**RIGA TECHNICAL UNIVERSITY**
Faculty of Computer Science and Information Technology
Institute of Information Technology

**Peteris GRABUSTS**

# INVESTIGATION AND DEVELOPMENT OF CONDITIONAL RULE CONSTRUCTION METHODS USING MULTIDIMENSIONAL DATA SAMPLES

**Summary of Doctoral Thesis**

Research supervisor
Dr.habil.sc.comp.,prof.
A. BORISOV

Riga 2006

# GENERAL DESCRIPTION OF WORK

Nowadays, intelligent data analysis experiences the period of boom. This is mainly due to the new idea implementation at the intersection of several research areas such as artificial intelligence, statistics, and database methods. Methods of data analysis and automatic processing are treated as knowledge discovery. In many cases it is necessary to classify data in some way or find regularities in the data. That is why the notion of *regularity* is becoming more and more important in the context of intelligent data processing systems. It is frequently required to ascertain how the data are interrelated, how various data differ or agree with each other, and what the measure of their comparison is. Intelligent data analysis can also be defined as follows. This is a decision making process that is based on finding regularities in the data. The majority of methods of that kind were initially developed in the 70s-80s of the past century as a result of studying artificial intelligence methods. However, it is only lately that they have been paid great attention due to rapidly increasing necessity for data processing.

The **topicality** of the research theme under consideration is caused by the fact that in data mining industry the results of regularity search are mostly represented using conditional rules in the form of *IF-THEN* statements. Rules of that kind help solve tasks of forecasting, classification, pattern recognition, etc. By employing various approaches: clustering algorithms, neural network methods, fuzzy rule processing methods, one can obtain the rules that characterise data in a form clear to humans, which enables the interpretation of data, finding regularities in them as well extracting new data characterising rules from the data.

The main **goal** of the research is to study task classes that employ methods of regularity obtaining and conditional rule construction and to increase the effectiveness of intelligent data processing systems with the aim of finding regularities or rules that characterise them, from multidimensional data samples.

In accordance with the goal stated, the following tasks were specified:

1) Estimate the present state-of-the-art in artificial neural network application to discovering regularities;
2) Study and compare application possibilities of classical clustering algorithms in modelling regularities;
3) Develop a fuzzy rule implementation model for finding regularities by fuzzy clustering methods;
4) Adapt architectures, methods and algorithms that enable using certain neural network classes to extract and process rules;
5) Perform data analysis using association rule extraction methods.

**The object of the research** is various kinds of regularities in the data that can be expressed using conditional rules.

**The subject of the research** is methodologies of data analysis which are employed to obtain regularities and/or rules of that kind.

**Research methods**

The research accomplished to achieve the outlined goal includes:

(1) application of artificial neural network methods to analyse the data obtained;

(2) using of clustering methods in classification tasks and in neural networks;

(3)application of fuzzy clustering to fuzzy rule processing;

(4)employing of algorithms of association rule extraction.

To solve the tasks under study, artificial neural network theory, algorithms of clustering methods, elements of fuzzy set theory and of association rule theory are employed. Applications of the research methods are illustrated with demos and case studies that characterise the class of tasks under examination.

As scientific and practical interest in data mining was increasing, research on the possibility of processing IF-THEN rules with the help of different techniques achieved worthy attention Thereby a knowledge base model is being obtained that can serve as a basis for further data analysis While developing the doctoral work, research was performed and different methods and techniques were examined, which helped obtain regularities that characterised the data

**The novelty** of the work under consideration is the summarised information about choosing the most rational data mining method that is validated through experiments. That helps select one of the three rule extraction methods being considered.

The study describes an idea of potential function method application in particular task class.

Within the study, a method is developed to extract fuzzy rule base from the numerical data using the fuzzy clustering algorithm.

**The practical value** consists in developing a methodology and software that enable using data analysis methods to process regularities. The software is written using the *Matlab* environment. It is used to perform experiments aimed at illustrating the execution of different data analysis methods. Programme modules and data employed are available on the author's home page at www.ru.lv/~peter.

In the course of work, the following research was accomplished and these methods were elaborated:

1) the suitableness of neural network application in searching for regularities is studied and analysed;
2) a procedure for rule extraction from a radial basis function (RBF) network is developed;
3) the choice of clustering method for studying regularities is validated. Several clustering methods are compared and analysed with the aim of finding the most effective one;
4) a fuzzy clustering model for rule extraction and processing is developed and implemented;
5) principles and methodology for rule extraction, using association rule discovery methods, are worked out.

**Major findings** of the work were reported at these significant international scientific conferences:

1. *Grabusts P. Extracting rules from trained RBF neural networks// Proc. of the 5th International Conference. - Rezekne: RA izdevniecība, 2005. -p. 33-39.*
2. *Grabusts P. Analysing Bankruptcy Data with Neural Networks// Proc. Iff International Conference on Soft Computing. -Brno, Czech Republic, 2004. -p. 111-117.*
3. *Grabusts P. Using Association Rules to Extract Regularities from Data// Proc. 6 International Baltic Conference on Data Bases and Information Systems. -Riga, 2004. - p. 117-126.*
4. *Grabust P., Borisov A. Using grid-clustering methods in data classification//Proceedings of the International Conference on Parallel Computing & Electrical Engineering-PARELEC'2002. - Warsaw, Poland, 2002. -p. 425-426.*

5. *Grabusts P., Borisovs A. RBF neironu tīklu pielietošanas perspektīvas// 11 Pasaules latviešu zinātnieku kongresa tēžu krājums. - Rīga, 2001. - 5 70. lpp*

6. *Grabusts P. Klaslerizācijas metodes izmantošana RBF neironu tīklos// III Starptautiskās zinātniskās konferences „Vide. Tehnoloģija. Resursi" materiāli -Rēzekne: RA izdevniecība, 2001. - 257.-262. lpp.*

7. *Grabust P. Using a thermal equilibrium method in the neural networks// International Conference on Parallel Computing & Electrical Engineering-PARELEC '98. - Bialystok, Poland, 1998. -p. 261-263.*

The results of the author's work are included in the 2002 report on the execution of Latvian Science Council's Grant (Decision No. 3-2-1 of the LSC Doctoral Grant Commission dated May 13, 2002 about doctoral grant awarding).

There are four publications and nine papers published in the proceedings of scientific conference related to the theme of the doctoral research.

**The structure and volume of doctoral thesis**

The work includes introduction, five chapters, conclusion, a list of references, and Appendix. The doctoral thesis contains 120 pages and is illustrated with 91 figures and 41 tables. The list of references consists of 68 titles.

Introduction validates topicality of the research performed, states the main research goal and tasks as well as provides a brief characterisation of principal research areas.

Chapter 1 presents a survey of ways and objectives of data analysis in the context of searching for regularities.

Chapter 2 describes application of association analysis to the process of obtaining regularities as well as a series of experiments on statistical data processing.

Chapter 3 examines potentialities of artificial neural networks as applied to performing data analysis; describes experimental results related to bankruptcy data analysis and suggests using the potential function method in data analysis.

Chapter 4 estimates possibilities of different clustering methods in data analysis, elaborates a methodology for rule base extraction with the help of fuzzy clustering and describes the experiments conducted to obtain the rules from the multidimensional data samples.

Chapter 5 considers neural network methods as applied to the process of conditional rule extraction and presents experimental results - the rules obtained from the bankruptcy data sample.

Conclusion provides a summary of the research conducted and outlines an area of future research.

Appendix demonstrates the author's developed software for association rule extraction from statistical data.

## CHAPTER OVERVIEW

**Introduction** validates topicality of the research performed, states main research goals, provides a short characterisation of principal research areas, as well as presents annotation of the chapters.

**Chapter 1** presents a survey of the existing kinds and objectives of data analysis in the context of searching for regularities. Nowadays the concept *of regularity* is acquiring more and more attention in the representation of intelligent data processing system operation. In many cases it is necessary to ascertain in what manner the data are interrelated, how various data differ or agree with each other, and what the measure of their comparison is. In various dictionaries the term *regularity* is interpreted as similarity, conformity with a law or conclusion by analogy. Regularity can be considered determined correctly if it explains the results of all experiments that relate to the given area of operation. The more experimental facts confirm, the more reliable the validation is. As the objective of searching for regularities in the data is a task of intelligent-data analysis, it is necessary to discuss today's approaches in that area. Currently the following concepts of data storing and analysis can be distinguished:

- Data Warehouse;
- On - Line Analytical Processing,
- Data Mining.

The area of intelligent data analysis (IDA) is considered as very important because the data stored are defined as *knowledge.* In English, IDA can be interpreted as *Knowledge Discovery in Databases* (KDD) and *Data Mining* (DM). Many researches use both of the terms as synonyms, whereas the majority of researches treat KDD in a wider sense, as a research area that comprises artificial intelligence methods, a combination of statistical and database methods that enables obtaining information from the data. In its turn, DM is treated as the process of acquiring that information. Thus, IDA can be defined as follows. *This is data analysis consisting in searching for hidden regularities in the data.* It should be noted that the majority of IDA methods were developed in the seventies and eighties of the past century as a result of researches on artificial intelligence methods. They, however, have attracted wide attention only nowadays due to rapidly growing necessity for data processing.

Majority of researchers distinguish the following global data mining tasks:

1. Classification. The most widespread task of IDA. It allows one to distinguish features that characterise object groups (classes) of one type so that a new object could be ascribed to the class according to the known characteristics.
2. Clustering. It logically continues the idea of classification in a more complex case when the classes are not specified. As a result of clustering, object distribution by classes is also performed. In general, clustering results are rather subjective, which depends on the similarity measure of the selected training set. Clustering algorithms are used in several artificial neural network models as well as fuzzy network models for data preprocessing in unsupervised learning.
3. Association rules. Association ruies enable finding regularities among several related events. Rules of that kind are based on the following statement. If event A occurs, then event B will also happen with probability X%. In the beginning, that task was employed in supermarkets to determine what products customers buy together. That is.why it is sometimes called market basket analysis.
4. Sequence. It is similar to associations, but sequence is valid for the events that do not begin at the same time. Instead, they happen with a certain time shift. If event A happens, then event B will happen after time T with probability X%.
5. Prediction. Using actual values to predict future values.

Researches conducted in the area of intelligent data analysis use a lot of different methods borrowed from several sciences. It is often emphasised that IDA comprises points of interaction of different science fields. Among most frequently cited are: decision analysis and regression trees, Bayesian classifier, artificial neural networks, fuzzy inference, association rules, clustering, fuzzy cluster analysis, etc.

One myth about the omnipotence of intelligent data analysis methods is that IDA finds regularities in the data automatically. Indeed, various IDA methods allow one to find regularities in the data examined, whereas the objective of the research process - what final result is required to achieve - must be stated clearly. To successfully perform data analysis, the following six steps should be taken:

1. determination of a specific objective;
2. data collection;
3. choice of methods of analysis;
4. choice of software;
5. performance of the analysis, and
6. making a decision how to use the result.

The main requirement which is put forward to the results of data analysis is that the results must always be interpreted as correctly as possible. The rules that represent the regularities found have to be stated as simple and easy to understand logical expressions. Namely, they must look as these logical rules:

IF {(Event 1) AND (Event 2) AND ... (Event N)} THEN ....

In what follows, the author will employ logical conditional rules (production rules) of this kind:

$$IF\ \underbrace{(Antecedent\ 1)\ and\ (Antecedent\ 2)\ and \ldots (Antecedent\ N)}_{A}\ THEN\ \underbrace{(Consequent)}_{B}$$

In the course of working, the need arose to study techniques enabling conditional rule extraction from the multidimensional data samples The scheme shown in Figure 1 was made as a result of a thorough study of the relevant literature. It represents the trends that nowadays dominate in the area of intelligent data analysis.

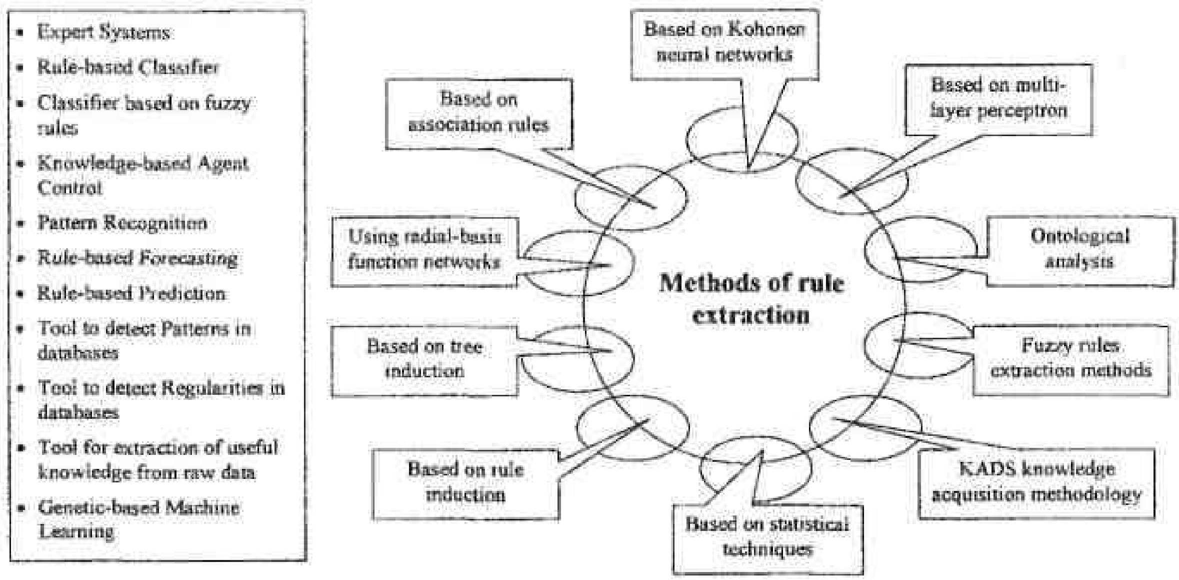



Figure 1. Rule extraction methods and their applications

So long as research interests of the author are related to artificial neural networks and clustering, the choice of RBF neural network based rule extraction methods is validated in the study because networks of that kind make use of clustering at the learning stage.

In its turn, fuzzy clustering can be employed at the initial stage of fuzzy rule formation, which will be described later.

As regards association rule extraction method, it is highly popular in the intelligent data analysis. Due to that, a desire to examine its application area arose.

