GENETIC ALGORITHMS IN DATA MINING

CSE 791-Knowledge Discovery in Databases and Data Mining Final Project

Lukasz Beca

1. Introduction 2

2. Data Mining 2

2.1. concept and purpose 2

2.2. Models used in data mining 3

3. Genetic algorithms 4

3.1. Motivation and purpose 4

3.2. Important concepts 4

3.3. Algorithm description 5

3.4. Other issues 5

4. Applications of genetic algorithms in data mining 6

4.1. Concept learning and classification 6

GABIL – concept learning system *7*

DOGMA (Domain Oriented Genetic Machine) system *8*

COGIN (Coverage-based genetic induction) system *10*

4.2. Clustering 11

4.3. Pattern Discovery 13

5. Advantages and disadvantages of Genetic Algorithms in context of Data Mining 15

5.1. Advantages 15

5.2. Disadvantages 15

7. Conclusion 15

8.References 15

1. INTRODUCTION

In recent years we can observe fast development of Knowledge Discovery field. Large amounts of data were gathered by companies or as a result of scientific research. Information that is contained in these data might be useful for gaining competitive advantage or lead to discovery of interesting relationships in researched fields. Unfortunately extracting information out of data is not always easy. Knowledge Discovery in Databases tries to address this problem.

Data mining is one of the stages of Knowledge Discovery process. It tries to extract interesting patterns out of provided data in order to create models. Those models are used to gain insight into the problem or to predict. Data mining uses many advanced methods to achieve this goal. Such techniques as statistical methods, visualization, OLAP, decision trees, association rules, neural networks and genetic algorithms are used. Each of these methods has its domain of applications. Genetic algorithms are applied for solving especially complex problems where data are noisy and traditional tools are not effective.

Genetic algorithms, as neural networks, follow the example of the processes that take place in nature. Using such concepts as survival of the fittest, population and mutation taken from biology, genetic algorithms offer powerful tool for solving whole array of optimization, search and machine learning problems. One of the application domains is also data mining. This paper discusses application of genetic algorithms in this area. The discussion is ordered as follows. Section 2 discusses shortly data mining. Short explanation of this concept is accompanied by description of the models and methods used for pattern search. Section 3 describes genetic algorithms. It explains main concepts such as coding, genetic operator and fitness function. It also explains how the genetic algorithm works. Section 4 discusses applications of genetic algorithms in data mining. The main focus is put on classification, concept learning and clustering problems. Section 5 describes advantages of genetic algorithms over other methods used in data mining. This section also discusses drawbacks of GAs. Section 6 tries to identify research issues that are currently pursued and that would make genetic algorithms more effective in data mining applications. Finally section 7 summarizes this paper and formulates conclusions.

Source: Data Mining by Adrianne

2. DATA MINING

2.1. CONCEPT AND PURPOSE

Currently huge amounts of data produced by companies and research institutions. Each company has its database system where necessary data are stored. In recent years size of such databases increased enormously. They contain much more records and fields than before. Moreover, the data usually are stored in format that is not appropriate for analysis. Commonly used database systems were designed and implemented long time ago when the data analysis was not a requirement. As a result, accessing the sought information in raw data become very difficult. Unfortunately manual analysis of data by experts is not feasible. Such process is too expensive, slow and subjective. New methods must be applied to solve this problem.

Data mining tries to address this problem. It is a part of the KDD process. Other steps include data cleaning, enrichment, coding, reduction and model interpretation. Data mining is defined as a discovery of interesting patterns in data. Such patterns can be used to create of models that in turn help to understand of underlying processes and provide insight into the problem. Such valid model can be used to in the decision process in the company. Models can be also employed in making prediction of future behavior. Currently data mining methods are used in such fields as marketing analysis, medical data analysis, defect prediction and many others. It uses many advanced techniques to achieve described goals. For example, machine learning methods such as neural networks and genetic algorithms are popular for finding interesting patterns in databases.

