Annales UMCS Informatica AI 4 (2006) 204-217

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

Discovery of association rules from medical data – classical and evolutionary approaches

Halina Kwaśnicka*, Kajetan Świtalski

Department of Computer Science, Wrocław University of Technology,
Wybrzeże S.Wyspiańskiego 27, 50-370 Wrocław, Poland

Abstract

The paper presents a method of association rules discovering from medical data using the evolutionary approach. The elaborated method (*EGAR*) uses a genetic algorithm as a tool of knowledge discovering from a set of data, in the form of association rules. The method is compared with known and common method – *FPTree*. The developed computer program is applied for testing the proposed method and comparing the results with those produced by *FPTree*. The program is the general and flexible tool for the rules generation task using different data sets and two embodied methods. The presented experiments are performed using the actual medical data from the Wroclaw Clinic.

1. Introduction

A growing number of institutions collect large sets of data, in the forms of databases or warehouses. These data usually contain hidden knowledge that can be very important and useful for the owners, but it is impossible to disclose this knowledge usings tandard tools and means. So, using more efficient solutions, producing better results, providing knowledge in the comprehensible form is very desirable. It leads to the new field of Artificial Intelligence known as Data Mining (*DM*) – a part of wider area – Knowledge Discovering from Databases (*KDD*) [1,2].

Recently, specialized algorithms of Data Mining as well as the whole process of Knowledge Discovery has been widely used in such domains as banking, marketing, telecommunication, power engineering, pharmacy, medicine, and others. In all domains it is possible to use the same Data Mining methods, but we must take into account a specificity of analysed data – in some cases the data have attributes with symbolic and numerical values, which must be considered in a selection of appropriate method. Some Data mining methods are not suitable for incomplete, noised data, etc., therefore it is important to consider both –

*Corresponding author: *e-mail address*: halina.kwasnicka@pwr.wroc.pl

a type of data and a goal of the data mining process. Domain experts should always be included in the Data Mining process to guarantee proper evaluation and verification of acquired knowledge. Sometimes experts are able to put forward an interesting hypothesis and Data Mining methods are able to verify them.

The main purpose of the paper is to study possibilities of association rules generation from medical databases by means of genetic algorithms and comparing the usefulness of developed method with the classical approach. The new genetic method, called titEGAR (***E****xtended* ***G****enetic* ***A****ssociation* ***R****ules*) is proposed. The word *Extended* means, that the authors base on the *GAR* method [3]. Developed computer program is designed to be flexible and user friendly, the results are presented in a readable form. Two medical databases applied here were the subject of our earlier study but for the classification task [4,5].

2. Generation of association rules as a Data Mining task

Knowledge Discovering from Data is a nontrivial process of searching some actual, new, potentially useful and comprehensible patterns present in databases [6]. This process is a sequence of tasks that should ensure gathering useful and understandable knowledge:

1. Understanding of the problem domain, identification of the final goal of the *KDD* process.
2. Data gathering.
3. Preprocessing of data, data consolidation and refining.
4. Selection of adequate data (data reduction) and enrichment.
5. Encoding the data.
6. Data mining (exploration the data):
   – selection of an adequate task of the data mining process,
   – selection a model of knowledge representation and an algorithm for data exploring,
   – running the algorithm – searching the data set and generation of found patterns,
7. Interpretation, presentation and explanation of ’mined’ knowledge.
8. Consolidation and practical exploitation of the disclosed knowledge.

The Data Mining task can be divided using different criteria, for example a form of generated knowledge or the goal of discovering process [8]. The most popular tasks are classification, clustering, dependency modelling and association rules generation. Each of these tasks can be performed using different algorithms [7-9]. The paper is focused on association rules generation.

Discovering association rules is the most general mechanism in the field of Data Mining and it can be used also for classification (but under some additional assumptions). A typical form of association rule appears like logic implication:

$$Body \Rightarrow Head \quad [support, confidence], \tag{1}$$

where:

- *Body* and *Head* are sets of attributes with corresponding values;
- *Support* is a frequency of coexistence (being together) of attributes values from the *Body* of a rule in database. It is measured in percentage. When the support of a rule is equal to 50%, it means that the half of all examples in the database have attributes with values specified in the *Body* of considered rule;
- *Confidence* is a frequency of coexistence values of attributes from the *Head* of rule in those records (examples), in which the values of attributes from the *Body* of the rule are present. The confidence of a rule equal to 50% means, that the half of examples (records) that have attribute values equal to those specified in the *Body* of the rule, have also values of attributes from the *Head* of rule.