The methods and tasks considered in this study as well as their structure and respective chapter numeration are represented in Figure 2.

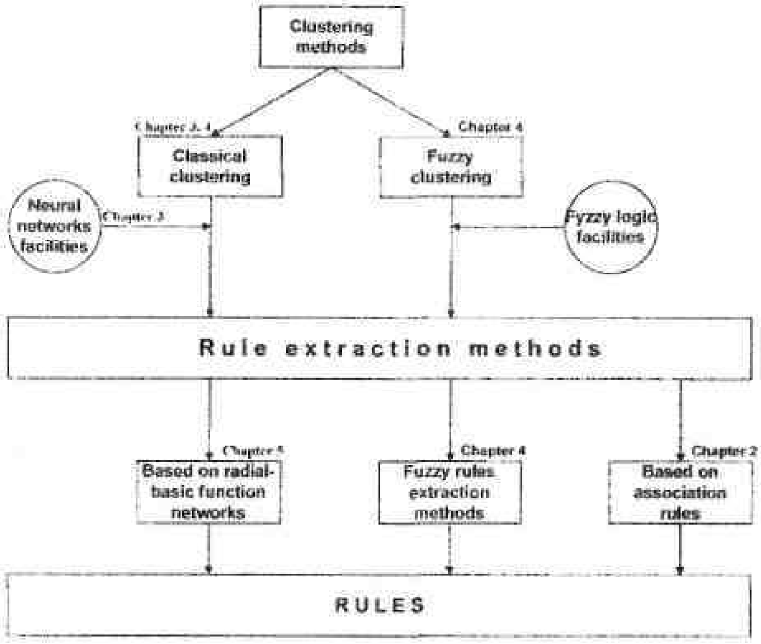



Figure 2. Rule extraction methods considered in the study

The motivation for studying rule extraction methods is the following:

- the amount of multidimensional data to be analysed becomes too large for the potentialities of statistical analysis;
- popular neural network methods operate by the black box principle that complicates interpretation of the results for the user;
- previously unknown regularities are present in the data;
- the regularities found can be represented in a way that is easy to perceive and understand for the user.

**Chapter 2** discusses application of association analysis in the process of obtaining regularities. Section 1 formulates the task of association analysis. Nowadays a great amount of data are accumulated in different areas of science, business, health care etc., so the necessity to analyse the data in order to better manage the area concerned, arises. Frequently the needs of business stimulate the development of new methods of intelligent database analysts, which are oriented towards practical business applications. As an example, one of the problems can be mentioned, which shop managers often face: when a customer purchases specific article, then X % of the time he also buys another article that is first article dependent. Say, if he buys bread and butter, then 90% of the time he also buys milk. Initially, that task was used to find a pattern of typical market basket in supermarkets. That is why it is frequently called *market basket analysis.* In general case these operations arc called *transactions.* The regularities, which could evidence for such events' relationships, are called *associations.* Associations or association rules enable one to find relationships among several dependant events. The underlying statement for those rules is the following: if event A occurs, then event B will also occur with probability X%.

Association rule mining is based on theoretical assumptions about the existence of rules of that kind developed by a group of researchers in 1993. In 1994 an effective algorithm for association rule mining was published. These studies have stimulated the development of numerous similar algorithms, which made it possible to analyse, for example, large scale shopping operations, and extend the task to one of the fundamental methods of intelligent data analysis. Association rules can be employed not only for market basket analysis. They can be applied to any data analysis by carefully examining the regularities found.

Section 2 describes two characteristics which define association rules - the confidence and the support. All rules that are specified in the form IF (X) THEN (Y), have these two characteristics.

1) confidence - part of instances when the rule was satisfied out of all its application instances (part of instances Y in relation to instances X);
2) support - part of instances when the rule was satisfied out of all the instances when Y was satisfied (part of instances X in relation to instances Y).

Let I = $\{i_1, i_2,..., i_m\}$ be a set of literals, called *items*. Let D = $\{t_1, t_2,..., t_n\}$ be a set of transactions, where each transaction, t, is a set of items such that $t \subseteq I$. Each transaction is associated with an identifier, called TID. Given an itemset $X \subseteq I$, a transaction t contains X if, and only if, $X \subseteq t$. The itemset X has *support*, s, in the transaction set D if s% of transactions in D contain X; we denote s = support (X).

An association rule is the implication in the form X => Y, where X, Y $\subset$ I, and $X \cap Y=0$. Each rule has two measures of value: support and confidence. The support of the rule X => Y is support $(X \cup Y)$:

$$s(X) = \frac{|\{T \in D | X \subseteq T\}|}{|D|}.$$

The *confidence*, c, of the rule X => Y in the transaction set D means that c% of transactions in D that contain X also contain Y, which can be written as the ratio:

$$c(XY) = \frac{s(X \cup Y)}{s(X)} \tag{1}$$

Support indicates the frequencies of the occurring patterns, and confidence denotes the strength of implication in the rule. Given a user specified minimum support (called minsup) and minimum confidence (called minconf), the problem of association rule mining is to find all association rules where support and confidence are larger than the user defined minsup and minconf.

Let us illustrate the aforementioned with an example. Table 1 represents transaction database. It is a set of products I that consists of products A, B, C, and D.

Table 1

Market TID

| TID | Market basket | TID | Market basket |
|---|---|---|---|
| 1 | {A,C} | 6 | {A,B} |
| 2 | {B} | 7 | {A,D} |
| 3 | {A,B,C,D} | 8 | {B,C,D} |
| 4 | {B,D} | 9 | !C,D} |
| 5 | {A,B,D} | 10 | {A,B,D} |

Each row of the table contains transaction identifier TID, which characterises the number of customer's performed operation as well as a set of the goods purchased. The support of itemset {A,B} is 0.4. The value of support of itemset (A, B, D} is 0.3. Hence, the confidence of rule {A,D} => {B} is 0.75. From (1) it follows that:

$$c(A, D \Rightarrow B) = \frac{s(A \cup B \cup D)}{s(B)} = \frac{0.3}{0.4} = 0.75.$$

If the boundary value of support, *minsup,* is less or equal to 0.3 but the boundary value of confidence, *minconf*, is less or equal to 0.75, the rule is considered acceptable. It can be concluded that this association rule is derived: "When a customer buys products A and D, it is possible that in 75% of cases he will buy product B as well".

Statements of that kind are without doubt acceptable only for large databases. Support and confidence values do not yet guarantee the suitability of the rule for modelling customer's behaviour. They can only assist in making decisions.

Sections 3 and 4 examine two algorithms for finding itemsets, *Apriori* and *AprioriTID*, whose execution also helps to derive association rules. In general case, the process of all association rule extraction is reduced to two subproblems:

1. The large itemsets. Find all itemsets that have support above the predetermined minimum support. These itemsets are called large itemsets. Sometimes they are also called frequent itemsets.
2. For each large itemset, derive all rules that have a greater than the predetermined minimum confidence as follows. For a large itemset X and Y, where $X, Y \subset 1$, and $X, Y \cap = 0$, if support $(X \cup Y)$ / support $(X) \geq$ minimum-confidence, then the rule X => Y is derived.

For transaction database D = {{A,C,D}, {B,C,E}, {A,B,C,E},{B,E}} with minimal support boundary value equal to 0.5, itemsets $L_k$, obtained as a result of *the Apriori* algorithm execution, and candidate sets $C_k$ after three iterations are shown in Figure 3.

$L_1$

| Itemset | Count |
|---|---|
| {A} | 2 |
| {B} | 3 |
| {C} | 3 |
| {E} | 3 |

$C_2$

| Itemset |
|---|
| {A,B} |
| {A,C} |
| {A,E} |
| {B,C} |
| {B,E} |
| {C,E} |

$L_2$

| Itemset | Count |
|---|---|
| {A,C} | 2 |
| {B,C} | 2 |
| {B,E} | 2 |
| {C,E} | 2 |

$C_3$

| Itemset |
|---|
| {B,C,E} |

$L_3$

| Itemset | Count |
|---|---|
| {B,C,E} | 2 |

Figure 3. Execution example ot the *Apriori* algorithm

The *AprioriTID* algorithm is similar to the *Apriori* algorithm and employs the recall function to generate candidate sets. The process of candidate set generation is illustrated with a transaction database shown in Figure 4 under the minimal support value 2.

Database

| TID | Elements |
|---|---|
| 100 | 1 3 4 |
| 200 | 2 3 5 |
| 300 | 1 2 3 5 |
| 400 | 2 5 |

$\bar{C}_1$

| TID | Itemsets |
|---|---|
| 100 | {{1},{3},{4}} |
| 200 | {{2},{3},{5}} |
| 300 | {{1},{2},{3},{5}} |
| 400 | {{2},{5}} |

$L_1$

| Itemset | Support |
|---|---|
| {1} | 2 |
| {2} | 3 |
| {3} | 3 |
| {5} | 3 |

$C_2$

| Itemset | Support |
|---|---|
| {1 2} | 1 |
| {1 3} | 2 |
| {1 5} | 1 |
| {2 3} | 2 |
| {2 5} | 3 |
| {3 5} | 2 |

$\bar{C}_2$

| TID | Itemsets |
|---|---|
| 100 | {{1 3}} |
| 200 | {{2 3},{2 5},{3 5}} |
| 300 | {{1 2},{1 3},{1 5}}<br>{{2 3},{2 5},{3 5}} |
| 400 | {{2 5}} |

$L_2$

| Itemset | Support |
|---|---|
| {1 3} | 2 |
| {2 3} | 2 |
| {2 5} | 3 |
| {3 5} | 2 |

$C_3$

| Itemset | Support |
|---|---|
| {2 3 5} | 2 |

$\bar{C}_3$

| TID | Itemsets |
|---|---|
| 200 | {{2 3 5}} |
| 300 | {{2 3 5}} |

$L_3$

| Itemset | Support |
|---|---|
| {2 3 5} | 2 |

Figure 4. Execution example of the *AprioriTID* algorithm

Section 5 describes a series of experiments aimed at discovering Independence of the count of extracted rules on the initial values of support. The main motivations for the experiments was the desire to find regularities in raw data using association rule mining method. The author did not have an opportunity to use professional software packages such as the data mining tool *Clementine* or similar, therefore the experimental part was carried out in the *Matlab* environment.

As experimental data, the data of the Latvian Central Statistics Office about reply variants obtained from 3044 respondents were used. The data selected for the experiments were related to the study of inhabitant migration process. The respondents were asked the following questions:

1. In what country were you born? (with 11 possible reply options offered);
2. How long have you lived in this place? (with 4 reply options offered);
3. Where did you live before moving to this place? (with 3 reply options provided);
4. Please designate the type of the place you lived in before moving to the current place, (with 7 possible reply variants);
5. What was the reason for you to move to the current place? (with 6 possible replies);
6. Are you planning to move to another place within the next 3 years? (with 5 possible reply options provided).

The objective of the experiment was to determine possible relationships in these data and to determine the dependence of the count of regularities on the preliminarily assigned boundary values of support and confidence.

In the first part of the experiment it was assumed that the confidence was minconf = 95 and the support was minsup = 95. See Figure 5 for the obtained graphs of rule support analysis and rule confidence analysis. As many as 58 rules were derived at these initial values. For each rule there were calculated support value and confidence value.

Below one can find some derived rules with greatest support values;

**12 74** $\Rightarrow$ **51** Support=592 and confidence is 98;

**46** $\Rightarrow$ **83** Support is 545 and confidence is 97;

**12 74 83** $\Rightarrow$ **51** Support is 530 and confidence is 98;

**12 46** $\Rightarrow$ **83** Support is 443 and confidence is 97;

**45 51 65** $\Rightarrow$ 83 Support is 303 and confidence is 95.

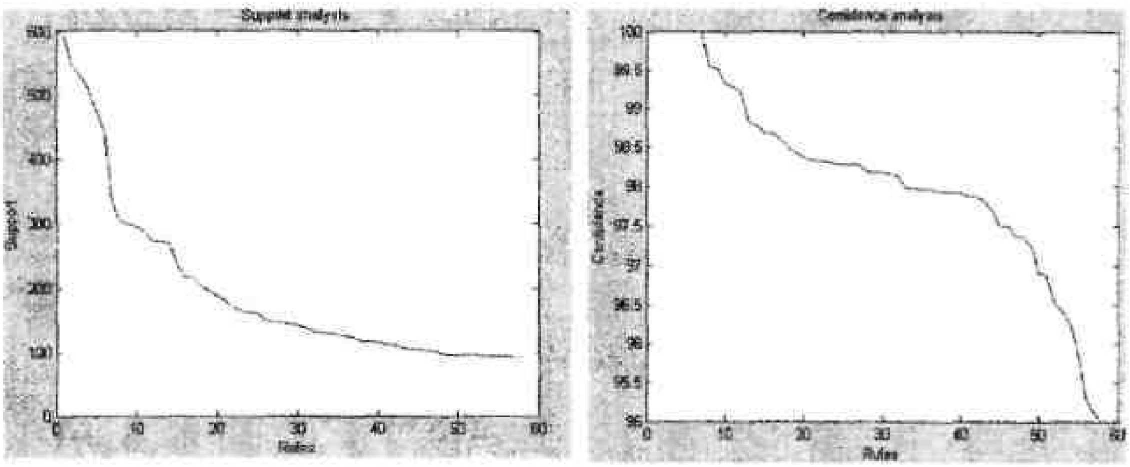



Figure 5. Dependence of rule count on support and confidence values

Taking into account the decoding results one can conclude that:

-the 1st rule determines: IF "you were born in Latvia" AND "moved to the current place due to family reasons" THEN "Before moving to the current place you lived in Latvia".

-the 2nd rule determines: IF "You have lived in this place ail the time" THEN "In the next 3 years you are not planning to move to any other place".

-the 3rd rule determines: IF "You were born in Latvia" AND "You have moved to this place due to family reasons" AND "You are not planning to move to any other place within the next 3 years" THEN "Before moving to the current place you lived in Latvia"

-the 4th rule determines that: IF "You were born in Latvia" AND "You have lived in this place all the time" THEN "You are not planning to move to any other living place within the next 3 years"
-the 5th rule determines that: IF "You lived in this place up to the age of 50 years" AND "Before moving to this place you lived in Latvia" AND "Before that you lived in a village" THEN "You are not planning to move to any other living place within the next 3 years".