Source: Data Mining by Adrianne

2.2. MODELS USED IN DATA MINING

As it was mentioned in previous section. In data mining stage of the KDD process specific models are obtained. Such models can be divided into several categories.

Classification maps a data item into one of several predefined classes. Such model may be represented by decision trees, neural networks or classifiers based on genetic algorithms

Regression maps a data item to a real-value prediction variable. This is usually done by the linear regression models.

Clustering maps a data item into one of several classes in which the classes must be determined from the data. Clusters are defined by finding natural groupings of data items based on similarity of some property or metrics.

Summarization provides description of the subset of data. Summarization is usually carried out using statistical tools. Such statistics can be calculated as the mean or standard deviation.

Dependency models describe dependencies among data items. Using such models dependent and independent variables can be identified.

Link analysis described by association rules determines relations between fields in database.

Sequence analysis characterizes sequential patterns. States of the process or trends over time can be discovered using sequence models.

Depending on application different models can be applied in the data mining process. There is no universal approach and model function must chosen that describes analyzed in the most appropriate way to obtain desired results.

Genetic algorithms that will be described in following section are most often applied to construct classification and clustering models.

Source: Data Mining and Knowledge Discovery in Databases by Usama Fayyad

3. GENETIC ALGORITHMS

3.1. MOTIVATION AND PURPOSE

The genetic algorithms are search algorithms based on the mechanics of natural selection and genetics. Using survival of the fittest rule every generation is created using parts of the best specimens of the previous generations. This application of methods present in nature helped to create powerful tool that was successfully applied to most complex problems.

Genetic algorithms were developed by the John Holland at the University of Michigan. His work aimed at understanding of adaptive processes present in natural systems and creation of system software that uses such interesting properties of natural systems. Since this initial work genetic algorithms proved to be successful in many business, scientific and engineering applications. They are widely used for solving different search and optimization problems where traditional approaches failed.

Traditional search methods are very limited. They assume various properties of investigated functions e.g. existence of derivatives or continuity. As a result they perform well only on narrow classes of the problems and are not robust. On the other hand genetic algorithms do not make any assumptions about search space. They obtain similarly good results solving problems from different classes. They also perform well in noisy environments with many local optima. Such behavior is induced by the properties of genetic algorithms that make them different from other search methods. Genetic algorithms work with coding of the parameter set, not the parameters themselves. All operations are executed on strings that represent various solutions. Search is carried out using population of points, not one point. This approach makes the algorithm less sensitive to local behavior of the investigated function. Genetic algorithms use objective function, not derivatives. As a result no assumption must be made about search space and this makes this method applicable in many problems where traditional methods cannot be used. Finally, probabilistic transition rules are used, not deterministic. Genetic algorithms using random operators find the regions in the search space where the existence of optimal solutions is more likely. All those properties make genetic algorithms efficient search tool for many complex problems.

3.2. IMPORTANT CONCEPTS

The most important concepts used by genetic algorithms are coding, operators and fitness function. All of them are based on the examples taken from natural systems.

In order to solve problem using genetic algorithm the parameter set must be coded. The parameters must be coded as a finite-length string over some finite alphabet. Of course there are many ways to code the problem parameters. There is no strict rule that defines the coding appropriate for the specific class of problems. However, the choice of the proper coding is important because the effectiveness of the algorithm depends that. For the specific problem such coding should be chosen that operators applied to the string do not disrupt substrings that represent properties relevant to the problem. Such substrings are called building blocks and play important role in genetic algorithms. Even using such directions often it is not easy to find proper coding. Also, the choice of the alphabet is not determined. The general rule says that the user should select the smallest alphabet that permits a natural expression of the problem. As a result, many problems are coded with binary coding i.e. parameters are represented as binary strings because this alphabet provides the best results.

Another important concept is an operator. Genetic algorithms use various operators. The most important are reproduction, crossover and mutation operators. With those three operators many problems can be solved successfully.

