An Effective Hybrid Classifier Based on Rough Sets and Neural Networks

Bai Rujiang,Wang Xiaoyue
Shandong University of Technology Library, zibo 255049,China
brj@sdut.edu.cn , wangxy@sdut.edu.cn

Abstract

Due to the exponential growth of documents on the Internet and the emergent need to organize them, the automated categorization of documents into predefined labels has received an ever-increased attention in the recent years. This paper describes a method developed for the automatic clustering of documents by using a hybrid classifier based on rough sets and neural networks, which we called as Rough-Ann,. First, the documents are denoted by vector space model and the feature vectors are reduced by using rough sets. Then using those feature vectors we reduced that are training set for artificial neural network and clustering the documents. The experimental results show that the algorithm Rough-Ann is effective for the documents classification, and has the better performance in classification precision, stability and fault-tolerance comparing with the traditional classification methods, Bayesian classifiers SVM and kNN, especially for the complex classification problems with many feature vectors.

1. Introduction

With the exponential growth of textual information available from the Internet, there has been an emergent need to find and organize relevant information in text collections. For this purpose, automatic text categorization becomes a significant tool to utilize text information efficiently and effectively. Text categorization aims to automatically place the documents on previously classes. It is an active research area in information retrieval, machine learning and natural language processing. At present there are many textual classification methods, such as the support vector machine (SVM), the Nearest Neighbor, the Decision Tree, the K Nearest Neighbor method (KNN) and others.

However, it is hard to find an algorithm performance in classification precision, stability and fault-tolerance in those traditional classification methods. Hence we proposed a hybrid classifier based on the combination of rough set theory and neural networks.

As we know many classification problems involve high-dimensional descriptions of input features. The traditional classification methods can't deal with this problem effectively. So we have to do much research on dimensionality reduction.[1] However existing work tends to destroy the underlying semantics of the features after reduction (e.g. transformation-based approaches [2]) or require additional information about the given data set for thresholding (e.g. entropy-based approaches [3]). A technique that can reduce dimensionality using information contained within the data set and preserving the meaning of the features is clearly desirable. Rough set theory (RST) can be used as such a tool to discover data dependencies and reduce the number of attributes contained in a data set by purely structural methods [4].

Due to RST is sensitive regarding the decision table noise data, we utilize the neural network algorithm carry on the classification.

The paper is organized as follows. In Section 2 we introduce basic notions. In Section 3 we detailedly discuss the algorithm Rough-Ann. In Section 4 the results of experiments on three standard data sets are included. In Section 5 we give the conclusion.

2. Basic notions

2.1 Rough Sets

An information system is a 4-tuple $S=\langle U,A,V,f\rangle$, where U is a finite set of objects, called the universe, A is a finite set of attributes. $V=U_{a\in A}V_a$ is a domain of attribute a, and $f:=U\times A\rightarrow V$ is called an information function such that $f(x,a)\in V_a$,for $\forall a\in A,\forall x\in U$.

In the classification problems, an information system is also seen as a decision table assuming that $A=C\cup D$ and $C\cap D=\phi$, where *C* is a set of condition attributes and *D* is a set of decision attributes.

Proceedings of the 2006 IEEE/WIC/ACM International Conference on
Web Intelligence and Intelligent Agent Technology (WI-IAT 2006 Workshops)(WI-IATW'06)
0-7695-2749-3/06 $20.00 © 2006 IEEE

Let $S=\langle U,A,V,f\rangle$ be an information system, every $P\subseteq A$ generates a indiscernibility relation *IND(P)* on *U*, which is defined as follows:

$$IND(P)=\{(x,y)\in U\times U: f(x,a)=f(y,a),\forall a\in P\} \tag{1}$$

$U/IND(P)=\{C_1,C_2,\cdots C_k,\}$ is a partition of *U* by *P*, every C_i is an equivalence class. For $\forall x\in U$, the equivalence class of x in relation $U/IND(P)$ is defined as follows:

$$[x]_{IND(P)}=\{y\in U: f(y,a)=f(x,a),\forall a\in P\} \tag{2}$$

Let $P\subseteq A, X\subseteq U$. The P-lower approximation of X (denoted by $P_{-}X$) and the P-upper approximation of X (denoted by $P^{-}X$) are defined as follows:

$$P_{-}X=\{y\in U:[y]_{IND(P)}\subseteq X\} \tag{3}$$

$$P^{-}X=\{y\in[y]_{IND(P)}\cap X\neq\phi\} \tag{4}$$

$P_{-}X$ is the set of all objects from U which can be certainly classified as elements of X employing the set of attributes P. $P^{-}X$ is the set of objects of U which can be possibly classified as elements of X using the set of attributes P.

