Applying Evolutionary Algorithms to Discover Knowledge from Medical Databases

Man Leung Wong
Department of Information Systems
Lingnan College
Tuen Mun
Hong Kong
mlwong@ln.edu.hk

Wai Lam
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong
wlam@se.cuhk.edu.hk

Kwong Sak Leung
Department of Computer Science and Engineering
The Chinese University of Hong Kong
ksleung@cse.cuhk.edu.hk

Jack C. Y. Cheng
Department of Orthopaedics and Traumatology
The Chinese University of Hong Kong
jackcheng@cuhk.edu.hk

ABSTRACT

Data mining has become an important research topic. The increasing use of computer results in an explosion of information. These data can be best used if the knowledge hidden can be uncovered. Thus there is a need for a way to automatically discover knowledge from data. In this paper, new approaches for knowledge discovery from two medical databases are investigated. Two different kinds of knowledge, namely rules and causal structures, are learned. Rules capture interesting patterns and regularities in the databases. Causal structures represented by Bayesian networks capture the causality relationships among the attributes. We employ advanced evolutionary algorithms for these discovery tasks. In particular, Generic Genetic Programming is employed as rule learning algorithm. Our approach for discovering causality relationships is based on Evolutionary Programming which learns Bayesian network structures.

1. Introduction

Data mining has become an important research topic [2]. The research in this area can be useful for a lot of real world problems. For instance, medical domain is a major area for applying data mining. With the computerization in hospitals, a huge amount of data has been collected. It is beneficial if these data can be analyzed automatically.

In this paper, we will introduce our approaches for discovering knowledge from two medical databases. Two different kinds of knowledge, namely rules and causal structures, are learned. Rules capture interesting patterns and regularities in the database. Causal structures represented by Bayesian networks capture the causality relationships among the attributes. We employ advanced Evolutionary Algorithms [6] [7] [3] for these discovery tasks. In particular, Generic Genetic Programming is employed as rule learning algorithm. Our approach for discovering causality relationships is based on Evolutionary Programming which learns Bayesian network structures. To handle continuous attributes, we employ Genetic Algorithm to find a good discretization policy.

This paper is organized as follows. Section 2 introduces the two medical databases we have analyzed. The tasks of learning rules and Bayesian networks are also introduced in this section. Section 3 describes our approach for rule learning. Sections 4 and 5 describe our approach for Bayesian network learning which is composed of two layers. Section 4 details the inner layer that learns Bayesian network structures from discrete variables, and Section 5 details the outer layer that discretizes continuous variables while learning network structures. The learning results are presented in Section 6. Finally Section 7 is the conclusion.

2. The Learning Tasks

2.1. Medical Databases

Our learning targets are two medical databases from the Orthopaedic Department of the Prince of Wales Hospital of Hong Kong. The first database, the fracture database, consists of records of children with limb fractures admitted to the hospital in the period 1984-1996. This data can provide information for the analysis of children fracture patterns. This database has 6500 records and 8 attributes which are listed in Table 1.

Name	Type	Explanation
Sex	Nominal	Sex
Age	Numeric	Age
Admday	Date	Admission date
Stay	Numeric	Length of staying in hospital
Diagnosis	Nominal	Diagnosis of fracture
Operation	Nominal	Operation
Surgeon	Nominal	Surgeon
Side	Nominal	Side of fracture

Table 1: Attributes in the fracture database

The second database contains clinical records of Scoliosis patients. Scoliosis refers to the spinal deformation. A Scoliosis patient has one or several curves in his spine. Among them, the curves with severe deformations are identified as major curves. The database stores measurements on the patients, such as the number of curves, the curve locations, degrees and directions. It also records the maturity of the patient, the class of Scoliosis and the treatment. The database has about 500 records and the attributes are shown in Table 2.