During reproduction process the strings – members of a population are copied according to their fitness function value. The string with higher value of fitness function has higher probability of creating offspring. In this way, the strings of higher value will be represented in larger amount in the next generation. After reproduction, crossover takes place. The crossover is carried out in two steps. In first step the strings obtained from reproduction are mated at random. Then the position in the string is randomly selected and swapping all characters between selected position and the end of the strings creates two new strings. In this way the information contained in two strings is exchanged and potentially more optimal string is created. Mainly those two operators decide about the processing power of the genetic algorithm. Another operator, which is not so significant as the previous two, is a mutation operator. Mutation is the occasional random change of the value of a string position. It is carried out usually with small probability. In more complex problems more complicated operators are used such as multiple point crossover, inversion and reordering.

Another important concept in genetic algorithms is a fitness function called also objective function. It maps the string into some value. The rule is that the greater the fitness of the string the greater value is assigned by the objective function. Each member of the population is evaluated using fitness function during reproduction process. It is used to promote the best population members. It defines conditions of competitions in the population and promotes favored properties.

3.3. ALGORITHM DESCRIPTION

The simple genetic algorithm is composed of following steps.

1. The initial population is chosen randomly.
2. The reproduction, crossover and mutation operators are applied to the population.
3. New generation is obtained and step 2. is repeated till improvement in the population is not detected or required threshold of fitness is reached.

Most of the genetic algorithms follow this pattern.

3.4. OTHER ISSUES

- similarity templates: GA as a similarity templates processing

- computer implementation: costly

- mapping objective functions to fitness form: minimum to maximum, fitness scaling,

- other operators: multiple point crossover, inversion, reordering

- dominance and diploids

- niche and speciation

Source: Genetic Algorithms in Search, Optimization and Machine Learning by Goldberg

4. APPLICATIONS OF GENETIC ALGORITHMS IN DATA MINING

Genetic algorithms find many applications and are used to solve complex problem. They are also employed in data mining. Data mining provides a lot of challenging problems. The significant complexity of those problems results from several reasons. First, underlying functions are often nonlinear, with variables that have different relationships in different points of the space. Second, data are accompanied by noise i.e. errors in recording attribute value or classifying the example. Third, the scale of data mining problems is often enormous because of large number of dimensions and mixed data types. Following sections describe how the genetic algorithms are applied in several data mining methods such as concept learning, classification problems and other domains.

4.1. CONCEPT LEARNING AND CLASSIFICATION

The concept learning and classification problems are the favorite domains for the genetic algorithm researchers. Majority of applications involve construction of classification systems and this area is well researched. In data mining, concept learning and classification methods are used for prediction. Using constructed models composed of classification rules the class of the presented data item can be determined. This functionality forms basis for construction of decision models.

Concept learning and classification represent supervised learning methods. Concept learning can be described as teaching of the concept by presenting the positive and negative examples. The algorithm, using the set of examples, must construct the model of the concept. It is important that the obtained model is complete i.e. recognizes all instances of the concept. Another critical feature of the model is consistency. If the model is consistent it does not classify positively negative examples. Constructed model should be both complete and consistent in order to provide reliable results. Classification is a generalization of concept learning. Classification algorithm learns how to classify several concepts at the same time. As a result such algorithm is more complex and more difficult to design and implement. The concept must be encoded so that it can be presented to the algorithm. The choice of concept description language is important. Used language introduces bias that makes some classes of concepts easy to learn and others not.

Source: Data Mining by Adrianne and Using Genetic Algorithms for Concept Learning

Genetic algorithm use two basic approaches to solve classification problems. First of them, Michigan approach uses population of fixed-length elements to represent classification model. Each element represents one classification rule. The solution is represented by set of chromosomes from the population i.e. a population of individual rules constitutes the decision model. This methodology is also known as Classifier System and was developed by Holland. Typical Michigan-style algorithm works in following way. First, the random population of classifiers is generated. Each classifier consists of set of specific values specific value that must be matched by the attributes of the testing example, predicted class and strength. For the case to be classified, each element is inspected to determine if the case matches the classification rule (values for all attributes). Then one of the rules is selected randomly but with probability proportional to the strength. The selected rule determines class. During the training process the correct predictions are rewarded by adjusting strength of the rules. The weak rules are eliminated and new rules are created as a result of reproduction, crossover and mutation operators.