Let $P,Q\subseteq A$, the positive region of classification $U/IND(Q)$ with respect to the set of attributes P, or in short, P-positive region of Q, is defined as

$$pos_P(Q)=\bigcup_{X\in U/IND(Q)} P_{-}X \tag{5}$$

$pos_P(Q)$ contains all objects in U that can be classified to one class of the classification $U/IND(Q)$ by attributes P. The dependency of Q on P is defined as

$$\gamma_P(Q)=\frac{card(POS_P(Q))}{card(U)}. \tag{6}$$

An attribute a is said to be dispensable in P with respect to Q, if $\gamma_p(Q)=\gamma_{P-\{a\}}(Q)$; otherwise a is an indispensable attribute in P with respect to Q.

Let $S=\langle U,C\cup D,V,f\rangle$ be a decision table, the set of attributes $P(P\subseteq C)$ is a reduct of attributes C, which satisfies the following conditions:

$$\gamma_P(D)=\gamma_C(D) \text{ and } \gamma_P(D)\neq\gamma_{P'}(D),\forall P'\subset P. \tag{7}$$

A reduct of condition attributes C is a subset that can discern decision classes with the same discriminating capability as C, and none of the attributes in the reduct can be eliminated without decreasing its discriminating capability.

2.2. Attribute reduction by rough sets

Attribute reduction (feature selection) is a process of finding an optimal subset of all attributes according to some criterion so that the attribute subset are good enough to represent the classification relation of data. A good choice of attribute subset provided to a classifier can increase its accuracy, save the computational time, and simplify its results [5].

In general, rough set theory provides useful techniques to reduce irrelevant and redundant attributes from a large database with a lot of attributes [6,7,8]. In Section 3 we detailedly describe this algorithm

3. Rough-Ann Classifier

The Rough Sets and the neural networks common characteristic is all can under the natural environment the very good work, but, Rough Sets theory method simulation humanity's abstract logical thinking, neural networks method simulation vivid intuition thought, thus the two also has the different characteristic.

First, the neural network processing information generally cannot input the information space dimension simplifies, when the input information space dimension is bigger, not only the structure of network is complex, moreover the training time is also long; Rough Sets method actually can find the relationship between the data, not only may remove the redundancy input information, moreover may simplify the input information the expression space dimension. Second, Rough Sets is more sensitive in processing the noise data, otherwise the neural network method can avoid the effect of noise data.

Therefore we present a hybrid system. It consists of preprocessing of documents, feature reduction process using Rough Sets, and web classification process using error back-propagation neural networks. Fig. 1 shows the procedures of our hybrid approach.

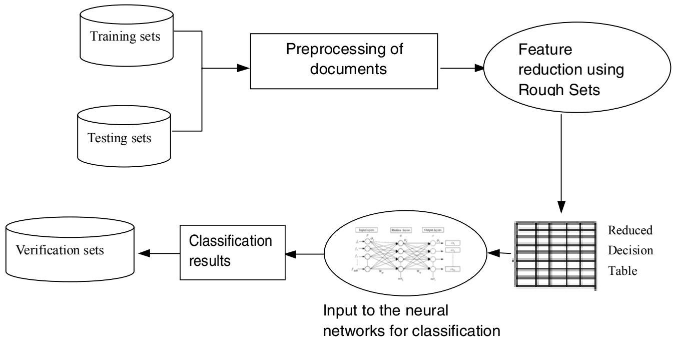

Training sets
Testing sets
Preprocessing of documents
Feature reduction using Rough Sets
Reduced Decision Table
Classification results
Verification sets
Input to the neural networks for classification

Fig. 1: The process of classifying documents using the Rough-Ann method.

3.1 Preprocessing of documents

Preprocessing of documents is a process of removing the most frequent word that exists in a document such as 'to', 'and', 'it', etc. Removing these words will save spaces for storing document contents and reduce time taken during the classification process. After these processes, we will represent them as the document-term frequency matrix ($Doc_j \times TF_{jk}$) as shown in Table 1.