2.2. Rule Learning

We investigate the task of discovering rules from these two databases. We make use of a rule representation which is easily understandable rule is a sentence of the form ``if *antecedents*, then *consequent*''. The antecedents are specifying certain characteristics of attributes. In general, the antecedent part is a conjunction of descriptions about attributes, while the consequent is a descriptor for a single attribute. Rule learning is the process of inducing rules from a set of training examples.

0-7803-5731-0/99$10.00 ©1999 IEEE

The accuracy or the confidence of a rule is the probability that the consequent occurs under the condition that the antecedents occur. If the accuracy is 100%, the rule is an exact rule. If the accuracy is near 100%, the rule is a strong rule. If the accuracy is not high but is already much larger than the average probability of the consequent, then the rule is a weak rule. A data mining approach should not only discover exact or strong rules. Because weak rules may also provide precious knowledge to analysts.

Name	Description
Sex	Sex
Age	Age
Lax	Joint Laxity (integer between 0 and 3)
1stCurveT1	Whether 1st curve started at vertebra T1
1stMCGreater	Whether the degree of 1st Major Curve is greater than the 2nd Major Curve
L4Tilt	Whether vertebra L4 is tilted
1stMCDeg	Degree of 1st Major Curve
2ndMCDeg	Degree of 2nd Major Curve
1stMCApex	Apex of 1st Major Curve
2ndMCApex	Apex of 2nd Major Curve
Deg1	Degree of 1st Curve
Deg2	Degree of 2nd Curve
Deg3	Degree of 3rd Curve
Deg4	Degree of 4th Curve
Class	Scoliosis Classification (K-I, K-II, K-III, K-IV, K-V, TL or L)
Mens	Period of Menstruation
TSI	Trunk Shift (in cm)
TSIDir	Trunk Shift Direction (null, left or right)
RI	Risser Sign (integer between 0 and 5), which measures the maturity of the patient
Treatment	Treatment (observation, surgery or bracing)

Table 2: Attributes in the Scoliosis database

2.3. Bayesian Network Learning

A Bayesian network [5] is a different model to represent probabilistic knowledge of data. It is a formal knowledge representation supported by the well-developed Bayesian probability theory. It captures the conditional probabilities between variables (i.e. attributes in the database), and focuses on the causality relationships between variables. In many real-life situation, the data cannot be described completely by a few rules. Building a complete model for such a database is difficult and usually results in a complicated model. A Bayesian network can be a complement to rules. Due to the graphical representation, a Bayesian network is easily understandable. It has a well-developed mathematical model and can be used to perform reasoning under uncertainty.

Formally, a Bayesian network is a directed acyclic graph. Each node represents an attribute and each edge represents a dependency between two nodes. An edge from node *A* to node *B* can represent a causality conveying the fact that the value of *B* depends on the value of *A*. The value of each variable should be discrete. Each node is associated with a set of parameters. Let N_i denote a node and $\prod N_i$ denote the set of parents of N_i. The parameters of N_i are conditional probability distributions in the form of $P(N_i|\prod N_i)$, with one distribution for each possible instance of $\prod N_i$. Figure 1 is an example Bayesian network structure showing the causality relationships among the attributes in a medical domain concerned with ``blue" baby diagnosis.

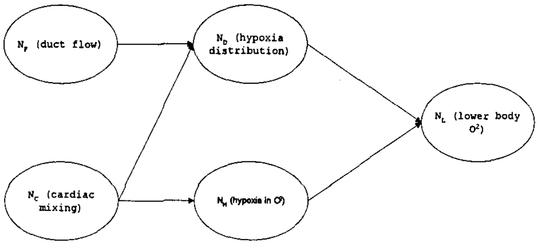

Figure 1: A Bayesian network structure in a ``blue" baby domain

The main task of learning a Bayesian network is to automatically find directed edges between the nodes, such that the network can best describe the causalities. Once the network structure is constructed, the conditional probabilities are calculated based on the data.