By analysing the rules derived one can state that the persons interrogated live in a small compact living place with a small migration trend. The rules derived are logically understandable and represent a real-world situation.

In the second part of the experiments, rule count values were obtained at different support values and fixed boundary confidence values. The results are shown in Table 2.

Table 2

Dependence of rule count on support values

| Support | 10 | 15 | 20 | 25 | 50 | 75 | 100 |
|---|---|---|---|---|---|---|---|
| Conf.-50 | 2690 | 2071 | 1628 | 1346 | 765 | 509 | 386 |
| Conf.-75 | 1717 | 1343 | 1029 | 839 | 473 | 318 | 241 |
| Conf.-90 | 983 | 736 | 553 | 436 | 246 | 159 | 121 |

The graphic form of the above correspondence can be seen in Figure 6.

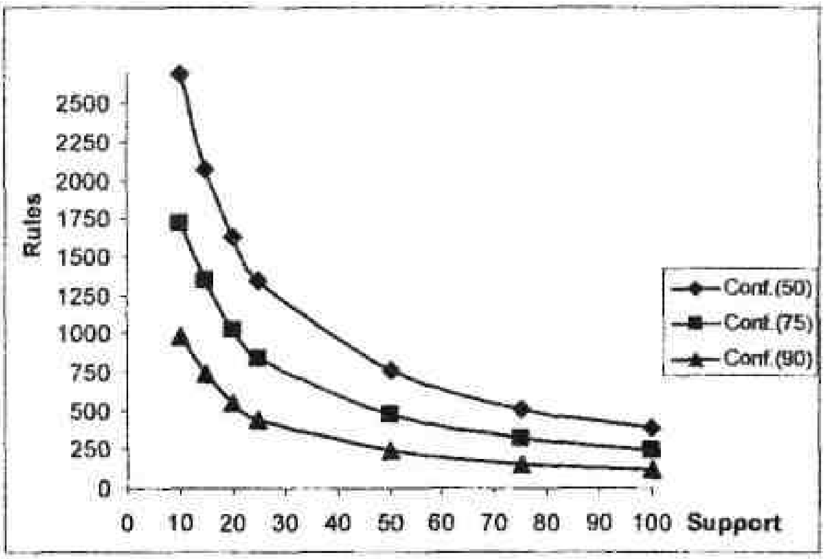



Figure 6. Dependence of rule count on support values

Based on the table data and graphically given correspondence, one can conclude that the greater the assigned confidence level and support's boundary value, the smaller the number of association rules discovered and, as a result, the stronger the rules.

Section 6 concludes that for certain classes of tasks the mechanism of association rule mining is very similar. It is important to realise that to be applied successfully, the derived association rules require a thorough analysis. The method of association rule mining, which was initially intended for the market basket analysis, proved to be a good tool for a wide range of tasks. With the help of that method, it is possible to mine and discover regularities in the form X => Y in different data types Nowadays the method has a widespread application in tasks of large database processing and analysis. Association rule mining method justly lies among the main intelligent data processing methods.

At the same time it should be noted that association rule analysis has its own bottlenecks whose study could turn out to be a valuable application area:

-Software implementation of association rules requires considerable time;
-The analysed data should be possibly homogeneous.
-Unfortunately, erroneous or strange data also participate in rule formation.

Chapter 3 discusses potentialities of artificial neural networks in performing data analysis.

The task of Section 1 is to examine potentialities of neural networks as applied to searching for regularities. Neural networks (this is a general notation of a certain class of algorithms) are able to learn from examples thus deriving hidden regularities in the data. If a relationship among the input and output data exists (even if traditional methods do not show it), a neural network can adjust to it automatically with certain accuracy. Moreover, many neural networks allow one to evaluate the importance of particular features of the input data, to reduce data amount without loosing the most essential data, to identify the approaching of the critical situation. In many cases neural networks help find regularities which cannot practically be discovered by analysing the data manually.

Section 2 describes the basic principles of neural network operation. A definition of a neuron and neural network is given. A number of activation function formulas are described, and a classification scheme of neural networks is shown. In particular sections of the work, three neural network models will be employed: a radial basis function network (RBF), multilayer network and the Boltzmann machine.

Section 3 studies potentialities of neural network methods in searching for regularities. The research is based on the RBF neural network, whose architecture is shown in Figure 7.

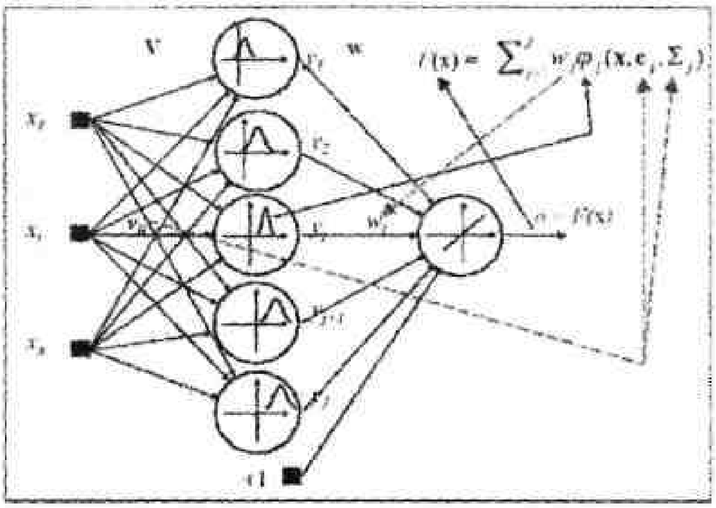


Figure 7. Architecture of the radial basis activation function network

In general, the RBF network contains a single hidden layer of units with radial activation functions. The most frequently used radial function in networks of that kind is Gaussian:

$$\Phi(x) = e^{-\frac{\|x_i - c_i\|}{\sigma^2}},$$

where $x_i$ - a component of the input vector,
$c_i$ - the centre of the activation function;
$\sigma$ - standard deviation.

The number of units of the hidden layer is determined in the course of learning. Normally, each unit of the hidden layer corresponds to a specific class of objects. For clearness it can be said that units of the hidden layer calculate the Euclidean distance between the input objects and the centre of the radial function. The output values of the hidden layer units serve as input signals for the output layer units. The network output is the weighed sum of input signals of the output unit.

The training of RBFs takes place in two stages (see Figure 8) That learning algorithm has been selected because it is the principal training method for neural networks of the above mentioned kind.

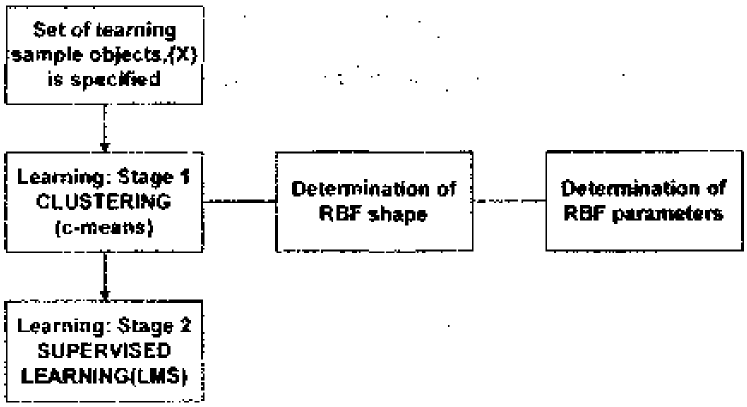



Figure 8. Learning scheme of RBF networks

At Stage 1, training in the hidden layer is performed with unsupervised learning algorithms which are called clustering algorithms. The aim of clustering is to divide the input data into groups of objects (clusters, classes, and taxons) and determine the centres of those objects. The essence of clustering is represented in Figure 9.

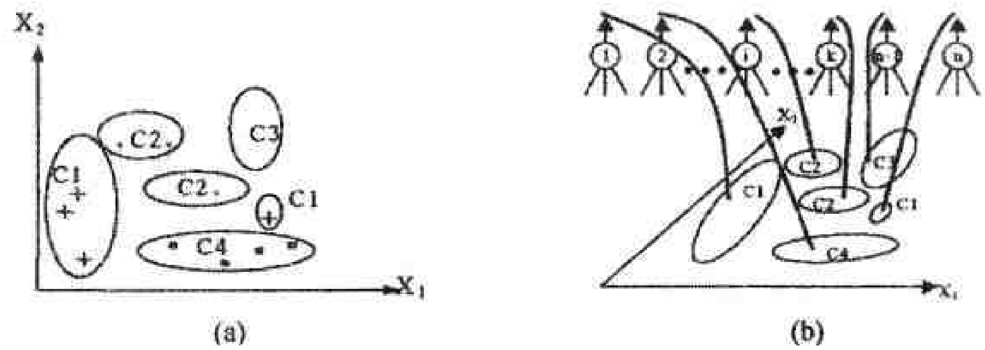



Figure 9. (a) splitting two-dimensional objects into clusters; (b) cluster representation in RBF units

The process of cluster analysis includes these stages:

- Formation of a set of data subject to analysis;
- Determination of limits or other characteristics defining this class of data (cluster);
- Splitting the initial data into clusters;
- Construction of the cluster hierarchy.

RBF training was accomplished using the *C-means* clustering algorithm. It minimises the quality indicator which is determined as the sum of squared distances of all points belonging to cluster space, from the cluster centre The name of the procedure has resulted from the calculation of the mean distances from the cluster centre, inside the cluster group. The algorithm is executed as follows:

(1) Initialise the cluster centres $w_i$ (set the initial cluster centres to the first *m* training data or to the *m* randomly chosen training data)

(2) Group all patterns with the closest cluster centre i.e. for each pattern $x_i$, assign $x_i$, to group $j^*$, where

$$\left\| x_i - w_{j*} \right\| = \min_j \left\| x_i - w_j \right\|$$

(3) Compute the sample mean for the cluster centre i.e. for each group wj,

$$\sigma_j^2 = \frac{1}{M_j} \sum_{x_i \in groupj} x_i$$ , where $m_j$, is the number of patterns in group j.

(4) Repeat by going to step (2), until no change in cluster assignments.

As a result of the algorithm execution, the final cluster centres, $w_j$, are determined with the restriction that the sum of squared distances of all the points, which belong to the group, from the cluster centre, has to be minimal.

Training of the hidden layer results in finding activation function's parameters They are cluster centres $w_j^h$ and cluster standard deviation $\sigma_j^2$, j being the count of clusters. The value of $\sigma_j^2$ is determined by formula

$$\sigma_j^2 = \frac{1}{M_j} \sum_{x \in \Theta_j} x_i (x - w_j^h),$$

where $\Theta_j$ - number of objects in the training set that are grouped around the cluster centre $w_j^h$;

$M_j$ - number of objects in $\Theta_j$, and T - notation of the transposed matrix or vector.

The advantage of the *C-means* clustering algorithm consists in its popularity, high effectiveness, and simple implementation. However, the algorithm may not achieve good results if object distribution is not uniform. In this case the parameters (the number of cluster centres) have to be changed, after that the algorithm must be executed again. As a disadvantage of the algorithm its non-universality can be mentioned.

After the training in the hidden layer is completed, and radial function's parameters are defined, training in the output network layer is performed using the so-called supervised learning. To that end, the least mean square error algorithm (LMS) is employed.

Training data consist of pairs {Uk/dk} where Uk is the input object for *the* output layer whereas dk are network target outputs, and k is the number of objects. The input signal for the output layer is processed in the hidden layer before it is forwarded to the outpur layer. In other words, the input values of the output unit are the hidden layer output values $u_k$i (radial finction values), where i=l,...,M. They are calculated by formula:

$$u_i = \exp\left[ -\frac{(x - w_{ji}^h)^T (x - w_{ji}^h)}{2\sigma^2} \right],$$

where x - input vector;

$w_j^k$ - the weight vector of the j-th unit (actually, this is the cluster centre to which the given input vector belongs);

$\sigma_j^2$ - i-th unit standard deviation;

M - units of the hidden layer;

N - number of units in the output layer.

The output $y_k$ of the output layer for the k-th input vector is calculated as follows:

$$y_k = w_i^0 u_i ,$$

where $w_i^\circ$ - the weight vector that connects the i-th unit of the hidden layer with the output unit;

$u_i$ - input signal for the output unit.

In the course of learning, taking into consideration both real and target output values (respectively, $y_k$ and $d_k$), neural network mean square error is calculated by formula:

$$E_k = \frac{1}{2} \sum_{k=1}^{K} (d_k - y_k)^2 = \frac{1}{2} \left\| d_k - y_k \right\|^2$$

A certain value of minimal error, $E_{min}$, is used as the stopping criterion. If the mean square error of the training cycle (iteration) is equal to or less than $E_{min}$, training is considered completed. Due to that, after the second training stage is performed, the RBF network can be considered trained and ready for further research experiments.

Section 3 then examines another neural network kind - the Boltzmann Machine, which is mostly used in optimisation tasks. Learning algorithm of the Boltzmann Machine is analysed, and the essence of the Simulated Annealing *(SA)* method is discussed

Section 4 discusses neural network application possibilities. In the first example, using a small training set the procedure of RBF network learning is described in detail for the pattern recognition task, in the second example, the SA method is applied to solve the classical Travelling Salesman Problem The shortest paths between Latvian towns are calculated (TSP-26).

Section 5 deals with practical task solving - bankruptcy data analysis. The objective of the task is to determine the financial state of a company (bankrupt or non-bankrupt).

Bankruptcy diagnostics is a directed financial analysis system whose area is crisis situation control at the enterprises. In the analysis of the general financial situation of the company a separate group of financial ratios is formed, using which it is possible to reason about the threat of bankruptcy.

Two basic approaches to bankruptcy prediction are commonly distinguished. The first approach is based on financial data and comprises working with different ratios including extremely popular Altaian's Z-score, Taffler's coefficients, etc., as well as "an ability to read the balance sheet". The second approach uses the data on bankrupt companies that were then compared to the data of the company under consideration.

The aim of the experiment was to compare different techniques used in the bankrupt data analysis and evaluate the results. For the purpose of experiments, balance sheet data of 63 companies were used (46 - bankruptcy and 17 - non-bankruptcy). It was decided to calculate the following financial ratios on the basis of the data available and further use them in all the experiments:

- R2: Cash Flow / Current Liabilities;
- R3: Cash Flow / Total Assets;
- R7: Current Assets / Current Liabilities;
- R9: Current Assets / Total Assets;
- R31: Working capital / Total assets.