Table 1 The document-term frequency data matrix after processes

Doc_j	TF_1	TF_2	$\ldots$	TF_m
Doc_1	2	4	$\ldots$	5
Doc_2	2	3	$\ldots$	2
Doc_3	2	3	$\ldots$	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Doc_n	4	3	$\ldots$	3

Doc_j is referring to each document that exists in the database where j = 1, . . . ,n. Term frequency TF_{jk} is the number of how many times the distinct word w_k occurs in document Doc_j where *k =1, . . . ,m*. The calculation of the terms weight x_{jk} of each word w_k is done by using a method that has been used by Salton [9] which is given by

$$x_{jk} = TF_{jk} \times idf_k, \qquad (8)$$

Where the document frequency df_k is the total number of documents in the database that contains the word w_k .The inverse document frequency $idf_k = \log(\frac{n}{df_k})$ where n is the total number of documents in the database.

3.2 Attribute reduction process using Rough Sets

The high dimensionality of feature vectors to be an input to classifier is not practical due to poor scalability and performance. Therefore, we develop our algorithms of attribute reduction which replaces complex set operations by simple bit-wise operations in the process of classification. Even if the initial number of attributes is very large, using the measure can effectively delete irrelevant and redundant attributes in a relatively short time.

3.2.1. Binary discernibility matrix[10]. Let $T = \langle U, C \cup D, V, f \rangle$ be a decision table, $U = \{x_1, x_2, \cdots x_m\}$, $C = \{C_1, C_2, \cdots C_n\}$.In general, *D* can be transformed into a set that has only one element without changing the classification for *U*, that is, *D ={d}*.Every value of d corresponds to one equivalence class of $U / IND(D)$,which is also called the class label of object.

A binary discernibility matrix represents the discernibility between pairs of objects in a decision table. Let M be the binary discernibility matrix of *S*, its element *M((s,t),i)* indicates the discernibility between two objects x_s and x_t with different class labels by a single condition attribute c_i ,which is defined as follows:

$$M((s,t),i) = \begin{cases} 1 & \mathrm{c_i}(x_s) \neq c_i(x_t), \\ 0 & \text{otherwise} \end{cases} \qquad (9)$$

Where $1 \leq s < t \leq m$ and $d(x_x) \neq d(x_t), i \in \{1,2,\cdots,n\}$.

It can be seen that *M* has *n* columns and its maximal number of rows is *m(m-1)/2*. Each column of *M* represents a single condition attribute and each row of *M* represents an object pair having different *d* values.

Definition 3.1. Let *M* be a binary discernibility matrix having *R* rows and *L* columns, and its element value in the ith row jth column is a_{ij}. The discernibility degree of an attribute c_k for classification is defined as

$$Deg(c_k) = \frac{1}{R} \sum_{i=1}^{R} a_{ik}, \qquad (10)$$

Where $k \in \{1, 2, \cdots, L\}$.

The Deg of an attribute c_k is in fact the rate of "1"s in the c_k column of M and can be used as a measure of classification capability of attributes.

3.2.2. Attribute reduction by rough sets. More than one reduct of condition attributes may exist for a decision table. Which is the best one depends on the optimality criterion associated with the attributes. Here we assume that the context of decision table is the only information source. Our objective is to find a reduct with minimal number of attributes. Based on the definition of the binary discernibility matrix, we propose our rough set attribute reduction (RSAR) algorithm of finding a reduct of a decision table, which is outlined below.

The Rough Sets Attribute Reduction algorithm is given in Alg. 1.

Input: a decision table $T = \langle U, C \cup D, V, f \rangle$,$U = \{x_1, x_2, \cdots x_m\}$,$C = \{C_1, C_2, \cdots C_n\}$

Output: a reduct of T , denoted as Redu.