3. Generation of association rules

One can distinguish two main groups of methods of association rules discovering: classical ones and so-called soft computing approach. Classical algorithms are known, described and applied in a number of computer systems, but soft algorithms are still under development. At current stage of research it is not possible to point to one, univocally the best method. A genetic algorithm seems to be the most promising soft computing approach. This approach is used in the paper.

3.1. The classical approach

Apriori is the simplest algorithms for rule discovering [7]. Strictly speaking, *Apriori* generates frequent patterns which can be used for generation of association rules. It is an iterative process, in which patterns containing k elements are used for defining $(k + 1)$ – element patterns. The basic property of frequent patterns is used in the method: each nonempty subset of frequent set is also a frequent set, where *frequent set* (called also *frequent pattern*) is understood as a pair of < *attribute*,*value* > coexists in the considered database with assumed frequency. If the given set *S* is not frequent, searching the frequent sets we can eliminate from consideration all such sets *S*', for which *S* is a subset (*S*' is called superset). It allows to confine a search space by generation candidates for $(k + 1)$ – elements frequent pattern among supersets of k – elements frequent sets. Apriori reveals week efficiency for large volume of data, therefore a number of modifications have been proposed [7].

FPTree (*Frequent Pattern Tree*) is an efficient algorithm of frequent patterns generation [2]. The core of *FPTree* is a structure called tree of frequent patterns, which allows to compress the data. The task of association rules generation

requires discovering the knowledge in the understandable form for users. The process of rules discovering from *Frequent Pattern Tree* is called *Tree Mining*, it is a recurrent process [2]. Literature review shows that extitFPTree is one of the most efficient classical algorithms of association rules discovering. It is complete – all patterns with assumed frequency are discovered, it looks over the database only once a run (it decides about high efficiency of *FPTree*), it does not require generation of candidates, the structure of *Frequent Pattern Tree* allows for compressed representation, frequency of found patterns decreases. One can find a number of different versions of *FPTree*, in these versions an efficiency of the method is optimised. Fig. 1 shows the pseudocode of our implemented method. In the presented paper a simple algorithm of association rules generation on the basis of obtained frequent patterns is used: For each frequent pattern W we generate all combinations of rules $A \Rightarrow B$, where A is any subset of the frequent pattern W, but B is a complement $W \setminus A$. The support of a such generated rule is equal to frequency of frequent patterns W and the confidence is quotient of support of the frequent set W and support of the set A.

```
function Mine(tree, support)
begin
        if ( has_one_branch(tree) ) generate_all_patterns_from(tree)
                    //generated patterns – all combinations of elements
                    //on the single branch
else if tree is empty=return {}
      for_each(element e of tree) do
          make conditional_pattern for element e
                    //finds paths contained e
          filter conditional_pattern, support
                    //leaves only elements with sufficient support
          make conditional_tree with conditional_patterns
                    // conditional patterns during construction of
                    // conditional tree are treated similar as records
                    // during  construction of main tree
          new_patterns = mine (conditional_tree, support)
          for_each (new_patterns) add element e
          result += new_patterns
      end_for_each
return result
end
```

Fig 1. A pseudocode of implemented *FPTree* algorithm

FPTree can work only using data with discrete attributes, but the authors' method accepts discrete (symbolic) and continuous values of attributes. We applied a discretization algorithm for *FPTree* which allows for comparison the results produced by both methods. We have used a domain-independent, simple,

unsupervised discretization algorithm, on the basis of values frequency. A number of intervals is assumed (as a program parameter), but their boundaries are selected to ensure a similar number of training examples in all intervals.

3.2. A genetic algorithm as a data mining tool (soft computing method)

Genetic algorithm refers to a class of algorithms based on probabilistic adaptation inspired by the principles of natural evolutions. It follows a search strategy of a population of individuals, each representing a possible solution of the considered problem. It does not guarantee finding a global solution, but is an acknowledged and widely used search technique, which gives usually satisfactory results [10-13].