3. Rule Learning using Generic Genetic Programming

We employ an advanced evolutionary algorithm called Generic Genetic Programming (GGP) to discover rules from a database. GGP [13] is an extension to Genetic Programming which uses a grammar to control the structures being searched. A grammar is provided by the user as a template for rules. A set of rules is derived by using this grammar and forms the initial population. Then, the main loop of GGP is entered. In each generation, individuals are selected stochastically to evolve offspring by the three genetic operators: crossover, mutation and dropping condition. In each generation, the number of new individuals evolved equals to the population size. Thus the total number of individuals in the population is doubled. All individuals participate in the token competition and the replacement step, so as to eliminate similar rules and increase the diversity. One half of the individuals with the higher fitness scores after token competition are retained and passed to the next generation. The whole process iterates until the maximum number of generations has been reached.

3.1. Grammar

The initial set of rules are created based on a grammar. The grammar of GGP governs the structures to be evolved. It serves as a template for the rules. The initial population is created by randomly `filling' in this template. GGP will then search for the best set of rules without violating the grammar.

The grammar specifies that a rule is of the form ``if *antecedents* then *consequent*". It specifies which attributes can appear in the antecedent part and which attributes can appear in the consequent part. It also specifies the descriptors used to describe each attribute. The rule formats in various problems can be different. Thus for each problem, a specific grammar is written so that the format of the rules can best fit the domain.

The use of grammar provides a powerful knowledge representation and allows a great flexibility on the rule format. Rules with the user desired structure can be learned because the user can specify the required rule format using the grammar.

3.2. Genetic Operators

In rule learning using GGP, the search space is explored by generating new rules using three genetic operators. The genetic operators change the attribute descriptors in order to search for better rules.

Crossover produces one child from two parents. One parent is designated as the primary parent and the other one as the secondary parent. A part of the primary parent is selected and replaced by another compatible part from the secondary parent. The offspring produced must be valid according to the grammar.

Mutation is an asexual operation. A part in the parental rule is selected and replaced by a randomly generated part. The offspring has to be valid according to the grammar, thus a selected part can only mutate to another part with a compatible structure.

Dropping condition is an genetic operator for rule learning, to avoid subsumed rules. The rules evolved in GGP may be too restrictive and include redundant constraints. Dropping condition [10] is used to generalize rules. A rule can be generalized if one descriptor in the antecedent part is dropped. Dropping condition selects randomly one attribute descriptor, and then turns it into `any'. That particular attribute is then no longer considered in the rule.

3.3. Evaluation of Rules

An evaluation function based on the support-confidence framework [1] is developed as the fitness function in our rule learning approach. *Support* measures the coverage of a rule. *Confidence factor (cf)* is the confidence of the consequent to be true under the condition that the antecedents are also true. For a rule `if *A* then *B*' and with a training set of *N* cases, support is $|A\&B|/N$ and confidence factor is $|A\&B|/|A|$.

When evaluating the confidence of a rule, we need to consider the average probability of consequent (*prob*). The value *prob* is equal to $|B|/N$. We defined cf_part as

$$cf\_part = cf \times (\frac{cf}{prog})$$

The log function measures the order of magnitude of the ratio *cf/prob*. A high value of *cf_part* requires simultaneously a high value on *cf* and a high value on the ratio *cf/prob*.

Support is another measure to be considered. If *support* is below a user-defined minimum threshold (*min_support*), the confidence factor of the rule is based on a small number of training examples, and we just ignore the confidence factor.

Our fitness function is defined to be:

$$raw\_fitness = \begin{cases} support, & if\ support < min\_support \\ w_1 \times support + w_2 \times cf\_part, & otherwise \end{cases}$$

where the weights w_1 and w_2 are user-defined weights used to control the balance between the confidence and the support in learning. These two values have been set to 1 and 8 respectively.

3.4. Token Competition

The token competition [9] technique is employed in our rule learning approach to search for a *set* of rules instead of just one rule. The concept is as follows: In the natural environment, once an individual has found a good place for living, it will try to exploit this niche and prevent other newcomers to share the resources, unless the newcomer is stronger than it is. The other individuals are hence forced to explore and find their own niches. In this way, the diversity of the population is increased, so that good individuals in different niches are maintained.