Source: quinlan

Another approach is the Pittsburgh method also called Learning System developed by Smith. In this method each individual represents a complete potential model which is composed of set of classification rules. Each model may contain different number of rules so potentially each individual has different length. It makes genetic operators more complicated because they must take into account the variable length of the chromosomes. In other aspects the Pittsburgh-style algorithm is similar to ordinary genetic algorithm.

Source: Competition-Based Induction of Decision Models from Examples by David Perry Greene, A Knowledge-Intensive GA for supervised learning

Training is an important issue in classification genetic algorithms. During the training phase the classification model is constructed so the decision about training method can be crucial for overall algorithm performance. The simples approach, called batch mode, presents all instances of the classified example data items to the system. On the contrary, the incremental approach presents only on of few cases to the system at a time. Also hybrid solutions are employed. Batch-incremental method presents one example to the learner. The learner trains using this example. Then, the obtained rule set is used to classify the next example. If the classification is correct, the next example is stored with the previous example and set of rules does not change. If the classification is wrong the GA is invoked to generate new rule set using two examples. So as each new additional instance is accepted, a prediction is made and if it is not correct the GA is rerun in batch mode with all examples. In this approach every instance works as testing and training instance.

Source: Using genetic algorithms for concept learning by Kenneth De Jong

The next sections provide several examples of real systems based on genetic algorithms that are used to solve classification problems.

GABIL – CONCEPT LEARNING SYSTEM

The GABIL (GA Batch-Incremental Concept Learner) is the system that learns and refines concept classification rules from its interaction with environment. The system uses such string representation of the concept that traditional GA operators can manipulate while preserving syntax and semantics of underlying concept description. It gives the possibility of using "off the shelf" genetic algorithm to find the solution to the concept learning problem.

The concepts are described by set of rules that are represented in disjunctive normal form. The left-hand side of each rule consists of a conjunction of one or more tests involving attribute value, the right-hand side indicates the concept or classification to be assigned to the element that matched the left-hand side of the rule. The restrictions are placed on number of conjunctions in the rule. Such restriction is necessary to obtain fixed-length encoding of the single rule. As a result more rules are required to express the concept. Each element of the conjunction is a test of following form:

If the value of attribute i of the example is in the given value set, return true, else return false

Then the rule is encoded as follows. Lets assume that F1 has values {small, medium, large} and F2 has values {sphere, cube, brick, tube}. Besides, we have two classes: widgets and gadgets. Then the rule has form:

If (F1 = small or medium or large) and (F2 = sphere) then it is a widget and is represented as a string of bits:

F1 F2 Class

111 1000 0

In GABIL system, the Pittsburgh approach used. Each individual is a variable-length string and represents an unordered set of fixed-length rules that define concept. As a result the crossover operation requires selecting two sets of two crossover points that may be different for each individual. During crossover operation the restriction is applied. The corresponding crossover points must match semantically i.e. if one parent is being cut on a rule boundary, the other must be cut on rule boundary as well. Mutation is carried out in standard way and performs bit level modification.

The fitness function involves only classification performance. It ignores length and complexity of the rules. The fitness of each individual describes the performance of the rule set on the set of training examples:

$$\text{fitness}(\text{individual i}) = (\text{percent correct})^2$$

Such function provides bias towards correct classification of all examples. It encourages consistency and completeness.