As it is mentioned above, potential solutions are encoded into individuals, a set of them is generated as an initial population – it can be done randomly. Next, each individual in the population is evaluated using a defined fitness function. A fitness function takes into account a quality of solution, which is encoded in the considered individual. Taking into account individuals' fitness, the reproduction process is introduced: better individuals have higher probabilities for the reproduction process, and mechanisms drawn from biological evolution, such mutation and crossover are applied. They guarantee that the next generation of population (population of children) differs from the previous one, because better individuals are preferred and better solutions appear in successive generations.

The problem is in parameters setting – one should secure the balance between exploring and exploiting abilities of an applied genetic algorithm. Without it, the algorithm can stop in a local optimum. Genetic algorithms are widely used in such domains, as scheduling problems, financial modelling, function optimisation, etc. They are also useful in medical data analysis, including data mining tasks using data collected in different clinics [4,11,14].

A genetic algorithm applied in the paper is based on experience of the authors of *GAR* system (***G****enetic* ***A****ssociation* ***R****ules*), which gives interesting results for continuous data [3]. *GAR* generates frequent patterns, next the patterns are used for association rules generation. Numerous databases, especially medical ones, contain symbolic (discrete) and continuous data, therefore we have extended *GAR* making it able to manage all types of attributes. Therefore the name of the method proposed here method is *EGAR* (***E****xtended GAR*). Representation of individuals and genetic operators (mutation and crossover) undergo modifications. Such adapted algorithm is more applicable and useful.

In *EGAR* the Mitchigan approach is used [10,11], so one individual contains information about partial solution of the problem, it means that one individual encodes a single frequent pattern. To evolve a number of frequent patterns we must run the program repeatedly, taking the best individual (frequent pattern)

from the last generation of each iteration. A number of obtained frequent patterns acquired from data depends on a number of runs.

Each individual consists of two chromosomes, they describe attributes present in the frequent pattern as well as intervals of their values. One chromosome contains continuous attributes, a gene in this chromosome is a triple $< a,l,u >$, where a is an attribute, l is a lower value (boundary) and u is an upper value of the attribute a. The second chromosome of the same individual contains discrete attributes, a gene is a pair $< a,v >$, where a is an attribute and v is its value. A length of chromosomes differs between individuals, it depends on a number of attributes in a frequent pattern represented by an individual.

In *EGAR* we applied the stochastic sampling selection method with elitism – the best individual from the current population is taken to the next generation without changes [10,11]. Adding to the global set of discovered frequent patterns the best individual from the last generation, we are sure that it is the best pattern found in a given run of evolution (iteration). The defined fitness function can decide about success of developed genetic algorithm, fitness of individual must reflect a degree in which the encoded solution fulfils the destined task. In principle, our problem is multi-objective: the *accuracy* (a number of covered training examples by an evaluated pattern) seems to be a very natural criterion, but using only this one, we obtain mainly trivial rules, consisting of single attribute [5]. Therefore the proposed fitness function takes into account different measures, for example, a *length of individual* (a number of attributes). Evolution of individuals with high accuracy leads to broadening intervals of attribute values up to their whole domains. Obviously, it is an undesired feature of evolution, therefore it is necessary to punish individuals for too large *average amplitude of intervals*, where the *average amplitude of intervals* means an average length of intervals encoded in the evaluated individual. This part of fitness calculation concerns chromosomes representing continuous attributes. To protect against generation patterns covering the same training examples, we introduce into our fitness a penalty for *covering examples that are already covered*. At the end of each iteration (run of evolution), when we add the best individual (frequent pattern) to the set of discovered frequent patterns, we mark the examples covered by this pattern. Summarizing, our fitness function of ith individual is calculated according to equation 2:

$$f_i = cov_i - a \cdot mark_i - b \cdot Ampl_i + c \cdot nAtr_i , \tag{2}$$

where:

- cov_i is a number of examples covered by i_{th} pattern,
- $mark_i$ is a number of examples covered by i_{th} pattern that were covered by earlier patterns,
- $ampl_i$ is an average amplitude of intervals in i_{th} pattern,
- $nAtr_i$ is a number of attributes encoded in i_{th} pattern,
- a, b and c are the weights, assumed by a user.

The defined fitness function depends on a number of objective measures of individual's quality and on parameters defining the influence of these measures. It causes that the weights must be carefully adjusted in *EGAR*.