1. construct the binary discernibility matrix *M* of *T* ;

```
2. delete the rows in the M which are all 0's, Redu=$\phi$
/* delete pairs of inconsistent objects*/
3. while $(M \neq \phi)$
  {
    (1) select an attribute ci in the M with the highest
        discernibility degree (if there are several
        $c_j$ (j=1,2,…,m) with the same highest
        discernibility degree, choose randomly an
        attribute from them);
    (2) Redu $\leftarrow$ Redu$\cup\{c_i\}$;
    (3) remove the rows which have ''1'' in the
        $c_i$ column from $M$;
    (4) remove the $c_i$ column from $M$;
  }endwhile
/* the following steps remove redundant attributes from Redu */
4. suppose that Redu = $\{r_1, r_2, \cdots r_k\}$ contains k
   attributes which are sorted by the order of entering
   Redu, $r_k$ is the first attributes chosen into Redu, $r_1$ is
   the last one chosen into Redu.
5. get the binary discernibility matrix MR of decision
   table TR=$\langle U, \mathrm{Re}du \cup \{d\}, V, f \rangle$;
6. delete the rows in the MR which are all 0's;
7. for i = 2 to k{
    remove the $r_i$ column from $MR$;
    if (no row in the MR is all 0's)
    Redu $\leftarrow$ Redu – { $r_i$ };
        else
        Put the $r_i$ column back to MR;
        Endif
    }Endfor
```

Alg. 1. Rough Sets Attribute Reduction algorithm

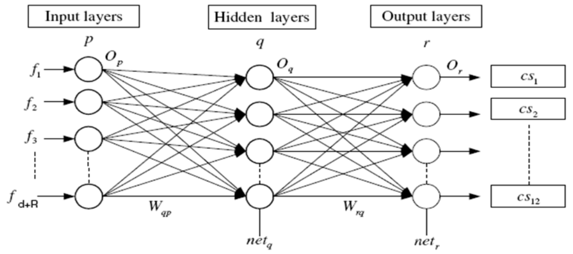
The steps 1–3 are a process of selecting a subset of *C* to discern all objects that can be discerned by the *C*, that is, the subset selected has the same classification capability as all the condition attributes. The steps 4–7 are to delete the redundant attributes of the subset so that it is minimal.

3.3 Input data to the neural networks for classification

After preprocessing of documents and attribute reduction process, the problem of high dimensionality of feature vectors is solved. The next step is input the reducted data to the neural networks for classification.

3.3.1 Characterization of the neural networks

The architecture of neural networks that have been used for classification process is shown in Fig. 2.



f_{d+R} represents input vectors

cs_a represents a class of training data or testing data where a=1,2,…,n

Fig. 2: The feature vectors from attribute reduction are fed to the neural networks for classification.

The number of input layers (*p*) is based on the number of feature vectors. The trial and error approach has been used to find a suitable number of hidden layers that provides good classification accuracy based on the input data to the neural networks. The number of output layers (*r*) is based on the number of classes in the training data set.

We have defined t as the iteration number, η is a learning rate, α is a momentum rate, θ_q is a bias on hidden unit q, θ_r is a bias on output unit r, δ_q is the generalized error through a layer q, and θ_r is the generalized error between layers q and r. The input values to the neural network are represented by $f_1, f_1, \cdots f_{d+R}$.

$$f_i = \begin{cases} 1, & \text{the ith characteristic word in document} \\ 0, & \textit{otherwise} \end{cases}$$