The first employed method was the multivariate discriminant analysis *(MDA).* To accomplish the MDA, the SPSS statistical package was used. The results of discriminant analysis are shown in Table 3.

Table 3

Discriminant analysis classification results

| | | | Predicted Group Membership | | Total |
|---|---|---|---|---|---|
| | | Bankrupt | 0 | 1 | |
| Original | Count | 0 | 39 | 7 | 46 |
| | | 1 | 4 | 13 | 17 |
| | % | 0 | 84.8 | 15.2 | 100.0 |
| | | 1 | 23.5 | 76.5 | 100.0 |

As a result, 82.5% of original grouped cases were classified correctly. Misclassified cases were: 14, 26,28,35, 36, 37,41, 58, 59, 60, and 62.

For the second approach, the author suggested using of one of pattern recognition methods - the method of potential functions - to test the assumption whether this method can be applied in bankruptcy diagnostics tasks. The algorithm of the potential function method is based on the hypothesis about the function character that separates sets according to different classes of patterns. For the purpose of experiments, this function was used as a potential function:

$$\varphi(R) = -\frac{\lambda}{1+\alpha R^2}$$

where α - learning parameter;

R - the distance of the point where the potential is calculated from the point of learning set;

$\lambda$- the value of potential that is assigned to the point in the process of learning (weight).

In the first part of the experiment, the effect of learning parameter on the duration of learning (epochs) was examined. As a learning set, the bankruptcy data were used. A sample of bankruptcy data on the same company served as a test data. Learning parameter $\alpha$ varied within the limits [0.1, 2.0]. Learning results are given in Table 4 and Figure 10.

Table 4

Experimental results (learning parameter a and number of epochs)

| α | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 | 1.1 | 1.2 | 1.3 | 1,4 | 1.5 | 1.6 | 1.7 | 1.8 | 1.9 | 2.0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Ep.** | 5203 | 1477 | 674 | 371 | 238 | 166 | 173 | 142 | 117 | 91 | 64 | 84 | 60 | 59 | 62 | 36 | 49 | 30 | 25 | 25 |

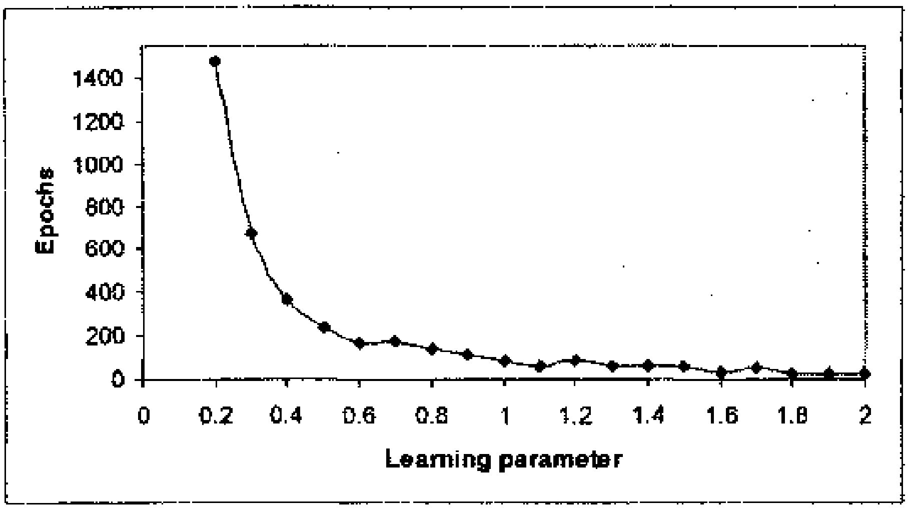



Figure 10. Graph of parameter dependence on the number of epochs

By analysing the results of learning, one can conclude that the algorithm converges at all the a values assigned but the number of learning cycles goes down as a value increases.

In the second part of the experiment, testing and analysis of learning algorithm execution depending on the value of potential, λ, were performed. It was found that after the learning the points of the given test set were identified correctly, i.e., λ values calculated in the course of learning enabled one to correctly determine the input data class: bankrupt or non-bankrupt. However, taking into account that the MDA method correctly classifies the input data in 82.5% cases only,a question arises-which points of the learning data faced identification problems? After the application of potential functions in the algorithm,the initial λ values are equal to 0.If a point is not classified correctly during learning, the value of that parameter, λ, is increased by 1. As a result of the experiment it was found that in bankruptcy data set there were 7 points for which correction of parameter *X* was performed most frequently during learning.

Table 5 represents these points and die number of corrections performed at different a values. Note that at the remaining *a* values the situation does not change essentially. The correspondence is shown graphically in Figure 11.

Table 5

Experimental results (misclassified cases and *X* parameter)

| $\alpha=0.3$ | Cases | $\lambda$ |
|---|---|---|
| | 4 | 249 |
| | 8 | 274 |
| | 14 | 357 |
| | 26 | 393 |
| | 37 | 424 |
| | 50 | 482 |
| | 59 | 104 |

| $\alpha=0.4$ | Cases | $\lambda$ |
|---|---|---|
| | 4 | 148 |
| | 8 | 153 |
| | 14 | 206 |
| | 26 | 212 |
| | 37 | 228 |
| | 50 | 265 |
| | 59 | 60 |

| $\alpha=0.5$ | Cases | $\lambda$ |
|---|---|---|
| | 4 | 92 |
| | 8 | 94 |
| | 14 | 168 |
| | 26 | 112 |
| | 37 | 140 |
| | 50 | 165 |
| | 59 | 39 |

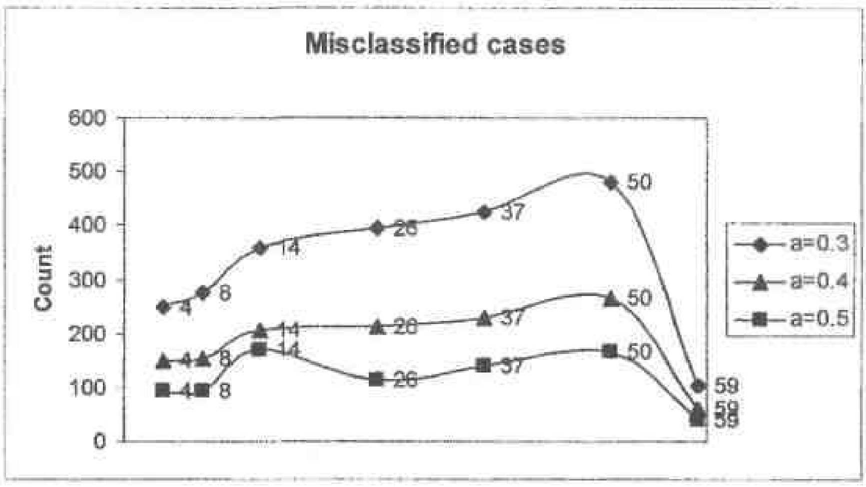



Figure 11. Misclassified cases dependence on the count of parameter *X*

So, for the case of potential functions the misclassified cases at the above bankruptcy data are 4, 8, 14, 26, 37, 50, and 59. It can be seen that the method of potential functions like MDA is unable to correctly classify cases 14, 26, 37 and 59, moreover other points also appear that are different as compared to MDA.

Hence, a conclusion can be made that potential function method can be used for bankruptcy data analysis. However, one has to be careful in interpreting the results. To demonstrate the third method application in bankruptcy data analysis the neural network approach was used. The neural network approach to bankruptcy prediction became extremely popular in the nineties of the past century. Since 1990 research on neural network application in bankrupt analysis has been performed periodically. Odom and Sharda were one of those who first employed neural network techniques in bankruptcy analysis. In the input of the network, Airman's Z-scores about 128 companies were used. It was shown that the neural network approach yielded better results than MDA.

Tarn and Kiang have compared different techniques applied in bankruptcy diagnostics (MDA, LA, ID3, single layer network, and multilayer network) and have shown that in the „one-year-ahead" data the multilayer network was most effective whereas in the "two-year-ahead" data the LA method proved to be most effective.

By analysing the NN network application in bankruptcy analysis, these conclusions were made:

- neural networks ensure approximately 90% accuracy as compared to the 80-85% accuracy of other methods (MDA, LA, and ID3);
- bankruptcy can be predicted several years before it happens, the accuracy of prediction being practically the same for the "one-year-ahead" data and for the "two-year-ahead".

In bankruptcy analysis, preference is mainly given to multilayer network (MLP) with error back propagation learning. Formally speaking, bankruptcy prediction task practically does not differ from that of classical pattern recognition, financial ratios are forvarded to the network input, etwork learning is accomplished, and the hidden units are employed. In the network output there re only two states: bankrupt or non-bankrupt. General scheme of the network that will be used in le experiments from now onwards is shown in Figure 12.

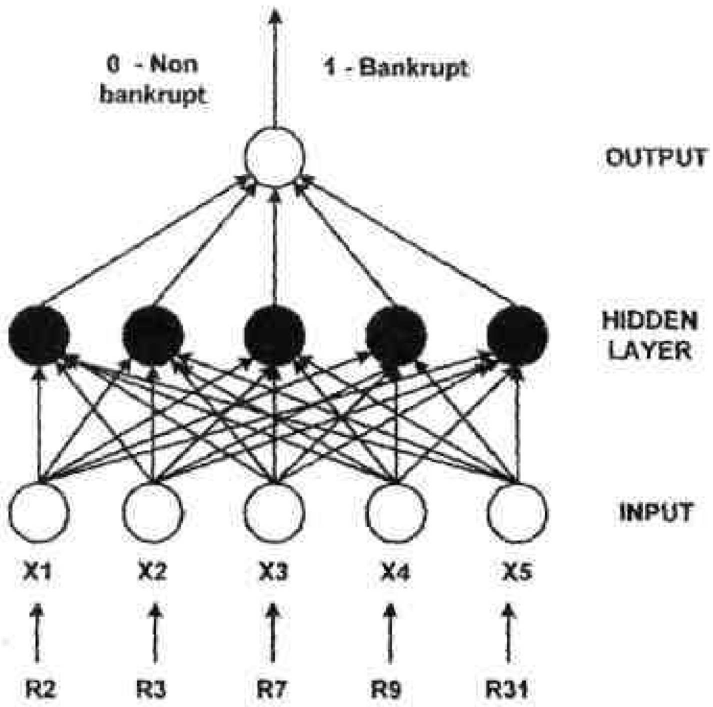



Figure 12. Multilayer feedforward neural network 5-5-1 architecture

The input nodes represent the financial ratios that the model will use to predict bankruptcy. The learning and momentum rates define the rate and accuracy with which the backpropagation algorithm converges on the error minimization solution. Larger numbers for momentum are used to achieve faster convergence, but run the risk of skipping over the optimal solution. The parameters were arrived at after extensive experiments.

In the first part of the experiment, the dependence of the learning quality on o value was tested. The given experimental model has shown that at the a values within 0.8 the following occurs: the algorithm either does not converge or else correct network learning does not happen. Valid learning results were obtained at $\alpha = 0.8$ and $\alpha = 0.9$.

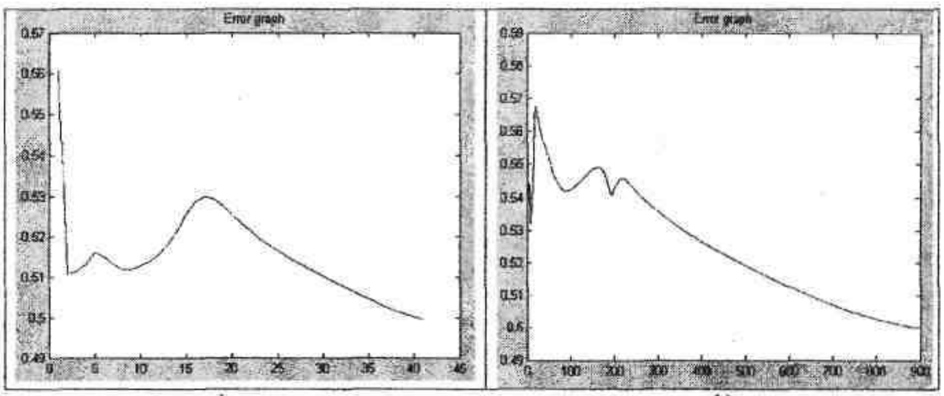
In the second part of the experiment, the dependence of learning quality on the slope of the tanh activation function - $\beta \in$ [0.1; 1] was examined. As a result of the experiments, it was found that acceptable learning quality was achieved at specific $\beta$ values only. Table 6 shows the most significant results and misclassified cases.

Table 6

Experimental results (parameters α, β and its misclassified cases)

| Parameter α | Parameter β | Epochs | Number of misclassified cases | Cases |
|---|---|---|---|---|
| $\alpha = 0.8$ | $\beta = 0.8$ | 41 | 9 | 14,26,36,37,41,58,59,60,62 |
| | $\beta = 0.9$ | 889 | 6 | 37,50,58,59,60,62 |
| $\alpha = 0.9$ | $\beta = 0.8$ | 46 | 9 | 14,26,35,36,37,41,58,59,62 |
| | $\beta = 0.9$ | 1489 | 7 | 37,50,58,59,60,62,63 |

In Figure 13 (a) and (b) error graphs for (3 = 0.8 and p = 0.9 are shown. All the results on the misclassified cases of the methods employed are summarised in Table 7.




a) b)

Figure 13. Error graph for p = 0.8 (a) and p = 0.9 (b)

Table 7

Summary table showing the used methods and misclassified cases

| Method | Parameters | Misclassified cases | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MDA | | | | 14 | 26 | 28 | 35 | 36 | 37 | 41 | | 58 | 59 | 60 | 62 | |
| Potential | | 4 | 8 | 14 | 26 | | | | 37 | | 50 | | 59 | | | |
| NN-1 | $\alpha = 0.8, \beta = 0.8$ | | | 14 | 26 | | | 36 | 37 | 41 | | 58 | 59 | 60 | 62 | |
| NN-2 | $\alpha = 0.8, \beta = 0.9$ | | | | | | | | 37 | | 50 | 58 | 59 | 60 | 62 | |
| NN-3 | $\alpha = 0.9, \beta = 0.8$ | | | 14 | 26 | | 35 | 36 | 37 | 41 | | 58 | 59 | | 62 | |
| NN-4 | $\alpha = 0.9, \beta = 0.9$ | | | | | | | | 37 | | 50 | 58 | 59 | 60 | 62 | 63 |

From Table 7 it can be seen that for the specific bankruptcy data sample, all the methods are unable to classify data vectors 37 and 59. Calculating in absolute numbers, we obtain that NN-2 correctly classified 90.5% cases, the potential function method and NN-4 - 89% cases, NN-1 and NN-3 - 85.7%, MDA - 82.5%. It can be concluded that for the given data sample the neural network method performs bankruptcy data classification more effectively, which actually corresponds to the conclusions made about the results achieved by Tam and Kiang.