We have defined specialized genetic operators working on particular chromosomes. The *crossover* produces two offspring and the better one is selected for the next generation. Crossover of chromosomes containing continuous attributes is the same as in *GAR*: the first offspring receives all attributes from the first parent but ranges of intervals are summarized from both parents. The second chromosome of the first offspring is created by copying genes from the first parent and next, adding randomly selected genes from the second parent. When the selected gene (attribute) is present in the created chromosome, its value is determined randomly, taking a value from the first or the second parent with equal probability. The second offspring is produced similarly.

Mutation of chromosome with continuous attributes is a change of one or more genes by modifications of borders of intervals. We applied four possible mutations: shifting the interval on the left, on the right, increasing, and decreasing the interval. Probability of mutation is set close to 1%, usually one gene is mutated. *Chromosomes with discrete attributes* can be mutated using two kinds of changes: the first is replacing a value of attribute where the new value is taken from randomly selected training example – it takes into account a distribution of values of a given attribute in a training set. Such mutation can increase a support of the mutated individual (pattern). The second mutation is replaces the mutated attribute by a new discrete attribute with a random value (from its domain). This mutation allows for selecting rare values, which leads to discovery of patterns with lower support, but maybe interesting ones.

4. Experiments using real medical databases

Both methods of association rules generation on the basis of frequent patterns, namely *FPTree* and *EGAR* (described in the previous sections), were implemented and studied. The developed computer program allows for using data with discrete and continuous attributes.

4.1. The Antlia system

The **Antlia** (the name Antlia comes from astronomy, Antlia is the name of constellation) is a very flexible and user-friendly computer system. It consists of the three main modules (see Fig. 2):

1. data preprocessing module,
2. data mining module,
3. results presentation module.

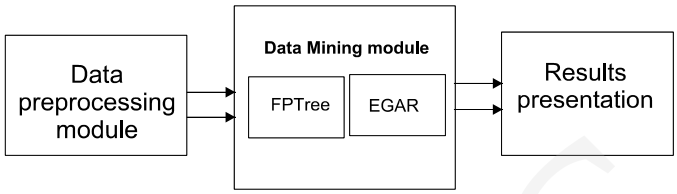

Fig. 2. The Antlia system – logical modules

Icons on a top of the main program window represent these parts – see Fig. 2, where in the left-upper part of the program screen we can see the icons *Data*, *Mining* and *Rules* – these names (as the whole **Antlia** program) are in Polish. Antlia allows for:

- data reading, preliminary filtering and sorting,
- data discretization,
- establishing a number of intervals for discretized domain,
- recording the preprocessed data,
- mining association rules using the *FPtree* and *EGAR* methods,
- setting all parameters of used mining method,
- observing influence selected parameters on an average fitness on a graph (only for the *EGAR* method),
- looking over, sorting and filtering the obtained rules,
- recording the obtained rules and their printing in a readable form (with preview option).

Antlia is written in the Microsoft Visual Studio .NET 2002 environment, in C++. To ensure high efficiency, all algorithms used in Antlia were optimised, using a profiler (a part of Compuware DevPartner Studio). A code of program is widely commented on according to the DoxyGene standard.

Data preprocessing module is responsible for reading data from the indicated file and their preliminary preparation for mining with using selected method. The program has defined own file format. This format is simple for users, one can export the data into a text file, which has to meet only two requirements:

1. the first line of the file has to contain names of attributes separated by tabulation (names can contain white characters, e.g., space),
2. each of the next line contains values of attributes separated by tab.

Read data are seen in a program window in the form of a table. The data can be sorted and filtered according to any attribute. This module has embodied the discretization method, a user can decide about discretization of all numerical attributes or only continuous ones.

Data mining module allows users to select the method (*FPTree* or *EGAR*), set all required parameters, and perform the mining association rules. For the *FPTree* users define minimal support and minimal confidence of generated rules. Possibility of indication attributes that should be considered in the

conclusion part of rules is an additional but a very useful option. If a value of an attribute selected for the conclusion part defines a class of the object we can perform the data clusterization task. The parameters of *EGAR* belong to the two groups: general – a minimal support and a number of searched frequent patterns, and specific for the genetic algorithm – a population size, a number of generation for searching a single frequent pattern, the weights for penalties and reward, and a mutation probability. The user has possibility of observing the influence of parameters on evolution character as a graph of average fitness – the right side of the screen window in Fig. 3. The left side of the screen contains the parameters setting dialog boxes (in Polish).