The algorithm employed by basic version of GABIL system is presented below.

```
procedure GA;
begin
    t = 0;
    initialize population P(t);
    fitness P(t);
    until (done)
        t = t+1;
        select P(t) from P(t-1);
        crossover P(t);
        mutate P(t);
        fitness P(t);
end.
```

The algorithm was trained using batch-incremental approach on two domains: one artificial and one natural. The artificial domain was designed to reveal behavior of the system when applied to incremental increases in target concept complexity. The natural domain was represented by database for diagnosing breast cancer. In the case of second domain the features described by numeric intervals the values were converted into nominal values e.g. age represented by {10-19, 20-29,…,90-99} was converted into {A1, A2, …A9} form. Such operation was forced by the representation of attribute values used by GABIL. The performance of the system was compared to other available classification tools such as AQ14, C4.5, ID5R, and IACL. It appears to be a good overall performer. It does not provide great results but performs uniformly well on large number of problem classes.

source: Using genetic algorithms for concept learning by Kenneth De Jong

DOGMA (DOMAIN ORIENTED GENETIC MACHINE) SYSTEM

problem description:

GAs are successful in avoiding local minima and produce near optimal solutions but require significant computational powers. On the other hand fast algorithms are available for creation of classification rules such as C4.5. However, fast algorithms are often trapped in local minima and do not provide optimal solutions. DOGMA is theory revision system that refines rules produced by fast classification algorithms. So in the first step the classification theory using rule-based learner is produced. In second step DOGMA takes as an input preclassified data and rule-based model of the data and tries to construct new, more accurate model. The completely new theory is built using substructures of the initial theory.

The rule is represented in first-order language similar to the language presented in description of GABIL system. The DOGMA also incorporates background knowledge by introduction of special background chromosomes. Genetic algorithm uses the hybrid Michigan/Pittsburgh approach to rule coding. On the lower level the chromosomes have fixed length and encode classification rules. On higher lever the chromosomes are grouped into families that represent classification models. The background knowledge is incorporated into models by introducing background chromosomes to the families.

used codings

DOGMA uses speciation of the chromosomes. Speciation is a method that encourages creation of the stable subpopulations of strings that serve different subdomains of a fitness function. This technique is used when the fitness function has several equally significant peaks and all of them should be found. In case of DOGMA system the chromosomes are divided into species according to which background chromosomes they may use. In this way each background chromosome represents single species. Information about species is used in several ways. First, it is used to control mating of chromosomes of different species. Second, grouping into families is carried out in such a way that chromosomes of the same species are not merged into the same family.

used operators

The algorithm uses wide array of genetic operators such as mutation, crossover, seeding and background seeding. Four types of crossover operators are used. Two-point crossover described in GABIL system, universal crossover, generalizing crossover and specializing crossover. Generalizing crossover works as follows. In two parent strings the predicates are selected and substrings are located in both strings that correspond to those predicates. Then, offspring are created by doing a bitwise ‘or’ operation on the located substrings. The specializing crossover works similarly but uses ‘and’ operator instead of ‘or’. The seed operator generates string that matches at least one training example. Such string is created from random string that is minimally modified to match the selected example. Background seeding is similar to seeding. It generates initial string by selecting random bits from background chromosome.

Apart from genetic operators, DOGMA uses so called family operators that allow creation of competing families. Such family operators allow to break or merge chromosome families. The break operator splits randomly family into two separate families. The join operator merges two families into a single family. Also make-family operator is used. It creates a family by selecting useful chromosomes of different species from the population. During the operation of the genetic algorithm first the genetic operators and then family operators are applied.

used fitness functions

The fitness function is constructed in such a way that the models that use description of minimal length for the model itself and the data are favored. Also consistency and completeness measures are incorporated into fitness function.

Algorithm

In order to evaluate DOGMA, the algorithm was applied to the rules created by C4.5 algorithm. The DOGMA system produced shorter descriptions than corresponding C4.5 rules and accuracy was improve nearly in all investigated domains.

source: GA-based Rule Enhancement in Concept Learning by Jukka Hekanaho

COGIN (COVERAGE-BASED GENETIC INDUCTION) SYSTEM

problem description

Another classification system. Simultaneously promotes competition in various classification niches within the model and constrains model complexity. So it tries to achieve model accuracy and simplicity. Model is constructed by induction from a set of pre-classified training examples. The organizing principle is the use of training set coverage, limitations on model complexity and diversity in the model. The system does not make any assumption about number of necessary rules at this point it differs from other system Michigan-style classifiers.