Section 6 provides conclusions about neural network potentialities as applied to data analysis tasks.

**Chapter 4** studies possibilities of different clustering methods in data analysis.

Section 1 presents task statement, it is formulated as follows. To examine possibilities of different clustering methods and discover their ability to extract IF-THEN rules.

Section 2 continues a survey of popular clustering algorithms that are employed in data analysis. Among hierarchical clustering algorithms, the taxonomy algorithm FOREL has received special recognition. That is why the essence of the algorithm and its operation principle are analysed using an example from the IRIS database.

Recently, wide attention has been paid to a new clustering algorithm - grid clustering. The conventional cluster algorithms calculate a distance based on a similarity metric (Euclidean etc.) between cluster centres The patterns are clustered according to the resulting similarity index. The grid clustering algorithm differs from the conventional cluster algorithms in that it organises not the patterns but the value space, which surrounds the patterns. To organise the value space, a variation of the multidimensional data structure of the grid file is used, which is called "grid structure". The patterns are treated as points in the d-dimensional value space and are inserted at random into the grid structure. The points are stored according to their pattern values. The grid structure partitions the value space and administrates the points by a set of surrounding rectangular shaped blocks.

The section also presents the grid clustering algorithm and a case study as well as considers another clustering method - *subtractive clustering* - that is based on density measure of the data points in the feature space. The idea is to find regions in the feature space with high densities of data points. The point with the highest number of neighbours is selected as a centre for a cluster. The data points within a prespecified, fuzzy radius are then removed (subtracted), and the algorithm looks for a new point with the highest number of neighbours. This continues until all data points are examined. In what follows, density measure of data points is defined and execution principle of the algorithm is outlined. To demonstrate its execution, experimental data set was used.

Section 3 focuses on fuzzy clustering. It is reasonable to assume that points between the two cluster centres have a gradual membership in both clusters. The fuzzy c-means algorithm allows each data point to belong to a cluster to a degree specified by a membership grade, and thus each point may belong to several clusters.

The *fuzzy c-means* algorithm (FCM) partitions a collection of K data points specified by tri-dimensional vectors $u_k$ (k=l,2,..., K) into c fuzzy clusters, and finds a cluster centre in each, minimising the objective function. Fuzzy c-means is different from hard c-means, mainly because it employs *fuzzy partitioning*,where a point can belong to several clusters with the membership values of the data points to the clusters. To accommodate the fuzzy partitioning, the membership matrix M is allowed to have elements in the range [0, 1].Total membership of data points in all clusters has to possess the following feature. The sum of elements of any column has to be aqual to 1, the sum of all elements must be equal to K and the objective function is set as follows:

$$J(M, c_1, c_2, ..., c_c) = \sum_{i=j}^{c} J_i = \sum_{i=1}^{e} \sum_{k=1}^{K} m_{ik}^q d_{ik}^2, \quad (2)$$

where $m_{ik}$ is the membership between 0 and 1;

$c_i$ is the centre of fuzzy cluster i;

$d_{ik} = \|u_k - c_i\|$ is the Euclidean distance of the k-th point from the i-th cluster centre;

$q \in (1, \infty)$ is a weighting exponent.

There are two necessary conditions for J to reach a minimum: $$c_i = \frac{\sum_{k=1}^{K} m_{ik}^q u_k}{\sum_{k=1}^{K} m_{ik}^q} \quad (3)$$

and $$m_{ik} = \frac{1}{\sum_{j=1}^{c} \left( \frac{d_{ik}}{d_{jk}} \right)^{2/(q-1)}} \quad (4)$$

Jang J S. has proposed the FCM algorithm which determines the cluster centres $c_i$ and the membership matrix M using the following steps:

(1) Initialise the membership matrix M with random values between 0 and 1.
(2) Calculate cluster centres $c_i$ (i=l,2,.., c) using (3).
(3) Compute the objective function according to (2). Stop if either it is below a certain threshold level or its improvement over the previous iteration is below a certain tolerance.
(4) Compute a new M using (4).
(5) Go to step 2.

The cluster centres can alternatively be initialised first, before carrying out the iterative procedure. The algorithm may not converge to an optimum solution and the performance depends on the initial cluster centres, just as in the case of the hard c-means algorithm.

In what follows the FCM algorithm is illustrated with experimental data set.

Section 4 considers rule extraction using fuzzy clustering. Here an attempt is made to extend the application area of the FCM algorithm and to obtain the regularities that characterise the data in the form of rules.

Membership functions are commonly represented with the help of linguistic variables, which helps interpret fuzzy systems easier. Such notions as *high, law* or *medium* can well characterise the state of unidimensional objects. Instead, in case of multidimensional objects it turns to be not so easy. To describe a fuzzy classification problem, the following assumptions are made. Assume, there exist p variables $x_1, x_2, ..., x_p$, which are defined in the interval $X_i = [a_i, b_i]$, $a_i < b_i$. The final class set C is given for which the following distribution is valid:

$$\text{Class: } X_1 \text{ x } X_2 \text{ x ... x } X_p \rightarrow C.$$

The objective is to find a classifier that could solve classification problem. The fuzzy classifier is based on the set of final rules R for which the following holds:

$$R: \text{If } x_1 \text{ is } \mu_R^{(1)} \text{ and ... and } x_p \text{ is } \mu_R^{(p)} \text{ Then class is } C_R$$

where $C_R \in C$. The $\mu_R^{(i)}$ are assumed to be fuzzy sets in $X_i$, i.e. $\mu_R^{(i)}: X_i \rightarrow [0,1]$. Fuzzy sets $\mu_R^{(i)}$ are directly included in the rule. In real-world situations they can be replaced by the corresponding linguistic variables.

In solving many practical applications, the information necessary for the development and implementation of a fuzzy system can be divided into two kinds: numerical (the result of measurements) and linguistic (obtained from experts). Most of fuzzy systems are implemented using the second kind of knowledge, which is commonly represented in the form of a fuzzy rule base. It should be noted that in the cases when a fuzzy system with the numerical data has to be developed, certain important problems appear.

Let us assume for clearness that a fuzzy system with two inputs (input signals) and one output is being constructed. Thus the following form of learning data is required:

$$(x_1(i), x_2(i), d(i)), i = l,2,...$$

where $x_1(i)$ and $x_2(i)$ denote the incoming data but d(i) is the expected output. The task of the system is to form fuzzy rules so that possibly best result would be obtained on the output. The task stated can be accomplished in five stages.

**Stage 1.** Separation of input and output data. Actually, the minimal and maximal values of the input data are known, so intervals are determined, in which the allowable values are located: $[x_1^-, x_1^+],[x_2^-, x_2^+]$ *and* $[d^-, d^+]$ . Each of the intervals is divided into (2N+1) parts. For particular interval parts linguistic variables can be set, for example, $S_N$ (small N),..., $S_1$ (small 1), M (middle), $L_1$ (large 1),..., $L_N$ (large N) and their membership functions can be determined. Figure 14 shows an example of similar distribution, where the domain of signal $x_1$ is divided into 5 subintervals (N=2), the domain of signal $x_2$ is divided into 7 subintervals (N=3) but the domain of the output signal is partitioned into 5 subintervals (N=2).

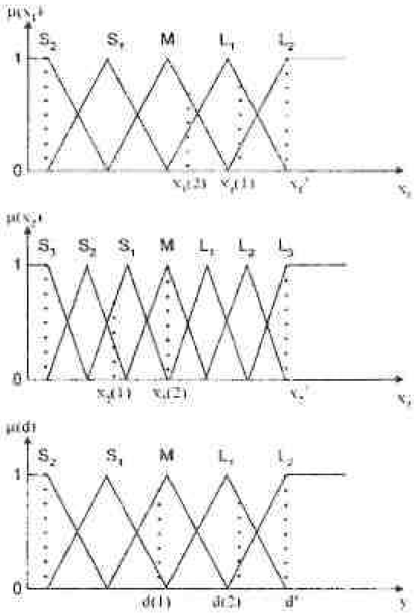



Figure 14. Partition of input and output data into intervals and corresponding membership functions

**Stage 2.** Construction of fuzzy rules using learning data set. At this stage, membership degrees of learning data ($x_1$(i), $x_2$(i)) and d(i) have to be determined for each of the selected domains. This is expressed using the values of membership functions. For example, in Figure 14 the membership degree of $x_1$(l) in domain $L_1$ is 0.8, in domain $L_2$ - 0.2; membership in other domains is 0. Similarly, the membership degree of $x_2$(2) in domain M is 1, whereas its membership in other domains is 0. in the same way we will ascribe $x_1$(i), $X_2$(i) and d(i) to those domains where they have maximal membership degrees. Say, xi(l) has maximal membership degree in domain $L_1$ , whereas $x_2$(2) in domain M. Thus for each pair of learning data a single rule can be set, for example, in this way:

($x_1$(l),$x_2$(l);d(l))→
{$x_1$(l)[max:0.8 in domain $L_1$], $x_2$(l)[max:0.6 in domain $S_1$]; d(l)[max:0.9 in domain M]}→
$R^1$: If ($x_1$ is $L_1$ and $x_2$ is $S_1$) Then y is M.

**Stage 3.** Determination of confidence degree for each rule. Taking into account that a lot of learning data pairs exist and for each of them a single rule can be generated, a possibility exists that the rules might be inconsistent. This relates to the rules having the same condition, but different conclusions. One of possible solutions of this problem might be assigning the confidence degree to each rule with a view to further choose the rule with the highest confidence degree. As a result, not only the problem of rule contradiction would be solved but also the total number of rules would decrease essentially.

For the rule in the form R: If ($x_1$ is $A_1$ and $x_2$ is $A_2$) Then (y is B) the confidence degree will be defined as follows:

$$SP(R) = \mu_{A_1}(x_1) \cdot \mu_{A_2}(x_2) \cdot \mu_B(y).$$

**Stage 4.** Formation of fuzzy rule base. Th $R^1$: **IF** ($x_1$ is $L_1$ **AND** $x_2$ is $S_1$) **THEN** y is M,

e principle of fuzzy rule formation is shown in Figure 15.

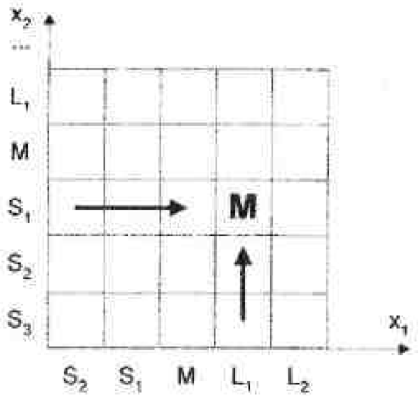



Figure 15. Formation of fuzzy rule base

The rule base is set in the form of a table, which is completed with rules as follows, if a rule is given in the form

$R^1$: **IF** ($x_1$ is $L_1$ **AND** $x_2$ is $S_1$) **THEN** y is M,

the value of a fuzzy set that is contained in the Then part of the rule, i.e. the value M in this example, is recorded at the point of intersection of column $L_1$ and row $S_1$. In case if various rules with the same condition exist, a rule with the highest confidence degree is selected of them.

**Stage 5.** Defuzzification. At this stage, mapping f: $(x_1, x_2) \to \overline{y}$, where $\overline{y}$ is the output value of the fuzzy system, has to be derived using the obtained knowledge base. The defuzzification is considered completed if a specific value for each linguistic variable is obtained. To accomplish that, the activity degree of the k-th rule is calculated using this formula:

$$\tau^{(k)} = \mu_{A_1^{(k)}}(x_1) \cdot \mu_{A_2^{(k)}}(x_2)$$

Actually, it is determined which of the obtained rules is more active for the specified input data vector.

Rule $R^1$ from the above-mentioned example has the activity degree $\tau^{(1)} = \mu_{L_1}(x_1) \cdot \mu_{M_1}(x_2)$ .

Now, a method of gravity centre determination, say, defuzzification by the gravity centre method *(COGS - Centre of Gravity for Singleton)* can be employed to calculate the output value $\overline{y}$ :

$$\overline{y} = \frac{\sum_{k=1}^{N} \tau^{(k)} \overline{y}^{(k)}}{\sum_{k=1}^{N} \tau^{(k)}}$$

After all the five stages have been completed successfully, a fuzzy rule base can be considered generated.

A well-known IRIS database was selected to perform experiments. The objective of the experiments was:

1) To acquire rules from the IRIS database using the FCM algorithm;
2) To ascertain the effect of membership function number on the count of acquired rules.
3) To check the quality of the rules obtained.

In the first part of the experiments, three membership functions were calculated for three clusters. Four rules were acquired for Class 1, three rules - for Class 2 and 11 rules for Class 3. The rules are shown in Table 8.