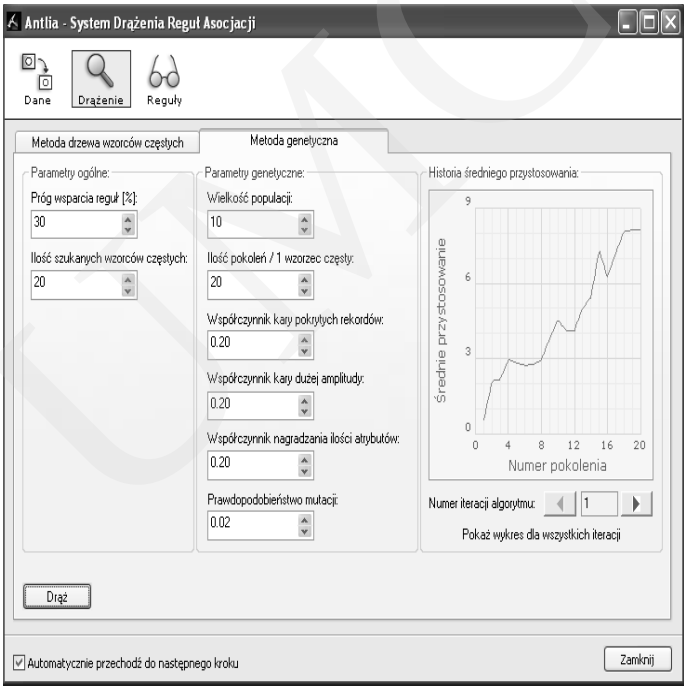

Fig. 3. Parameters setting in *EGAR*

The last module of **Antlia**, the presentation module, allows for review and verification of the obtained association rules. We can sort and filter the obtained rules according to their support and confidence. It is possible to write the filtered rules to the selected file and print them in a readable form.

4.2. Characteristics of databases used in experiments

We have experimented with two medical databases: *Sutek.xls* – it concerns breast cancer, and *Szyjka.xls* – cancer of the cervix uteri. Both databases were the subjects of our earlier studies: as a classification task using statistical

analysis [15]; evolutionary generation of classifier rules [4]; and data visualization [5]. Those studies were performed in collaboration with medical doctors, they were very helpful in goal defining and analyying the results.

Our databases, as almost all medical databases, are difficult for automatic knowledge acquiring, they contain missing and erroneous attributes which are of different types – both symbolic and numerical (discrete and continuous). The majority of data(examples) in medical databases describe so-called typical instances which favours generation of typical, commonly known knowledge, not the interesting one.

Sutek.xls database contains 101 records and 21 different attributes. The second database, *Szyjka.xls* consists of 530 records and 12 different attributes.

4.3. Analysis of the results

At the beginning, the FPTree method was used for experiments. It allows to generate all frequent patterns (and association rules) with assumed support and confidence.

FPTree results

Using *Sutek.xls*, without discretization and filtering, assuming minimal support and confidence as 70%, *FPTree* generates 340 association rules. An average support is 73% and confidence 93%. According to our expectations, majority of rules were confident but trivial, for example, 'if the illness has not recured then the patient is alive' or 'if the interview indicates that the patient's relatives did not have breast cancer, they also did not have any other cancer' (92% confidence). Discrete attributes were absent in the rules – there was not discretization. The assumed support was high, one can expect that such high support can have only typical instances, treated by medical doctors in a routine way. So, the next experiments have been performed assuming 50% support, and, to acquire not accidental knowledge, the confidence was enlarged up to 95%. After filtering the rules with the support higher than 70% (*FPTree* is deterministic, these rules were analysed in the previous experiment), there were 355 rules with the average support 56% and the confidence 99%. Majority of the rules concern dependency the attribute 'period of survive' from such attributes as a 'time of cancer resumption' and a 'degree of histopathologic maliciousness'. However, for laymen such knowledge can seems to be interesting, for specialists it is nothing new.