Pre-classified data sets are used for training. Pre-classified data denote set of examples with assigned discrete class. Given such set, the algorithm must construct rules that divide the data items into desired classes and will form basis model used for prediction. Such decision model has usually form of IF-THEN rules.

Predictive ability of the model can be measured during training by accuracy of the classification. Simplicity of the model is assured by preference for smaller models.

The fundamental organizing principle is the concept of a coverage-based constraint. According to this constraint the examples are treated as niches in ecology. The exact number of niches is unknown but it should be smaller than the number of examples. The algorithm tries to find the model that classifies correctly and requires the fewest niches. So two spaces are searched simultaneously: the number of niches and the best individual within each niche.

The system model is represented as a population of fixed length rules. The population size varies from generation to generation. This behavior is enforced by the function of coverage constraint that is applied to each generation. Such function works as follows. In the competition step of each algorithm cycle all candidate rules are ordered by fitness. Then the coverage-based filter allocates the training examples to rules that correctly classified them in fitness order. When all examples are covered, the rest of the rules are discarded and new decision model is created. In next step, the new model is evaluated using the training set and if it achieves the best score in the search it is copied for future reference. When the search ends, the best model is taken as the final model.

used codings

System uses synthesis of Michigan and Pittsburgh approach to rule coding. The decision model is represented by set of fixed-length strings with coding similar to one described in GABIL system. One of the extensions to the coding are additional codes for classes.

used operators

The reproduction and single-point crossover is applied to each generation of rules. The fitness function is based on lexicographically-screened entropy and is composed of two components. First component favors correct rules. Second component – information gain of the rule – favors high discriminability of the rule. All the rules – those from current model and those generated using crossover operators – are subjected to the fitness test. Weaker rules are eliminated by the coverage-based filter. The initial model is constructed by randomly generating rules till all examples from training set are covered so already at the beginning of the search all the niches are covered. The rules are later refined by the application of the algorithm.

The application of the COGIN system to the test cases shows its good performance especially for complex problems compared with traditional classification tools(CN2 and NewID).

source: Competition-Based induction of decision models from Examples By David Greene, Using Coverage as a Model Building Constraint in Learning Classifier Systems

DELVAUX system

problem description

used codings

used operators

used fitness functions

algorithm

results

source: Learning Bayesian classification rules through genetic algorithms by Christoph Eick

4.2. CLUSTERING

Another interesting application of genetic algorithms is data clustering. In the clustering problem the natural groupings of data are searched for. The analysis can be carried out in many dimensions at the same time and can be based on different metrics. Usually the attributes of data must be scaled properly in order to obtain reliable distance measures in the search space. Because of that, the results of clustering algorithms depend to the large degree on data encoding. Using clustering algorithm, a data miner can discover interesting subsegments in data space and information about dependencies among data attributes. Genetic algorithms perform well in the search for clusters in data because of their natural ability to deal with complex search spaces and noise. As a objective function various metrics can be used and analysis can be carried out in parallel manner in arbitrary number of dimensions.

One of the approaches treats the clustering problem as an instance of more general, grouping problem. Grouping problem can be defined as partitioning of a set of objects into a collection of

mutually disjoint groups of objects. The objective of grouping can be defined as an optimization of a cost function defined over the set of all valid groupings.

For solving such problems specific encoding was proposed that can be used by genetic algorithm. Each chromosome represent one clustering solution. It is composed of two parts. The object part encodes each object on one gene. For example chromosome ABBDE would encode solution where the first object is in group A, second and third in B, fourth in D and fifth in E. The group part encodes the groups on a one gene for one group basis. For example it has form EDBA for object part presented above. This string means that all objects are divided among groups E, D, B, and A. The target chromosome has form ABBDE:EDBA. Such method of coding promotes significance of grouping in the search process. Another consequence of the proposed encoding is variable length of the chromosomes.