Based on this mechanism, we assume each record in the training set can provide a resource called token. If a rule can match a record, it set a flag to indicate the token is seized. Other weaker rules then cannot get the token. The priority of receiving tokens is determined by the strength of the rules. A rule with a high score on *raw_fitness* can exploit the niche by seizing as many tokens as it can. The other rules entering the same niche will have their strength decreased because they cannot compete with the stronger rule. The fitness score of each individual is modified based on the token it can seize. The modified fitness is defined as :

$$modified\_fitness = raw\_fitness \times count\ /\ ideal$$

where *raw_fitness* is the fitness score obtained from the evaluation function, *count* is the number of tokens that the rule actually seized, *ideal* is the total number of tokens that it can seize, which is equal to the number of records that the rule matches.

4. Learning Bayesian Networks from Databases

Besides from learning rules from data, we have developed an approach to learn Bayesian network structures from discrete variables. The approach is based on Evolutionary Programming (EP) and the Minimum Description Length (MDL) principle. The MDL principle has been applied on Bayesian network learning in our previous work [8]. The principle [11] states that the best model of a collection of data is the one that minimizes the sum of the encoding lengths of the data and the model itself. The MDL metric is defined in [8] to measure the *total description length DL* of a network structure *G*. The total description length of a network is the sum of description lengths of each node. This length of each node is defined based on two components, the *network description length* and the *data description length*. The first part is the description length for encoding the network structure, which measure the simplicity of the network. The second part is the description length for encoding the data, which measure the accuracy of the network.

To search for a good network structure, we develop an approach called MDLEP [12] which uses Evolutionary Programming to optimize the MDL metric, so as to learn the best Bayesian network structure. A Bayesian network is a directed acyclic graph (DAG). A set of DAGs is randomly created to make up the initial population. Each DAG is evaluated by the MDL metric. Then each DAG produces a child by performing a number of mutations. The child is also evaluated by using the MDL metric. The next generation of population is selected among the parents and children by tournaments. One half of DAGs with the highest tournament scores are retained for the next generation. The process is repeated until the maximum number of generations is reached. The network with the lowest MDL score is output as the result.

Offspring in EP are produced by using a number of mutations. The probabilities of using 1, 2, 3, 4, 5 or 6 mutations are set to 0.2, 0.2, 0.2, 0.2, 0.1 and 0.1 respectively. The mutation operators modify the edges of a DAG. If a cyclic graph is formed after the mutation, edges in the cycles are removed to keep it acyclic. The approach uses four mutation operators, with the same probabilities of being used:

1. Simple mutation randomly adds an edge between two nodes or randomly deletes an existing edge from the parent.
2. Reversion mutation randomly selects an existing edge and reverses its direction.
3. Move mutation randomly selects an existing edge. It moves the parent of the edge to another node, or moves the child of the edge to another node.
4. Knowledge-Guided mutation is similar to simple mutation, but the MDL scores of the edges guide the selection of the edge to be added or removed. The MDL metric of all possible edges in the network is computed before the learning algorithm starts. This mutation operator stochastically adds an edge with a small MDL metric to the parental network or deletes an existing edge with a large MDL metric.

5. Discretizing Continuous Variables while Learning Bayesian Networks

Bayesian network can only represent discrete variables. One approach to handle the databases with continuous variables is to discretize them first. The continuous variables are usually discretized by thresholds specified by human. However, different discretization policy will produce different network structure. The causality will be lost if the discretization is not suitable. Thus it is desirable to search for the best discretization policy before Bayesian networks are induced.

A *discretization sequence* λ defines a function that maps a continuous variable to a discrete variable. Each discretization sequence contains a list of threshold values. The variable will be discretized according to the range specified by the thresholds. A *discretization policy*, $\Lambda = \{\lambda_i : X_i \text{ is continuous}\}$, is a collection of discretization sequences for each continuous variable. The policy defines a new set of variables $U^*=\{X_1^*,\ldots, X_n^*\}$ where $X_i^*=f_{\lambda_i}(X_i)$ if X_i is continuous and $X_1^*= X_i$ otherwise.