Table 8

Rules derived from the IRIS database

| |
|---|
| Rule 1: if X1 is MF1 to degree 0.75 and X2 is MF2 to degree 0.87 and X3 is MF1 to degree 0.92 and X4 is MF1 to degree 1 then Class 1 to degree 0.57<br>Rule 2: if X1 is MF1 to degree 0.94 and X2 is MF1 to degree 0.88 and X3 is MF1 to degree 0.95 and X4 is MF1 to degree 0.92 then Class 1 to degree 0.68 |
| Rule 3: if X1 is MF2 to degree 0.67 and X2 is MF1 to degree 0.58 and X3 is MF2 to degree 0.68 and X4 is MF2 to degree 0.67 then Class 2 to degree 0.17<br>Rule 4: if X1 is MF1 to degree 0.67 and X2 is MF1 to degree 0.75 and X3 is MF2 to degree 0.58 and X4 is MF1 to degree 0.54 then Class 2 to degree 0.15<br>Rule 5: if X1 is MF2 to degree 0.56 and X2 is MF2 to degree 0.54 and X3 is MF2 to degree 0.63 and X4 is MF2 to degree 0.63 then Class 2 to degree 0.11<br>Rule 6: if X1 is MF1 to degree 0.80 and X2 is MF1 to degree 1 and X3 is MF1 to degree 0.58 and X4 is MF1 to degree 0.63 then Class 2 to degree 0.28<br>Rule 7: if X1 is MF2 to degree 0.56 and X2 is MF1 to degree 0.88 and X3 is MF2 to degree 0.58 and X4 is MF1 to degree 0.5 then Class 2 to degree 0.13<br>Rule 8: if X1 is MF1 to degree 0.75 and X2 is MF1 to degree 0.71 and X3 is MF1 to degree 0.51 and X4 is MF2 to degree 0.54 then Class 2 to degree 0.14<br>Rule 9: if X1 is MF1 to degree 0.53 and X2 is MF1 to degree 0.71 and X3 is MF2 to degree 0.69 and X4 is MF2 to degree 0.63 then Class 2 to degree 0.15<br>Rule 10: if X1 is MF1 to degree 0.53 and X2 is MF2 to degree 0.58 and X3 is MF2 to degree 0.59 and X4 is MF2 to degree 0.63 then Class 2 to degree 0.11 |
| Rule 11: if X1 is MF2 to degree 0.94 and X2 is MF2 to degree 0.75 and X3 is MF2 to degree 0.97 and X4 is MF2 to degree 0.88 then Class 3 to degree 0.57<br>Rule 12: if X1 is MF1 to degree 0.83 and X2 is MF1 to degree 0.79 and X3 is MF2 to degree 0.59 and X4 is MF2 to degree 0.67 then Class 3 to degree 0.25<br>Rule 13: if X1 is MF2 to degree 0.94 and X2 is MF1 to degree 0.75 and X3 is MF2 to degree 1 and X4 is MF2 to degree 0.92 then Class 3 to degree 0.62 |

In the second part of experiments, different initial values of membership functions were selected and rule extraction was performed (see Figure 16 and Table 9).

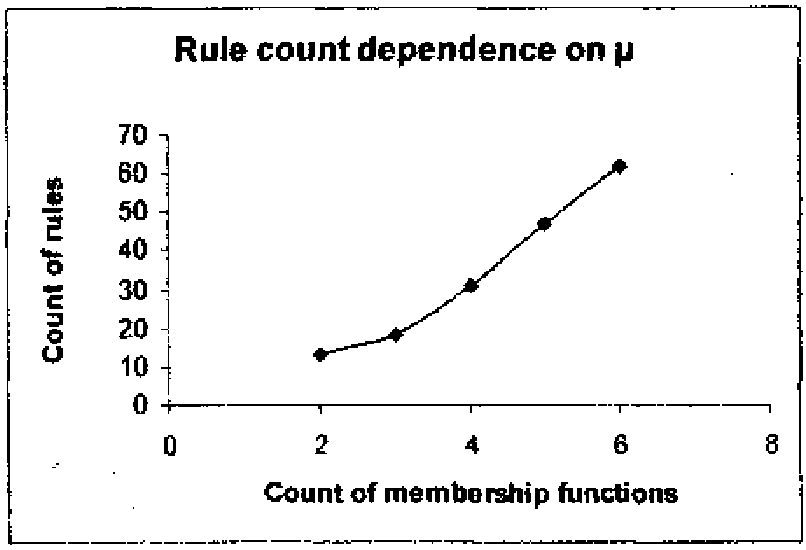



Figure 16. Graph of rule count dependence on the count of membership functions

Table 9

Dependence of the obtained rule number on the count
of initially set membership functions

| Number of memberships | Class 1 | Class 2 | Class 3 | Count |
|---|---|---|---|---|
| 2 | 2 | 8 | 3 | 13 |
| 3 | 4 | 3 | 11 | 18 |
| 4 | 7 | 11 | 13 | 31 |
| 5 | 13 | 16 | 18 | 47 |
| 6 | 21 | 19 | 22 | 62 |

Section 5 provides conclusions regarding clustering algorithm applications. These algorithms are mainly intended for multidimensional statistical data processing when the data are specified in the form of a table *object-feature.* They enable one to group objects in certain groups in which the objects are connected with each other according to a certain condition. No matter how these groups are called - taxons, clusters or classes; the main thing is that they sufficiently accurately represent the features of the objects.

**Chapter 5** considers neural network methods with relation to the process of conditional rule extraction.

Section 1 contains problem statement and analysis of the related research. Using the neural network technique it is possible to broaden the class of solvable tasks thus extending the potentialities of neural networks. Popular have become the rules that help *read* symbolic or linguistic information from artificial neural networks. A direct way of converting neural to symbolic knowledge is through rule extraction. This process provides a limited form of an explanation facility of how a neural network may classify any given input pattern. Rule extraction is a process that discovers the hyperplane positions of the input-to-hidden units and the hidden-to-output units of a neural network. These positions are then formulated as IF-THEN rules with the most important input unit labels acting as the rule antecedents. The discovery of the hyperplane positions can be found by a number of techniques that analyse the weights and biases of the neural network.

Rule extraction can be carried out using a variety of neural network types such as multilayer perceptions, Kohonen networks, radial basis function networks and recurrent networks. Rule extraction process in a common case is shown in Figure 17.

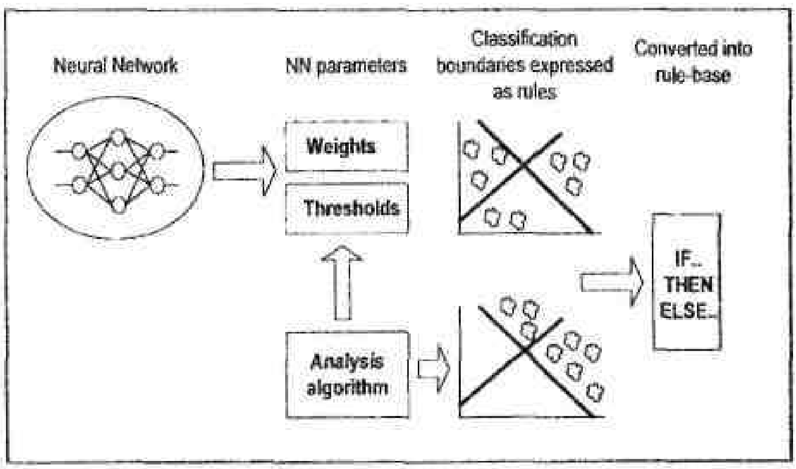



Figure 17. Rule extraction process

In what follows, a number of approaches to rule extraction from neural networks are outlined: VIA, N-of-M, and KBAN. The advantages of rule extraction from neural networks can be summarized as follows:

- The knowledge learned by a neural network is generally difficult to understand for humans. The provision of a mechanism that can interpret the network input/output mappings in the form of rules would be very useful.
- Deficiencies in the original training set may be identified. Thus the generalization of the network may be improved by the addition/enhancement of new classes. The identification of superfluous network parameters for removal would also enhance network performance.
- Analysis of previously unknown relationships in the data. This feature has a huge potential for knowledge discovery or data mining and possibilities may exist for scientific induction.

Section 2 examines the applicability of neural networks to rule extraction. Typical networks have hundreds or thousands of real-valued parameters. These parameters encode the relationships between the input features, x and the target value, y. Although parameter encodings of this type are not hard to understand, a great number of parameters in the typical network can make the task of understanding them very difficult. Moreover, in multilayer networks these parameters may represent varied relationships between the input features and the target values. In this case it is usually not possible to determine the effect of a given feature on the target value, because this effect may be mediated by the values of other features.

These relationships are represented by the hidden units in a network, which combine the inputs of multiple features, thus allowing the model to take advantage of dependencies among the features. A hypothesis was stated that using elements of the hidden layer can help characterise the interrelation among the input data.

Section 3 deals with the methodology of rule extraction from RBF networks. The nature of RBF networks makes them a suitable tool for rule extraction process. Section 3 of Chapter 3 described the general RBF network learning scheme. It was also shown that RBF network learning is normally undertaken as a two-stage process. The first stage consists of an unsupervised process in which the RBF centres (hidden units) are positioned. The second stage of learning involves the calculation of the hidden unit to output unit weights and output response. The radial basis functions are implemented by kernel functions in the hidden layer, which operate over a localized area of input space. The effective range of the kernels is determined by the values allocated to the centre and the width of the radial basis function. As was already mentioned, this is Gaussian function that is most frequently employed as a basis function. It is represented as

$$Z_1(x) = \exp(-\frac{\|x-\mu\|^2}{\delta_j^2} \, . \qquad (5)$$

The response of the output unit is calculated using equation:

$$y = \sum_{j=1}^{J} W_{ij} Z_j(x), \qquad (6)$$

where: W - weight matrix;

Z - hidden units activations;

x - input vectors;

μ - centre of basis function;

σ - width of receptive field.

Later the RBF network will be used in the context of rule extraction. The following hidden element's feature will be used for rule extraction: after training, each hidden element actually represents a single class of elements. The local nature of each RBF hidden unit enables a simple translation into a single rule:

IF $\text{Feature}_1$ is TRUE AND

IF $\text{Feature}_2$ is TRUE AND

IF $Feature_n$ is TRUE THEN $Class_x$,

where a *Feature* is composed of upper and lower bounds calculated by the RBF centre $\mu_n$ positions, RBF width σ and feature steepness S. The value of the steepness was discovered empirically to be about 0.6 and is related to the value of the width parameter. The values of μ and σ are determined by the RBF training algorithm. The upper and lower bounds are calculated as follows:

$$X_{lower} = \mu_i - \sigma_i + S \quad \text{un} \quad X_{upper} = \mu_i + \sigma_i - S. \qquad (7)$$

The RBF rule extraction algorithm RULEX can be seen in Figure 18:

| | |
|---|---|
| Input: | Hidden weights μ (centre positions)<br>Gaussian radius spread σ<br>Steepness S |
| Output: | One rule per hidden unit |
| Procedure: | Train RBF network on data set<br>For each hidden unit:<br>    For each $\mu_i$<br>        $X_{lower} = \mu_i - \sigma_i + S$<br>        $X_{upper} = \mu_i + \sigma_i - S$<br>Build rule by:<br>    antecedent=[ $X_{lower}$, $X_{upper}$]<br>    Join antecedents with AND<br>    Add class label<br>Write rule |

Figure 18. Rule extraction algorithm

Below an example is given to help understand the principle of rule extraction. For demonstration purposes, a two dimensional data set was employed, which is shown in Table 10.

Table 10

Experimental data

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $X_1$ | 1 | 3 | 6 | 10 | 2 | 2 | 5 | 6 | 4 | 8 | 8 | 4 | 9 | 1 |
| $X_2$ | 3 | 4 | 1 | 6 | 3 | 8 | 5 | 5 | 3 | 6 | 3 | 9 | 1 | 6 |

By using the RBF training algorithm, we derive two clusters and their centres in three iterations (see Figure 19).

As a result, the following weight vectors were obtained: $\mu_1$=(-0.73; 0.26) and $\mu_2$=(0.97;-0.35).

There were also derived radius values $\sigma_1^2 = 1.07$ and $\sigma_2^2 = 1.04$ corresponding to the clusters.

At the second stage of learning, radial functions and network output were calculated by formulae (5) and (6). In this case RBF network is considered trained.

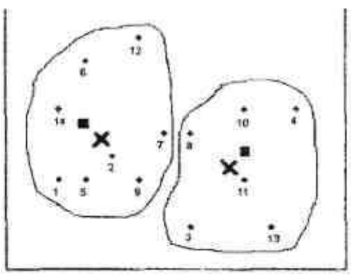


Figure 19. Two clusters with centre at points (-0.73; 0.26) and (0.97; -0.35)

Further RULEX algorithm execution yields values $X_{lower}$ and $X_{upper}$ calculated by formula (3) for each cluster.

Class 1. $X_{1\_lower}$= -0 73 - 1 03 + 0.6= - 1.16; $X_{2\_lower}$ = 0.26 - 1.03 +0.6 = -0.17;

$X_{1\_upper}$ = -0.73+ 1.03 -0.6 = -0.3; $X_{2\_ upper}$= 0.26 + 1.03 - 0.6 = 0.69.

Class 2. $X_{1\_lower}$ = 0.97 -1.01+ 0.6= 0.56; $X_{2\_lower}$ = -0.35 - 1.01 +0.6= -0.76;
$X_{1\_upper}$ = 0.97+ 1.01-0.6= 1.38; $X_{2\_upper}$=-0.35+1.01 -0.6 = 0.06.

Thus, for each hidden unit that represents the class we have derived the following rules:

IF ($x_1 \geq$ -1.16 AND $\leq$ -0.3) AND IF ($x_2 \geq$ - 0.17 AND $\leq$ 0.69) THEN CLASS 1.

IF ($x_1 \geq$ 0.56 AND $\leq$ 1.38) AND IF ($x_2 \geq$ - 0.76 AND $\leq$ 0.06) THEN CLASS 2.

By analyzing the input data, one can conclude that:

- at steepness value=0.6, the extracted rules erroneously describe 12 input vectors out of 14 (86% errors),
- at steepness value=0.2, the extracted rules erroneously describe 9 input vectors out of 14 (64% errors), and
- at steepness value=0, the extracted rules erroneously describe 4 input vectors out of 14 (28% errors). (Points 3, 6, 12 and 13 in Figure 19).

The rules extracted in this case look as follows:

IF ($x_1 \geq$- 1.76 AND $\leq$ 0.3) AND IF($x_2$ $\geq$-0.77 AND $\leq$ 1.29) THEN CLASS 1.

IF ($x_1 \geq$ -0.04 AND $\leq$ 1.98) AND IF ($x_2$ $\geq$ -1.36 AND $\leq$ 0.66) THEN CLASS 2.

Regions of rules are represented in Figure 20.