With such designed encoding the operators work only with group part of the chromosomes. The object part only identifies which object belongs to which group and is not modified directly by the operators. However, it will be changed as a result of operations on the group part of the chromosome. The crossover operator fits following pattern.

1. Two crossing points are selected in group part of the chromosome. They delimit the crossing section in each of two parent strings.

2. Content of the crossing section of the first parent string is injected at the first crossing point of the second parent string. This operation has meaning of moving some of the groups from the first parent string into the second.

3. All objects occurring twice are eliminated from the groups they were members of in the second parent. New groups have higher priority and as a result some of the groups present in the second parent are changed.

4. If necessary, the resulting groups are adapted according to the constraints and the fitness function.

5. Steps 2, 3 and 4 are applied to the parent strings with their roles reversed in order to obtain second child.

The mutation operator should create new groups of eliminate an existing group. Also for some applications the objects may be shuffled randomly among their groups. Another operator that can be used is the inversion operator. It is also applied to the group part of the chromosome. Its purpose is to facilitate transmission of good groupings from parents to offspring. Potentially it places closer genes that represent good clustering and in this way probability of transmitting them during crossover operation is greater.

With coding and operators defined as above number of grouping problems can be solved. Those concepts can be also applied to clustering where groups correspond to data clusters. Fitness function can be defined in such a way that it promotes the minimal distance between objects in one cluster. Then, using presented coding, operators and fitness function in basic genetic algorithms we can find optimal clustering of data.

source: A new representation and operators for genetic algorithms applied to grouping problems by Emanuel Falkenauer

Different and simpler approach is presented in "Data Mining". The genetic algorithm is used there to find good clustering of data about magazines subscription. Clustering is described there as a set of guide points in the space. The clusters are limited by borders, which defined as lines equidistant from both guide points. Graphical representation of such clusters is called Voronoi diagram. Using described approach the space may be divided into several regions using guide points. The coordinates of the five guide points are used to encode a potential solution to the problem. The solution string has fixed length i.e. one individual consists of ten real numbers. It means that the algorithm tries to find five clusters in presented data. In initial stage, twenty individuals are chosen randomly. Following operators are used. The crossover operator takes three points from one individual, two points from other and exchanged them. The mutation operator generates new points and places in randomly chosen individual.

As a fitness function the average distance of the points in the sample space to the closest guide point is used. The best clustering yields minimal value of the distance. The fittest individuals have lowest value of the distance function. After certain numbers of generations, stable clustering is found.

Source: Data Mining by Adrianne

4.3. PATTERN DISCOVERY

Another domain of genetic algorithm applications is pattern search in large databases. During pattern discovery process, the algorithm tries to extract interesting information from data. In undirected search no specification of desired rules is necessary. Such approach is also known as pure data mining. The algorithm tries to find relevant information by itself. This approach can be employed when no background knowledge about data is available. Then using extracted information more focused data processing can be applied. For this task the directed approach is more appropriate. It analyzes in depth interesting patterns and tests proposed models.

The GA-MINER tool presented in … allows both undirected and directed analysis of data. The system is implemented in RPL2 (Reproductive Plan Language). RPL2 is a language, interpreter and run-time system for the implementation of stochastic search algorithms with special focus on evolutionary algorithms.

The system uses the concept of pattern template in order to characterize the degree of freedom the system has in pattern search. The pattern template can be used to constrain the system to particular forms of patterns. This concept makes directed search possible by tight restrictions imposed on parts of the pattern template. In the most restricted form the system is used for hypothesis testing and refinement which is achieved by seeding the system with the hypothesis.

The basis for a pattern representation is the subset description. Syntax of the subset description is defined by grammar that allows expression of specific attribute-values or attribute-range constraints. The attribute constraints can be aggregated in sentences with logic operators 'or' and 'and'. Patterns are constructed from a number of such subset descriptions. The explicit pattern rule has form "if C then P" or "when S, if C then P" where C represents the condition, P represents the prediction of a rule and S represent additional condition. Subset descriptions are manipulated directly by the genetic algorithm

As a fitness function a number of functions are applied. Among them is information gain function. Special care is taken to discard the tautologies relatively early in the search. Also pattern templates are used to steer the system away from the obvious rules. Genetic algorithm applied in GA-MINER encourages diversity and populating of local niches by choosing strings for genetic operations from local neighborhoods. In this way, the several areas of search space are investigated at

the same time and several patterns can be found in single run. The crossover and mutation operators are complicated and must understand complicated structure of subset description. At the end of the search process the gathered rules are presented to the user. The system was successfully used in analyzing in searching patterns from databases of industrial companies.

