pima	0.8556	0.7852	0.6580	0.7204	0.7736	0.6980	0.7092

Table 4. GM for evaluated approaches and data sets (basic – no pre-processing, relabel – relabeling and amplification, weak - weak amplification, strong – strong amplification, multiplier - strength multiplier)

Data set	None	SMOTE	NCR	Relabel	Weak	Strong	Multiplier
Acl	0.8232	0.8252	0.8701	0.8905	0.8880	0.8880	0.8406
Breast cancer	0.5062	0.5546	0.5775	0.5869	0.5568	0.5714	0.5563
Bupa	0.6529	0.6562	0.5187	0.5737	0.6014	0.6077	0.6777
Cleveland	0.2617	0.4211	0.5090	0.4367	0.3882	0.4210	0.2968
Ecoli	0.6538	0.7721	0.8201	0.7783	0.7752	0.7790	0.6731
Glass	0.4085	0.5235	0.5723	0.5221	0.5588	0.5506	0.4195
Haberman	0.4421	0.4923	0.6418	0.5804	0.5487	0.5866	0.5436
Hepatitis	0.5941	0.6230	0.6740	0.6930	0.6230	0.6506	0.6625
New-thyroid	0.8937	0.9386	0.8930	0.9158	0.9145	0.9357	0.9008
Pima	0.6444	0.6947	0.7225	0.7290	0.7281	0.7176	0.7071

Table 5. Overall accuracy for evaluated approaches and data sets (basic – no pre-processing, relabel – relabeling and amplification, weak - weak amplification, strong – strong amplification, multiplier - strength multiplier)

Data set	None	SMOTE	NCR	Relabel	Weak	Strong	Multiplier
Acl	86.86%	86.29%	85.43%	88.86%	88.71%	88.71%	87.00%
Breast cancer	65.97%	60.02%	55.62%	60.07%	62.85%	58.58%	61.62%
Bupa	69.36%	64.79%	54.54%	57.94%	59.80%	60.38%	69.65%
Cleveland	85.35%	82.05%	83.72%	85.36%	85.17%	85.43%	84.63%
Ecoli	91.62%	91.38%	90.36%	91.90%	92.87%	92.45%	91.56%
Glass	91.80%	92.43%	91.49%	91.90%	92.47%	92.55%	91.52%
Haberman	66.29%	65.04%	64.93%	65.23%	65.45%	65.12%	64.73%
Hepatitis	80.79%	82.34%	78.87%	83.19%	80.59%	82.00%	80.39%
New-thyroid	96.01%	97.03%	95.63%	96.47%	96.48%	97.03%	95.91%
Pima	72.65%	72.58%	70.55%	72.65%	74.29%	71.20%	70.77%

5 Conclusions

In the paper we introduced the new approach to selective pre-processing of imbalanced data that aims at improving sensitivity of an induced classifier, while keeping overall accuracy at an acceptable level. It combines selective filtering of the majority classes with over-sampling of the minority class. Our approach removes less examples than NCR and, unlike SMOTE, it does not introduce any artificial examples, but replicates some of existing ones. Moreover, it does not require the parameterized degree of oversampling as it identifies minority class regions difficult to classify and modify only these examples, which could be misclassified. Within the proposed approach we developed three techniques of processing these examples involving amplification of examples from the minority class and relabeling examples from the majority classes – relabeling and amplification, weak amplification and strong amplification.

Our approach was verified in the experimental study where we compared it to other pre-processing methods, the basic basic approach with no-preprocessing and the approach that changes classification strategy. All these approaches were combined with rule-based classifiers. Results of experiments supported our initial intuition for NCR as a method strongly oriented toward improvement of sensitivity by extensive "cleaning" examples from the majority classes. Such cleaning made the majority classes more difficult to classify, thus, improvement of sensitivity was at a cost of decreased accuracy for the majority classes. Our approach was a bit worse (but it was the second best among all evaluated approaches) in terms of improving sensitivity, however, it demonstrated better specificity and overall accuracy. Moreover, when comparing the three techniques within the new proposed approach we could notice that more radical techniques (relabeling and amplification, strong amplification) were more efficient than weaker changes of class distribution (weak amplification). Similar experiments were also conducted using the C4.5 algorithm and tree-based classifiers. Although relative improvements of sensitivity were smaller, general behavior of compared

approaches remained unchanged. Thus, we can conclude that the new proposed selective pre-processing approach leads to improved sensitivity for the minority class while preserving overall accuracy for various types of classifiers.

References

1. Blake, C., Koegh, E., Mertz, C.J.,: Repository of Machine Learning, University of California at Irvine 1999 [URL: http://www.ics.uci.edu/ mlearn/MLRepository.html].
2. Batista, G., Prati, R., Monard, M.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter, **6**(1), (2004) 20–29
3. Caballero, Y., Bello, R., Garcia, M., et al.: Using rough sets to edit training set in k-NN method. In: Proc. of 5th ISDA 2006 Conf., IEEE Press, (2006) 456-463.
4. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: Synthetic Minority Over-sampling Technique. J. of Articifical Intelligence Research, **16** (2002) 341–378.
5. Chawla, N., Japkowicz, N., Kolcz, A.: Editorial: Special Issue on Learning from Imbalanced Data Sets. ACM SIGKDD Explorations Newsletter, **6**(1), (2004) 1–6.
6. Grzymala-Busse, J.W., Goodwin, L.K., Grzymala-Busse, W.J., Zheng X.: An approach to imbalanced data sets based on changing rule strength. In: Proc. Learning from Imbalanced Data Sets, AAAI Workshop at the 17th Conference on AI, AAAI-2000, Austin, TX, July 30–31 (2000) 69–74.
7. Grzymala-Busse, J.W., Stefanowski, J., Wilk, Sz.: A comparison of two approaches to data mining from imbalanced data. In: Proceedings of the KES 2004, 8-th International Conference on Knowledge-based Intelligent Information & Engineering Systems, Wellington, New Zealand, Springer LNCS **3213** (2004) 757–763.
8. Nickerson, A., Japkowicz, N., Milios, E.: Using unspervised learning to guide resampling in imbalanced data sets. In: Proc. of the 8th Int. Workshop on Artificial Intelligence and Statistics, (2001) 261–265.
9. Kubat, M., Matwin, S.: Addresing the curse of imbalanced training sets: one-side selection. In: Proc. of 14th Int. Conf. on Machine Learning ICML 97, (1997) 179-186.
10. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. Tech. Report A-2001-2, University of Tampere (2001).
11. Stefanowski, J.: The rough set based rule induction technique for classification problems. In: Proc. of 6th European Conference on Intelligent Techniques and Soft Computing EUFIT'98, Aaachen 7-10 Sept. (1998) 109-113.
12. Stefanowski, J.: On combined classifiers, rule induction and rough sets. In Peters J. et al. (eds.): Transactions on Rough Sets VI, Springer LNCS **4374** (2007) 329-350.
13. Stefanowski, J., Wilk, Sz.: Rough sets for handling imbalanced data: combining filtering and rule-based classifiers. Fundamenta Informaticae Journal, **72**(1-3), (2006) 379-391.
14. Weiss, G.M.: Mining with rarity: a unifying framework. ACM SIGKDD Explorations Newsletter, **6**(1), (2004) 7–19.
15. Wilson, D.R., Martinez, T.: Reduction techniques for instance-based learning algorithms. Machine Learning Journal, **38** (2000) 257–286.
16. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann (1999).

Author Index