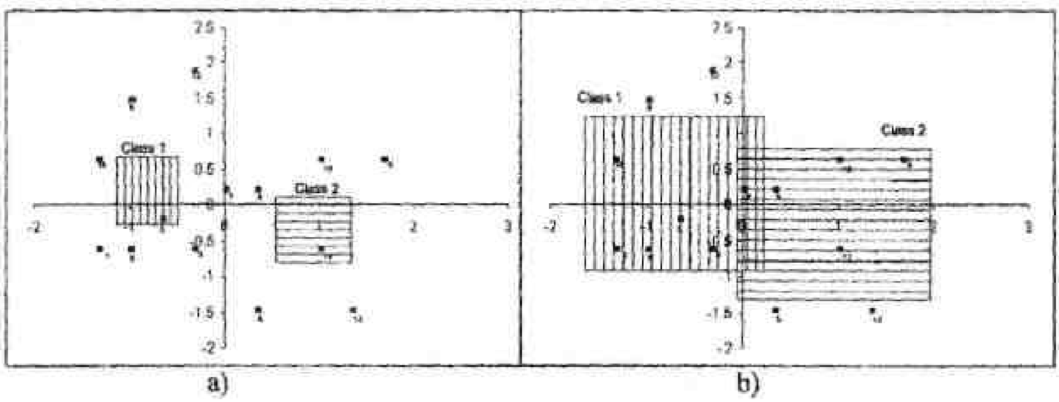



Figure 20. Region of rules: steepness=0.6 (a) and steepness=0 (b)

A conclusion can be made that the derivation of correct rules from the trained neural networks depends on the number of clusters originally set. Actually, the greater number of clusters is set, the more rules will be extracted.

Then two experiments were made aimed at solving practical task with bankruptcy data set and IRIS database. In the first experiment, bankruptcy data described in Section 6 of Chapter 3 were used (46 bankrupt companies and 17 non-bankrupt companies), whose data fragment is shown in Table 11. The objective of the experiment was to extract rules from the bankruptcy data

Table 11

Bankruptcy data fragment

| Bankrupt | | | | Non-bankrupt | | | |
|---|---|---|---|---|---|---|---|
| R3 | R7 | R9 | R31 | R3 | R7 | R9 | R31 |
| 0.29 | 1.75 | 0.67 | 0 | 0.01 | 1.01 | 0.93 | 0.04 |
| 0.28 | 1.75 | 0.65 | 0.08 | 0.04 | 1.04 | 0.97 | 0.05 |
| 0.37 | 2.77 | 0.58 | 0.08 | 0.05 | 1.05 | 0.95 | 0.04 |
| 0.05 | 1.13 | 0.39 | 0.02 | -0.45 | 0.51 | 0.47 | -0.13 |
| ... | ... | ... | ... | ... | ... | ... | ... |

Table 12 shows the results obtained in the course of the experiment whereas Table 13 lists the rules obtained under separate S values (Class 1 contains data on non-bankrupts but Class 2 - data on bankrupt companies).

Table 12

Results of bankruptcy data set training

| Correct | Values of parameter S | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | -0.9 | ... | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 |
| Class 1 | 15 | | 12 | 10 | 9 | 6 | 5 | 1 | 0 |
| Class 2 | 44 | | 44 | 44 | 44 | 44 | 44 | 44 | 44 |
| % | 93.7 | ... | 88.9 | 85.7 | 84.1 | 79.4 | 77.8 | 71.4 | 69.8 |

Table 13

Bankruptcy data set: characteristics of the extracted rules

| | Parameter S= -0.9 | Parameter S= 0.4 |
|---|---|---|
| Values of centres and radii | Class 1= 0.03 1.25 0.74 0.01<br>Class 2= 0.13 1.86 0.59 0.10<br>Values of radii = 0.68 3.92 | Class 1= 0.03 1.25 0.74 0.01<br>Class 2= 0.13 1.86 0.59 0.10<br>Values of radii = 0.68 3.92 |
| Rules correctly describe elements of classes (%) | 93.7 | 77.8. |
| Rule of Class 1 | IF (X1>= -1.54 AND < 1.61 ) AND<br>IF (X2>= -0.33 AND < 2.83) AND<br>IF (X3>= -0.84 AND < 2.32) AND<br>IF (X4>= -1.57 AND < 1.59)<br>THEN NON-BANKRUPT | IF (X1>= -0.24 AND < 0.31 ) AND<br>IF (X2>= 0.97 AND < 1.53) AND<br>IF (X3>= 0.46 AND < 1.02) AND<br>IF (X4>= -0.27 AND < 0.29)<br>THEN NON-BANKRUPT |
| Rule of Class 2 | IF (X1>= -4.69 AND < 4.95 ) AND<br>IF (X2>= -2.97 AND < 6.68) AND<br>IF (X3>= -4.23 AND < 5.41) AND<br>IF (X4>= -4.72 AND < 4.93)<br>THEN BANKRUPT | IF (X1>= -3.39 AND < 3.65 ) AND<br>IF (X2>= -1.67 AND < 5.38) AND<br>IF (X3>= -2.93 AND < 4.11) AND<br>IF (X4>= -3.42 AND < 3.63)<br>THEN BANKRUPT |

From Table 12 it can be seen that the rules obtained correctly describe bankruptcy data (44 out of 46) within the whole domain of parameter S, i.e., it can be stated that bankruptcy data are located in a fairly compact class.

In the second experiment, the !R1S database was employed that contained three flower classes of 50 elements each: *setosa, versicolor* and *virginica.* Every flower has 4 attributes: SL-sepal length, SW-sepal width, PL-petal length and PW-petal width. Table 14 shows data fragments of each class.

Table 14

IRIS data fragment

| Setosa | | | |
|---|---|---|---|
| SL | SW | PL | PW |
| 5.1 | 3.5 | 1.4 | 0.2 |
| 4.9 | 3.0 | 1.4 | 0.2 |
| 4.7 | 3.2 | 1.3 | 0.2 |
| 4.6 | 3.1 | 1.5 | 0.2 |
| ... | ... | ... | ... |

| Versicolor | | | |
|---|---|---|---|
| SL | SW | PL | PW |
| 7.0 | 3.2 | 4.7 | 1.4 |
| 6.4 | 3.2 | 4.5 | 1.5 |
| 6.9 | 3.1 | 4.9 | 1.5 |
| 5.5 | 2.3 | 4.0 | 1.3 |
| ... | ... | ... | ... |

| Virginica | | | |
|---|---|---|---|
| SL | SW | PL | PW |
| 6.3 | 3.3 | 6.0 | 2.5 |
| 5.8 | 2.7 | 5.1 | 1.9 |
| 7.1 | 3.0 | 5.9 | 2.1 |
| 6.3 | 2.9 | 5.6 | 1.8 |
| ... | ... | ... | ... |

The tasks of the experiment were as follows:

1. To accomplish network training by the RULEX algorithm at different training sets, namely:
   - Training set A - first 25 elements of every class,
   - Training set B - arbitrary 20 elements of every class;
   - Training set C - all 50 elements of every class.
2. To examine the effect of parameter S on the quality of extracted rules.

For example in case B, 20 elements arbitrarily selected from every class were employed as a training set. Table 15 shows the rules extracted at different S values, but Table 16 demonstrates the results of the experiment.

Table 15

Training set B: characteristics of the extracted rules

| | Parameter S=-0.9 | Parameter S=0 |
|---|---|---|
| Values of centres and radii | Class 1= 5.04 3.45 1.49 0.25<br>Class 2 = 5.99 2.77 4.32 1.35<br>Class 3 = 6.54 2.95 5.44 1.94<br>Values of radii = 0.19 0.43 0.73 | Class 1 = 5.04 3.45 1.49 0.25<br>Class 2 = 5.99 2.77 4.32 1.35<br>Class 3 = 6.54 2.95 5.44 1.94<br>Values of radii= 0.19 0.43 0.73 |
| Rules correctly describe elements of classes (%) | 98.67 | 42.67 |
| Rule of Class 1 | IF (X1>= 3.95 AND < 6.13 ) AND<br>IF (X2>= 2.36 AND < 4.54) AND<br>IF (X3>= 0.40 AND < 2.59) AND<br>IF (X4>= -0.84 AND < 1.34)<br>THEN SETOSA | IF (X1>= 4.85 AND < 5.23 ) AND<br>IF (X2>= 3.26 AND < 3.64) AND<br>IF (X3>= 1.30 AND < 1.69) AND<br>IF (X4>= 0.06 AND < 0.44)<br>THEN SETOSA |
| Rule of Class 2 | IF (X1>= 4.66 AND < 7.32 ) AND<br>IF (X2>= 1.44 AND < 4.10) AND<br>IF (X3>= 2.99 AND < 5.65) AND<br>IF (X4>= 0.01 AND < 2.68)<br>THEN VERSICOLOR | IF (X1>= 5.56 AND < 6.42 ) AND<br>IF (X2>= 2.34 AND < 3.20) AND<br>IF (X3>= 3.89 AND < 4.75) AND<br>IF (X4>= 0.91 AND < 1.78)<br>THEN VERSICOLOR |
| Rule of Class 3 | IF (X1>= 4.91 AND < 8.17 ) AND<br>IF (X2>= 1.33 AND < 4.58) AND<br>IF (X3>= 3.81 AND < 7.06) AND<br>IF (X4>= 0.31 AND < 3.57)<br>THEN VIRGINICA | IF (X1>= 5.81 AND < 7.27 ) AND<br>IF (X2>= 2.23 AND < 3.68) AND<br>IF (X3>= 4.71 AND < 6.16) AND<br>IF (X4>= 1.21 AND < 2.67)<br>THEN VIRGINICA |

Table 16

Training set B: characteristics of the extracted rules

| | Values of parameter S | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Correct | -0.9 | -0.8 | -0.7 | -0.6 | -0.5 | -0.4 | -0.3 | -0.2 | -0.1 | 0 | 0.1 | 0.2 | 0.3 | 0.6 |
| Class 1 | 49 | 49 | 48 | 48 | 45 | 40 | 39 | 27 | 14 | 9 | 2 | 0 | 0 | 0 |
| Class 2 | 50 | 49 | 49 | 48 | 45 | 44 | 40 | 36 | 28 | 20 | 10 | 3 | 0 | 0 |
| Class 3 | 49 | 49 | 48 | 47 | 45 | 43 | 43 | 42 | 39 | 35 | 29 | 23 | 16 | 0 |
| % | 98.7 | 98 | 96.7 | 95.3 | 90 | 84.7 | 81.3 | 70 | 54 | 42.7 | 27.3 | 17.3 | 10.7 | 0 |

The data obtained prove that parameter S plays an essential role in the application of the RULEX algorithm: the greater the negative value of S, the more the lower boundary of rule performance range, $X_{lower}$, decreases at the same time raising the upper boundary, $X_{upper}$, of the range. That causes the enlargement of the cluster describing antecedent part and thus increases the value of the area in which the extracted rule is fulfilled. This effect is illustrated with Figure 21.

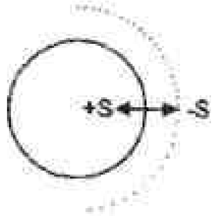



Figure 21. The effect of parameter S increase/decrease

For training sets A, B, and C, the dependence of the total count of elements, correctly describing rules, on parameter S is shown in Table 17 and represented as a graph in Figure 22.

Table 17

The dependence of the total count (%) of rule satisfying elements on S

| | Values of parameter S | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| % | -0.9 | -0.8 | -0.7 | -0.6 | -0.5 | -0.4 | -0.3 | -0.2 | -0.1 | 0 | 0.1 | 0.2 | 0.3 |
| A | 100 | 99.3 | 99.3 | 98.7 | 97.3 | 96 | 92 | 88 | 79.3 | 66 | 55.3 | 44.7 | 32 |
| B | 98.7 | 98 | 96.7 | 95.3 | 90 | 84.7 | 81.3 | 70 | 54 | 42.7 | 27.3 | 17.3 | 10.7 |
| C | 100 | 98.7 | 98.7 | 97.3 | 97.3 | 92.7 | 87.3 | 84 | 70.7 | 58.7 | 47.3 | 34.7 | 23.3 |

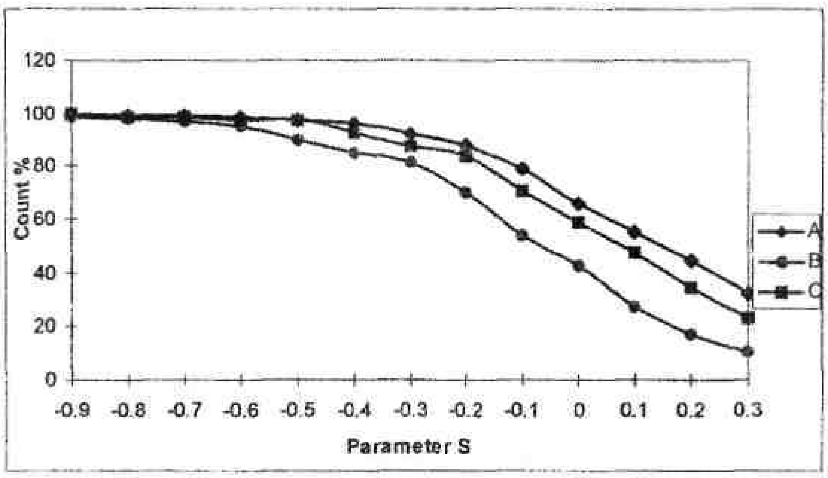



Figure 22. Dependence of the count (%) of elements correctly describing rules on parameter S

Section 4 presents conclusions on the suitableness of rule extraction from neural networks.

**Conclusion** examines the results of the doctoral thesis and outlines suggestions for future research.

**Appendix** outlines the authors developed software for making experiments aimed at association rule extraction from statistical data. To write the software, the *Matlab* programme package was used.

# MAJOR FINDINGS OF THE RESEARCH

The main goal of the doctoral thesis was to study and develop methods for creating conditional rules that are based on three rule extraction techniques.

**Major findings** of the research can be summarised as follows:

(1) Studying possibilities of artificial neural networks

- neural network functioning is examined using a radial basis function network;
- the RBF network learning algorithm is analysed with its focus on the first stage of learning, i.e. clustering
- neural network application possibilities in pattern recognition and optimisation tasks are considered;
- experiments are performed to show neural network application possibilities in different data analysis tasks;
- analysis of bankruptcy data using different methods is conducted and
    - as a novelty, application of potential function method is proposed, which provides good results in bankruptcy data analysing;
    - a neural network architecture with particular learning parameters is obtained experimentally; network operation results at these parameters confirm other researchers' opinion on the effectiveness of neural network application in bankruptcy diagnostics.