Source: A Genetic Algorithm – based approach to Data mining by Ian Flockhart

deriving queries from the results

Source: Deriving Queries from Results using Genetic Programming by Tae-Wan Ryu

5. ADVANTAGES AND DISADVANTAGES OF GENETIC ALGORITHMS IN CONTEXT OF DATA MINING

5.1. ADVANTAGES

The genetic algorithms are powerful search and machine learning tools that are successfully applied in data mining domain. The main advantages are the robustness and uniform approach to large number of different classes of problems. It the solution for given problem exists the genetic algorithm with proper coding, operators and fitness function will find it. This is an obvious advantage over other methods such as regression models that can be used only in specific cases. Such generality is desirable in data mining where the search spaces are complex and contain noise. Neural networks are similar in this sense. They also do not need any assumptions about the search space, but the solutions found by genetic algorithms are more readable and relatively easy to understand. Another important feature of genetic algorithms is their easy adaptation to changing conditions in the system. The training simply can be continued during the regular operation. Such features cannot be obtained by traditional methods such as decision tree algorithms.

5.2. DISADVANTAGES

Unfortunately genetic algorithms have also several drawbacks. First of all, significant computing power is required to obtain relevant results in real life applications. Traditional tools are simply faster. Such poor performance is caused by large over-production of individuals and the random character of the search process. Also genetic algorithms do not obtain so good result as specially designed tools. They are robust and behave similarly well in many domains but methods customized for specific class of problems can yield better results. Moreover, it is difficult to find good encoding for complex problem. Also definition of proper fitness function and operators is not easy. Determining all those elements properly is part of the art. As a result current solutions applied in data mining are still in research stage. No methodology is available which would make application of genetic algorithms straightforward for real life data mining problems.

Source: Competition-based induction of decision models from examples by David Greene, Data mining by Adrianne

7. CONCLUSION

8.REFERENCES AND PAPERS USED IN THE PROJECT

Adriaans, P and Zantinge, D. "Data Mining", Addison-Wesley Pub. Co, 1996

Fayyad, U and Uthurusamy, R. "Data Mining and Knowledge Discovery in Databases", ACM Communications Vol 39, No.11, 1996

Goldberg, D. E.. "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wessley Pub. Co., 1989

De Jong K. A., Spears W.M., Gordon D.F.. "Using Genetic Algorithms for Concept Learning", Machine Learning 13, 161-188 (1993)

Quinlan, J. R.:"C4.5 Programs for Machine Leaning, Morgan Kaufmann, 1993

Greene, D. P.: "Competition-Based Induction of Decision Models from Examples", Machine Learning 13, 161-188 (1993)

Janikow, C. Z. "A Knowledge-Intensive Genetic Algorithm for Supervised Learning", Machine Learning 13, 161-188 (1993)

Hekanaho, J.:"GA-based Rule Enhancement in Concept Learning" KDD-97

Greene, D. P., Smith S. F. :"Using Coverage as a Model Building Constraint in Learning Classifier Systems", Evolutionary Computation 2(1):67-91, 1994

Greene, D. P., Smith S. F. :"Competition-Based Induction of Decision Models from Examples", Machine Learning 13, 229-257 (1993)

Falkenauer, E "A New Representation and Operators for Genetic Algorithms Applied to Grouping Problems" , Evolutionary Computation 2(1):123-144, 1994

Flockhart, I.W., Radcliffe, N.J.:"A Genetic Algorithm – Based Approach to Data Mining", Genetic Algorithms