Adaptation of the weights between hidden (q) and input (p) layers is given by

$$W_{qp}(t+1) = W_{qp(t)} + \Delta W_{qp}(t+1), \tag{11}$$

where

$$\Delta W_{qp}(t+1) = \eta \delta_q O_p + \alpha \Delta W_{qp}(t), \tag{12}$$

$$\delta_q = O_q(1 - O_q) \sum_r \delta_r W_{rq} \tag{13}$$

Note that the first transfer function at hidden layer (q) is given by

$$net_q = \sum_q W_{qp} O_p + \theta_q \text{ ,} \tag{14}$$

$$O_q = f(net_q) = 1/(1 + e^{-net_q}). \tag{15}$$

Adaptation of the weights between output (r) and hidden (q) layers is given by

$$W_{rq}(t+1) = W_{rq(t)} + \Delta W_{rq}(t+1), \tag{16}$$

where

$$\Delta W_{rq}(t+1) = \eta \delta_r O_q + \alpha \Delta W_{rq}(t), \tag{17}$$

$$\delta_q = O_r(1 - O_r)(t_r - O_r) \tag{18}$$

Then the output function at the output layer (k) is given by

$$net_r = \sum_r W_{rq} O_q + \theta_r \qquad (19)$$

$$O_r = f(net_r) = 1/(1+e^{-net_r}). \qquad (20)$$

The neural networks classifier algorithm is given in Alg.2.

```
Input: input vectors $f_1, f_1, \cdots f_{d+R}$.
Output: the class of data set
1 for k=1 to L do
  1.1Init $W(k)$;
2 Init $\varepsilon$ ;
3 $E=\varepsilon+1$;
4 while $E>\varepsilon$ do
    4.1 $E=0$;
    4.2 To in S each sample（$Xp,Yp$）：
        4.2.1 Calculated $Xp$ Corresponding actual output $Op$；
        4.2.2 Calculated $Ep$；
        4.2.3 $E=E+Ep$；
        4.2.4 Adjusts $W(L)$ according to the corresponding formula；
        4.2.5 $k=L-1$；
        4.2.6 while $k\neq 0$ do
            4.2.6.1 Adjusts $W(k)$ according to the corresponding formula；
            4.2.6.2 $k=k-1$
    4.3 $E=E/2.0$
```

Alg.2. The neural networks classifier algorithm

4. Experiments

4.1. The datasets

In our experiment, we use three corpora: Reuter-21578,Industry Sector and TDT-5.

4.1.1. Reuter-21578. The Reuters-21578 text categorization test collection contains documents collected from the Reuters newswire in 1987. It is a standard text categorization benchmark and contains 135 categories. We used its subset: one consisting of 92 categories and in total 10,346 documents.

4.1.2. Sector-48. The Industry Section dataset is based on the data made available by Market Guide, Inc. (www.marketguide.com). The set consists of company homepages that are categorized in a hierarchy of industry sectors, but we disregarded the hierarchy. There were 9637 documents in the dataset, which were divided into 105 classes. We use a subset called as Sector-48 consisting of 48 categories and in all 4581 documents.

4.1.3. TDT-5.TDT-5 is the NIST Topic Detection and Tracking text corpus version 1.1 released in September 10, 2004. This corpus contains news data collected daily from news sources in three languages (American English, Mandarin Chinese and Arabic), over a period of 6 months (April 1–September 30 in 2003). We selected the English documents having annotated topics. The resulting dataset contains 126 categories and in total 6364 documents.

4.2. The performance measure

To evaluate a text classification system, we use the F1 measure that combines recall and precision in the following way:

$$\mathrm{Re}\,call = \frac{\text{number of correct positive predictions}}{\text{number of positive examples}} \qquad (21)$$

$$\mathrm{Pr}\,ecision = \frac{\text{number of correct positive predictions}}{\text{number of positive predictions}} \qquad (22)$$

$$F_1 = \frac{2\,\mathrm{Re}\,call \times \mathrm{Pr}\,ecision}{\mathrm{Re}\,call + \mathrm{Pr}\,ecision} \qquad (23)$$

For ease of comparison, we summarize the *F1* scores over the different categories using the Micro- and Macro- averages of *F1* scores:

Micro-F1 = *F1* over categories and documents

Macro-F1 = average of within-category *F1* values

The *Micro-* and *Macro-F1* emphasize the performance of the system on common and rare categories, respectively. Using these averages, we can observe the effect of different kinds of data on a text classification system.

4.3. Experiment design

In all our experiments, we adopt three-fold cross validation. We split each dataset into three parts. Then we use two parts for training and the remaining third for test. We conduct the training-test procedure three times and use the average of the three performances as final result. Algorithms are coded in C++ and running on a Pentium-4 machine with single 2.0 GHz CPU. 4.4

4.4. Comparison and analysis

Now we present and discuss the experimental results. Here we compare Rough-Ann against KNN, VSM, Naïve Bayes on three text corpora.

Fig. 3 and 4 show the best-performance comparison in MicroF1 and MacroF1. On Reuter-21578, the MicroF1 of Rough-Ann is 84.97%, which is approximately 14% higher than that of SVM, 5.9% higher than that of Naïve Bayes, 7% higher than that of KNN. On Sector-48, the MicroF1 of Rough-Ann beats Naïve Bayes by 6%, SVM by 5%, KNN by approximately 4%. Fig. 5 and 6 show the class time and the training time of different methods on three corpora. We can obtain that the training and class time of Rough-Ann is the shortest. In a word Rough-Ann yields top-notch performance among these algorithms. Consequently, we can say that Rough-Ann is a competitive algorithm in text classification.