According to the interest of medical doctors found out in our earlier studies, we have included the attributes 'time of cancer resumption' in the conclusion part. A set of 56 rules has been produced with the minimal support and confidence 70% (average support 74%, confidence 96%). As previously, lots of rules were trivial, but more interesting appeared, e.g., a 'period of survive' is larger than five years when there was no 'recurrence' and 'relatives' free of the

cancer. The detailed analysis of the rules shows that they contain mainly one value of a 'period of survive' equal to at least five years. It is by reason of frequency this value in database – attribute 'period of survive' has the value equal to at least five years in 86% examples. Rules with a different value of this attribute are not able to obtain the assumed high support. We can remove these examples from the database, but the rest of database is too small for the data mining task. Another approach is to decrease the assumed support, but it causes an enormous number of generated rules. We applied 'by hand' discretization: for the two attributes – 'period of survive' and 'time of cancer resumption' we distinguished two intervals (values): one contained only one, maximal value, the other one covered all values. The obtained association rules attained 100% confidence and the support near 16%. This set of rules also contains trivial ones, but some of them seem to be more interesting. The precise evaluation of its importance requires consultation with medical doctors.

Next experiments were made with discretization of all continuous attributes. Discretization requires diminution of support (dividing a domain of an attribute into n sets causes that maximal possible support is equal to 100/*n* %), so we assumed that it is equal to 30%. The generated rules have 39% support and 88% confidence, they contain continuous attributes. The rule which seems to be worth detailed analysis by a specialist is: if 'the level of cancer development according UICC' is 2*b* and 'a size of tumour' is [3.5-7.5 cm] then 'the patient's relatives' have no cancer. Discretization of all numerical attributes causes generation of numerous rules. Such a large set of rules is difficult for analysis; it is hard to find interesting rules.

Experiments with the second database, *Szyjka.xls*, causes the similar problems as described above. As interesting we can recognize the rule: if a patient 'lives' in a town then the 'histopathologic kind' of cancer is *Ca plano* (confidence 91.18%). Values of numerical attributes are not recurrent, therefore they are absent in generated rules even when the support is diminished up to 25%. Proceeding in the same way as with the previous database, i.e., making discretization, reducing minimal support, and limiting a list of attributes for the conclusion part, *FPTree* produced a number of interesting rules, for example: if 'the second therapy' is radioactive isotope – radium and 'histopathologic kind' is *Ca plano* then 'patient survive' is dead.

***EGAR* results**

This algorithm requires setting a larger number of parameters which can be perceived as difficulty for users. But, on the other hand, *EGAR* works on all types of attributes, also continuous – the method is able to choose a number and boundaries of intervals. Visualisation of changes of average fitness during evolution helps in adjusting parameters.

As it was mentioned, a single evolution run gives one frequent pattern, the process is iterated, and a number of iterations is set as a parameter of the method. From the genetic algorithm point of view, important study concerns the sensitivity of *EGAR* to its parameters setting, more resistant algorithm is better because it releases users from parameters tuning phase. In the paper, there is no space to present all performed experiments. Here we focus only on the summary of results. *Sutek.xls* contains a number of numerical attributes, experiments with this data should reveal abilities of the method for tuning the number of intervals and boundaries. We are interested in rules with high confidence and not high support, focusing on finding interesting, new knowledge. To counteract the inclination of establishing broad intervals, covering all attribute domains, we have carefully set the influence of the average amplitude of intervals on the fitness function (weight b in equation 2). A number of experiments were conducted for tuning a penalty coefficient for covering previously covered examples. Observing the results of a huge number of experiments and taking into account the theoretical analysis of *EGAR* features, we can say that at the beginning of evolution the support and confidence play the main role in directing evolution, next, the rest evaluating factors, i.e., a number of examples covered by the evaluated pattern that were covered by earlier patterns ($mark_i$) and the average amplitude of intervals ($ampl_i$) of numerical attributes become decisive. Under the above condition *EGAR* allows to discover the rules with high support that are not generated by FPTree, for example, that patients 'having protein nm23' no more than six, 'survived' at least three years (support 92%).

The experiments have shown that proper values of penalty coefficients allow for efficient evolution. Setting too low the prize for the 'length of individual' (a number of attributes), we obtain relatively short rules which means that these rules are general but usually such rules do not contain interesting knowledge. Preferring a bit longer patterns (rules) and simultaneously allowing for relatively high penalty for large amplitude, we are able to obtain efficient evolution and interesting rules, containing numerical attributes.

We have conducted experiments using *EGAR* and database *Sutek.xls* with discretized numerical attributes. *EGAR* generates only 14 rules with high confidence, the evolution stops in local optimum, premature convergence of population was observed.

Experiments with both databases show that increasing the probability of mutation allows to eliminate the premature convergence. This observation is in agreement with theoretical premises, mutation is responsible for emerging new values of genes. On the other hand, randomness introduced by mutation of genes causes problems with proper direction of evolution.