(2) Studying application of clustering methods

- application area of clustering algorithms is studied;
- most typical clustering methods and their potentialities are examined and evaluated;
- choice of clustering methods for modelling regularities is validated;
- a series of experiments aimed to demonstrate the execution of clustering algorithms are performed.

(3) Studying potentialities of fuzzy clustering

- a method of fuzzy rule base extraction from the numerical data is developed and discussed;
- as a novelty, application of the fuzzy clustering algorithm FCM to rule base extraction is proposed.

(4) Studying methods of conditional rule obtaining from neural networks:

- RBF neural network applicability to rule extraction from the trained neural network is studied;
- a procedure of rule extraction from the RBF neural network is developed;
- the effect of parameter S on the rule quality is examined;
- the method is applied to solving practical tasks.

(5) Studying application potentialities of association analysis

- basic algorithms for association rule extraction are discussed and examined;
- the influence of association rules confidence and support values on the process of rule extraction is studied;
- principles and methodology for rule extraction with association rule methods are worked out; software for association rule extraction from the data is written;
- the method is applied to solving practical tasks related to statistical data processing .

In the course of work it was concluded that using IF-THEN rules enables execution of classification, pattern recognition, association obtaining and other tasks. These methods can be applied to different data samples. Many clustering algorithms are mainly employed either separately or at the stage of neural network learning data preparation.

The motivation for studying rule extraction methods was the following:

- the amount of multidimensional data to be analysed becomes too large for the potentialities of statistical analysis;
- previously unknown regularities are present in the data;
- the regularities found can be represented in a way that is easy to perceive and understand for the user.

The method of association rule extraction is recommended for use in the cases when the data are integers and repeating elements can be observed in them.

The RBF network based method and fuzzy rule method are equivalent. Their performance can only be compared upon checking the indices of the obtained rule's quality, i.e. upon calculating the number of elements that correctly describe the rules.

One result of the research was confidence that different rule extraction methods are able to extract rules from the multidimensional data and these obtained rules are qualitative.

**Future research** will be focused on exploring clustering methods and rule extraction from artificial neural networks.

Of particular interest are fuzzy clustering models in which clusters have the form of an ellipse. This enables one to more accurately describe clusters and diminish information losses.

It is also planned to continue studying rule extraction techniques from neural networks. Main attention will be concentrated on RBF neural networks.

Software development intended for the application of rule extraction techniques to solving economic tasks could arouse a practical interest.

## REFERENCES


1. *Ackley D.H., Hinton G.E.. Sejnowski T.J. A learning algorithm for Boltzrnann machines// Cognitive Science. - 1985. -9. -p. 147-169.*
2. *Aggarwal C.C., Procopius C, Yu P.S. Finding localized associations in market basket data// IEEE Transactions on Knowledge and Data Engineering. - 2002. -vol. 14. - p. 51-62.*
3. *Agrawal R., Imielinski T., Swami A. Mining association rules between sets of items in large databases//Proc. Conf. on Management of data. - ACM Press, 1993. -p. 207-216.*
4. *Agrawal /?., Srikant R. Fast algorithms for mining association rules// Proceedings of the 20th International conference on Very Large Databases. - Chile., 1994. - p. 487-499.*
5. *Altman E. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy// Journal of Finance. - 1968. -vol. 13. -p. 589-609.*
6. *Andrews R., Diederich J., Tickle A. A survey and critique of techniques for extracting rules from trained artificial neural networks// Knowledge-Based Systems. -1995. -8(6). -p. 373-389.*
7. *Andrews R., Gewa S. Rule extraction from a constrained error back propagation MLP// Proc. 5th Australian Conference on Neural Networks. -Brisbane, 1994. -p. 9-12.*
8. *Andrews R., Gewa S. RULEX and CEBP networks as the basic for a ride refinement system// Hybrid Problems, Hybrid Solutions. - 1995. -p. 1-12.*
9. *Back B., Laitinen T., Sere K. Neural networks and bankruptcy prediction: funds flow, accrual ratios and accounting data// Advances in Accounting. -1996. -14. -p. 23-37.*
10. *Beaver R. Financial ratios as predictors of failure, Empirical Research in Accounting: Selected Studies//J.Accounting Research. - 1966. - vol. 4. -p. 71-111.*
11. *Bishop CM. Neural networks for pattern recognition. - Oxford.Oxford University Press, 1995.- 504 p.*
12. *Chiu S. Fuzzy Model Identification Based on Cluster Estimation// Journal of Intelligent and Fuzzy Systems. - 1994. -2. - p. 267-278.*
13. *Coughlin J.P., Baran R.H. Neural Computation in Hopfield Networks and Boltzrnann Machines. - Newark: University of Delaware Press, 1995. - 281 p.*
14. *Craven M., Shavlik J. Extracting tree-structured representations of trained networks// Advances in Neural Information Processing Systems. -1995. - vol. 8. -p. 24-30.*
15. *Duch W., Setiono R., Zurada J.M. Computational Intelligence Methods for Rule-based Data Understanding// Proceedings of IEEE. - 2004. - 92(5). -p. 771-805.*
16. *Everitt B.S. Cluster analysis. - London:Edward Arnold, 1993. -170 p.*
17. *Fausett L. Fundamentals of Neural Networks: Architectures, Algorithms and Applications. - New York: Prentice Hall International Inc., 1994. - 461 p.*
18. *Fayyad U., Piatetsky-Shapiro G., Smyth P. From data mining to knowledge discovery in databases//'AI Magazine. - 1996. - 17 (3), -p. 37-54.*
19. *Fuzzy Cluster Analysis/ Hoppner F., Klawonn F, Kruse R. etc. - England: John Wiley and Sons, 1999. -289 p.*
20. *Giudici P. Applied Data Mining: Statistical Methods for Business and Industry. - John Wiley & Sons Ltd, 2003. - 359p.*

21. *Grabusts P. Extracting rules from trained RBF neural networks// Proc. of the 5th International Conference. - Rezekne: RA izdevnieclba, 2005. - p333-39.*

22. *Grabusts P. Analysing Bankruptcy Data with Neural Networks/AProc. I0'h International Conference on Soft Computing. -Brno, Czech Republic, 2004. -fi. 111-117.*

23. *Grabusts P. Using Association Rules to Extract Regularities from Data// Proc. 6th International Baltic Conference on Data Bases and Information. Systems. -Riga, 2004. -p. 117-126.*

24 *Grabusts P. Asociāciju likumu analīzes izmantošana likumsakarību meklēšanā// RTU Zinātniskre Raksti: Datorzinātne. Datorvadības Tehnoloģijas.* -Rīga: *RTU izdevniecība, 2003. - 15.sējums. -110.-118. lpp.*

25. *Grabust P., Borisov A. Using grid-clustering methods in data classification// Proceedings of the International Conference on Parallel Computing & Electrical Engineering-PARELEC'2002. - Warsaw, Poland, 2002 -p. 425-426.*

26. *Grabusts P., Borisovs A. RBF neironu tlklu pielietosanas perspektlvas// II. Pasaules latviešu zinātnieku kongresa tēžu krājums. - Rīga, 2001. - 570. lpp.*

27 *Grabusts P. Klasterizācijas metodes izmantošana RBF neironu tīklos// III Starptautiskās zinatniskās konferences „ Vide. Tehnoloģija. Resursi" materiāli. -Rēzekne: RA izdevniecība, 2001. - 257.-262. lpp.*

28. *Grabusts P. A study of clustering algorithm application in RBF neural networks// Scientific Proceedings of Riga Technical University: Computer Science. Information Technology and Management Science. - Riga: RTU, 2001. -vol. 5. -p. 50-57.*

29. *Grabusts P. Solving TSP using classical simulated annealing method// Scientific Proceedings of Riga Technical University: Computer Science. Information Technology and Management Science. - Riga: RTU, 2000. - vol. 2. - p. 32-39.*

30. *Grabust P. Using a thermal equilibrium method in the neural networks// International Conference on Parallel Computing & Electrical Engineering-PARELEC'98. - Bialystok, Poland, 1998. -p. 261-263.*

31. *Grabusts P. A thermal equilibrium method for Boltzmann machines// Modern Aspects of Management Science. - Riga: RTU, 1998. - Nr. 2. -p. 13-19.*

32. *Hopfield J. Neural networks and physical systems with emergent collective computational abilities//Proc. Natl.Acad.Sci. - USA, 1982. -p. 2554-2558.*

33. *Hush D.R., Home B.G. Progress in Supervised Neural Networks. What's new since Lippmann?//IEEE Signal Processing Magazine. -1993. - vol. 10,-No I. -p. 32-43.*

34. *Izplūdusī loģika, iespējamību teorija un to pielietojumi: Metodiskais līdzeklis/ A.Borisovs, L.Dubrovskis, L.Aleksejeva u.c. - Riga: RTU, 1995. -136 lpp.*

35. *Jang J.S., Sun C.T.,Mizutani E. Neuro-Fuzzy and Soft Computing. - New Jersey: Prentice Hall, 1997. - 614p.*

36. *Kirkpatrick S., Gelatt C.D., Vecchi M.P. Optimization by simulated annealing//Science.*

*-1983. -220. -p. 671-680.*

37. *Laarhoven P.J., Aarts E.H. Simulated Annealing: Theory and Applications. - Holland: D.Reidel Publishing Company, 1987. - 204p.*

38. *McCarry K., Wermter S., Macintyre J. The extraction and comparison of knowledge from local function networks// International Journal of Computational Intelligence and Applications. - 2001. - 1(3). -p. 369-382.*

39. *Mākslīgie neironu tīkli: arhitektūra, atgoritmi un pielietojumi: Mācību līdzeklis/ A.Borisovs, L.Dubrovskis, G.Kulesova u.c. - Riga: RTV, 1998 - 110 lpp.*

40. *Newquist H.P. Data Mining: the Al metamorphosis// Database programming and design. - 1997. - Nr. 9.*

41. *Odom M., Sharda R. A neural network model for bankruptcy prediction// Proc. Int. Joint Con}. Neural Networks. - San Diego, CA, 1990. -p. 163-168.*

42. *Roiger R.J.,* *Geatz M.W. Data mining. A tutoral-based primer. - Addison Wesley, 2003.*

*-364 p.*

43. *Rudorfer G. Early bankruptcy detecting using neural networks// APL Quote Quad. -1995. -vol. 25, N. 4. -p. 171-178.*

44. *Rumelhart D.E., Hinton G.E., Williams R.J. Learning representations by backpropagatingerrors//Nature. - 1986. -323(9). -p. 533-536.*

45. *Russel S., Norvig P. Artificial Intelligence: A Modern Approach. - Englewood Cliffs, NJ: Prentice Hall, 1995. - 932 p.*

46. *Saito K., Nakano R. Medical diagnostic expert system based on PDP model// Proceedings of the IEEE International conference on Neural Networks - IEEE Press, 1988.-p. 255-262.*

47. *Schikuta E. Grid-Clustering: A fast hierarchical clustering method for very large data sets//Proc. 13th Int.Conf on Pattern Recognition. - 1996. -p. 101-105.*

48. *Software for data mining - Ghost Miner/ Internet. - http://www.faspl.com.pl/?a=product list&id=1*

49. *Software for statistical analysis - SPSS/ Internet. - http://www.spss.com/*

50. *Software for technical computing - Matlab/ Internet. - http://www.mathworks.com/*

51. *Srikant R., Agrawal R. Mining quantitative association rules in large relational tables// Proc. Int. Conf. Management of Data. - ACM Press, 1996. -p. 1-12.*

52. *Tarn K., Kiang M. Managerial applications of the neural networks: The case of bank failure predictions// Management Science. - 1992. - vol. 38. -p. 416-430.*

53. *Tan P.N., Steinbach M., Kumar V. Introduction to Data Mining. - Pearson Education, Inc., 2006. - 769p.*

54. *Thrun S. Extracting Rules from Artificial Neural Networks with Distributed representations// Advances in Neural Information Processing Systems. - 1995. - vol. 7. -p. 505-512.*

55. *Towell G., Shavlik J. Extracting of refined rules from Knowledge-based neural networks//Machine Learning. - 1993. - vol. 13, No 1. -p. 71-101.*

56. *Towell G., Shavlik J., Noordwier M. Refinement of approximate domain theory by Knowledge-based Neural Networks// Eight National conference on Artificial Intelligence. - 1990. -p. 861-866.*

57. *Tresp V., Hollatz J., Ahmad S. Network structuring using Rule -based knowledge// Advances in Neural Information Processing Systems. - 1993. - vol. 4. -p. 871-878.*

58. *XJCl Machine learning repository/ Internet. - http://www.ics.iici.edu/~mlearn/MLSummary.html*

59. *Wang L.X., Mendel J.M. Generating fuzzy rules by learning from examples// IEEE Transactions on Systems, Man, and Cybernetics. - 1992. - vol. 22, nr 6. -p. 1414-1427.*

60. *Wielinga B.J., Schreiber A. Th., Breuker J.A. KADS: a modelling approach to knowledge engineering//Knowledge Acquisition- - 1992. - 4(1). -p. 5-53.*

61. *"laden LA. Fuzzy Sets, Fuzzy Logic and Fuzzy Systems: Selected Papers by Lotfi A. Zadehf/ Advances in Fuzzy Systems - Applications and Theory. - 1996. - Vol 6. - 826 p*

62. *Zhou Z.H. Rule extracting: using neural networks or for neural networks?// Journal of Computer Science and Technology. -2004. - 19(2). -p. 249-253.*

63. *Zhou Z.H., Jiang y., Chen S.F. Extracting symbolic rules from trained neural network ensembles//AI Communications. -2003. - 16(1). -p. 3-15.*

64. *Дюк В., Самойленко А.* Data Mining. *-Санкт-Петербург: Питер, 2001. - 366 с.*

65. *Забродин В.Ю. Проблемы классификации (обзор)//НТИ, 1980. - Сер. 2. - с. 35 -40.*

66. *Загоруйко Н.Г., Елкина В.П., Лбов ГС. Алгоритмы обнаружения эмпирических закономерностей. - Новосибирск: Наука, 1985. -110 с.*

67. *Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. - Москва: Горячая линия-Телеком, 2004. - 383 с.*

68. *Хорошевский В.Ф., Гаврилова Т.А. Базы знаний интелектуальных систем. -Санкт-Петербург: Питер, 2001. - 382 с.*