Fig. 3. The best MicroF1 of different methods on three corpora

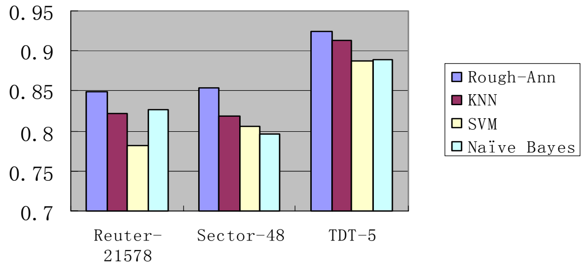


Fig. 4.The best MacroF1 of different methods on three corpora

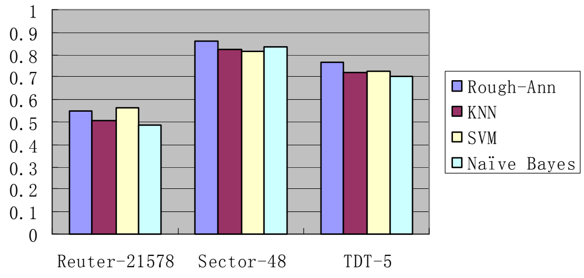


Fig. 5. The training time of different methods on three corpora

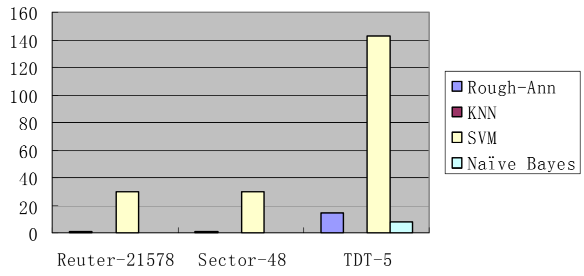
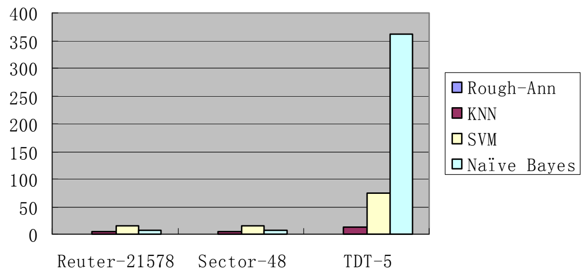


Fig. 6. The time of classification of different methods on three corpora



5. Conclusion

In this paper we proposed an effective refinement classifier, called Rough-Ann. Our technique does not need to generate sophisticated models but only requires simple statistical data and the traditional Rough sets and neural networks. The experiments on three benchmark evaluation collections showed that Rough-Ann is effective for the documents classification, and has the better performance in classification precision, stability and fault-tolerance comparing with the traditional classification methods. The results reported here are not necessarily the best that can be achieved. To seek new techniques to enhance the performance of Rough-Ann is an important issue in future study.

References

[1] Dash, H. Liu, Feature selection for classi6cation, Intell. Data Anal. 1 (3) (1997) 131–156.

[2] P. Devijver, J. Kittler, Pattern Recognition: A Statistical Approach, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[3] T. Mitchell, Machine Learning, McGraw-Hill, New York, 1997.

[4] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishing, Dordrecht,1991.

[5] M. Last, A. Kandel, O. Maimon, Information-theoretic algorithm for feature selection, Pattern Recognition Letters 22 (2001) 799–811.

[6] J. Jelonek, K. Krawiec, R. Slowinski, Rough set reduction of attributes and their domains for neural networks,Computational Intelligence 11 (2) (1995) 339–347.

[7] R. Swiniarski, L. Hargis, Rough set as a front end of neural-networks texture classifiers, Neurocomputing 36 (1–4) (2001) 85–102.

[8] N. Zhong, J. Dong, S. Ohsuga, Using rough sets with Heuristics for feature selection, Journal of Intelligent Information Systems 16 (2001) 199–214.

[9] Salton&McGill, Introduction to modern information retrieval, New York, McGraw-Hill, USA, 1983.

[10] R. Felix, T. Ushio, Rule induction from inconsistent and incomplete data using rough sets, in: IEEE International Conference on Systems, Man, and Cybernetics, 1999, pp. 154–158.

[11] Stepaniuk J.: Knowledge discovery by application of rough set models, L.Polkowski, S. Tsumoto, T.Y. Lin, (Eds.), Rough Sets: New Developments,Physica-Verlag, Heidelberg, 2000, 137-233.

[12] R.Swiniarski, A software system for data mining and rough and fuzzy sets based classification:RoughFuzzyLab, in: S. Tsumoto, Y.Y. Yao (Eds.), Bull. Int. Rough Sets Society 2 (1) (1998) 40-41.