5. Conclusions

All performed experiments, as well as thorough analysis of the studied methods, allow for the conclusion that *FPTree* is an effective method of association rules generation from the databases containing attributes with discrete values. One can obtain all frequent patterns relatively quickly which means – **all association rules with assumed minimal support and confidence**. *EGAR* does not guarantee finding all such rules which can be perceived as a weakness of the method. The other weakness is sensitivity of *EGAR* on the parameters setting. However, *EGAR* performs better with continuous and discrete attributes, it allows to point direction of searching by indication attributes that are desired to be present in the conclusion part of discovered rules. Elasticity of automatic developing of intervals and their boundaries for numerical attributes reveals advantage over artificial division of attribute domains on the constant, arbitrary chosen number of intervals. It allows for generation of more interesting rules with high support. Embodied (into the **Antlia** program) visualisation of evolution trajectory helps users adjust the parameters carefully.

Working with actual medical databases we must pay attention to specific character of these databases – they contain mainly typical cases (examples), therefore to discover interesting rules we must select the method of knowledge discovery and its parameter ensuring prevention from high promotion of rules support. Without it the obtained rules are usually general, commonly known and employed in knowledge of medical doctors. It is worth generating less general rules but with high confidence and not high support.

The developed tool can be used for association rules discovery from databases. It provides possibilities of mining method selection, taking decision about discretization process, filtering and sorting the data used for rules mining. Additionally, the discovered rules are presented in a convenient way; they can be sorted, filtered and organized for printing version with the print preview option. By indicating the attributes for the rules conclusion part, the **Antlia** system can be used as classification rules discovering tool. We plan to extend the system adding data visualization functionality – usefulness of data visualisation for the considered databases is proven in our previous study, presented in [5].

References

[1] Francisci D., Brisson L., Collard M., *A Scalar Evolutionnary Approach to Rule Extraction*. Laboratoire Informatique Sinaux et Systemes, (2003).
[2] Mao R., Yin Y., Pei P., *Data Mining and Knowledge Discovery*. Kluwer Academic Publishers, (2004).
[3] Mata J., Alvarez J.L., Riquelme J.C., *An Evolutionary Algorithm to Discover Numeric Association Rules*. Universidad de Huelva, (2001).

[4] Kwasnicka H., Markowska-Kaczmar U., Matkowski R., Dryl J., Mikołajczyk P., Tomasiak J., *Rule Discovery from Medical Data Using Genetic Algorithm*. Fourth International ICSC Symposium on Engineering of Intelligent Systems, Portugal, (2004).
[5] Kwaśnicka H., Markowska-Kaczmar U., Matkowski R., Dryl J., Mikołajczyk P., Tomasiak J., *Discovering Dependencies in Medical Data by Visualisation*. International ICSC Symposium on Engineering of Intelligent Systems, Portugal, (2004).
[6] Piatetsky-Shapiro G., Frawley W.: *Knowledge Discovery from Databases*. Cambridge MA, (1991).
[7] Han J., Kamber M., *Data Mining: Concepts and Techniques*. Morgan Kauf., (2000).
[8] Olivia Parr Rud, *Data Mining Cookbook*. Wiley Computer Publishing, John Wiley & Sons, Inc, (2001).
[9] Aggarwal R., Prasad V., *A tree projection algorithm for generation of frequent item sets*. Journal of Parallel and Distributed Computing, (2001).
[10] Goldberg D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, (1989).
[11] Kwaśnicka H., *Evolutionary computation in medicine*. in: Compendium medical informatics, Radoslaw Zajdel et all (Eds.,) [Bielsko-Biała]: Alfa Medica Press, corp. (2003) 365, in Polish.
[12] Freitas A., *A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery*. Pontificia Universidade Catolica do Parana, (2002).
[13] Lavington S.H., Freitas A., *Mining Very Large Databases with Parallel Processing*. Kluwer, (1998).
[14] Pena-Reyes C.A, Sipper M., *Evolutionary computation in medicine: an overview*. Artificial Intelligence in Medicine, 1 (2000) 1.
[15] Matkowski R., *Forecast value of estrogenic receptor and nm23 gene product in the cells of cancer. Data mining methods in medical applications and their correlation with selected clinical parameters*. Wroclaw Medical University, (2002), in Polish.