5.1. MDL for discretization policy

Friedman and Goldszmidt extend the MDL score to evaluate the discretization policy while learning Bayesian network structures [4]. They have also described a greedy approach for learning the discretization policy and Bayesian networks [4]. The approach learns the discretization policy and network structures alternatively. It starts with an initial discretization policy and learns Bayesian networks from the discretized data set by using the MDL metric. Based on the learned structure, a discretization policy is learned by using the MDL metric. In learning the discretization policy, only one variable is re-discretized at a time. The discretization sequence of this variable is reset to empty (i.e. no threshold values) first. The greedy approach searches for the `split' that gives the largest decrease in the MDL metric. The process is repeated until there is no improvement.

However, the greedy search algorithm can be easily trapped in a local optima. This approach also greatly depends on the initial settings. If the initial guess of discretization policy or network structure is not good, the result can be poor.

5.2. Learning Discretization Policy using Genetic Algorithms

A Genetic Algorithm (GA) is applied to optimize the new MDL metric, and thus the best network structure as well as the best discretization policy can be learned. It is less likely that the algorithm will be trapped in a local optima, because there is a population of individuals to explore the search space in parallel.

Our approach starts with an initial discretization policy. MDLEP is then used to learn the best network structure. Based on this structure, GA is used to learn the best discretization policy. The process is iterated until the maximum number of iterations is reached.

The genetic algorithm starts with an initial randomly generated population. Each individual in the population is evaluated by the new MDL score defined in [4]. The good individuals are selected to produce offspring using the genetic operators. The offspring in turn produces the next generation until the maximum number of generations is reached.

6. Learning Results

6.1. Fracture Database

6.1.1. Results of Bayesian Network Learning

The relationships among the attributes are analyzed by learning a Bayesian network. We have used a population size of 50 for both MDLEP and GA. The discovered network structure is drawn in Figure 2. Day, Month, Weekday and Year refer to different parts of the admission date. The age is discretized into 0-4, 5-9, 10-12 and 13-16. The day and month are discretized into just one range, which means that they are not involved in any relationship in the Bayesian network. Year is divided into 3 ranges. Stay is divided into 3 ranges.

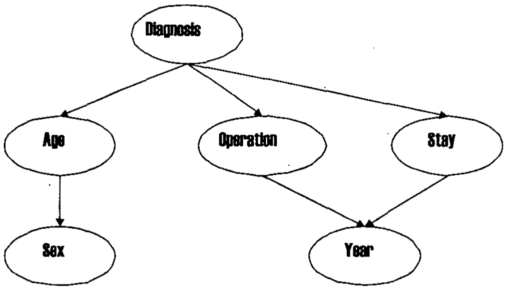

Figure 2: The discovered network structure for the fracture database

From the network structure constructed, the following interesting relationships are observed:

- The value of Diagnosis affects the values of Operation and Stay. Different fractures are treated with different operations, and require different time for recovery.
- The value of Diagnosis affects the value of Age. Some fractures are more frequently occurred in particular age groups.
- The value of Age affects the value of Sex. It is observed that the young patients are more likely to be female, and elder patients are more likely to be male.
- The value of Operation and Stay affects the value of Year. It is observed from the database that the length of stay in hospital is longer in the year 1985, 1986 and 1994, and open-reduction occurs more frequently for earlier years.

6.1.2. Results of Rule Learning

Based on the learned Bayesian network, we observe a causality model between diagnosis, operation and stay. We wish to learn knowledge about these attributes. Firstly, sex, age and admission date are the possible causes of diagnosis. Secondly, these three attributes and diagnosis are the possible causes of operation and surgeon. Thirdly, length of stay has all other attributes as the possible causes. A grammar is written as a template for these three kinds of rules. We have used a population size of 300 to run for 50 generations in the rule learning step. The results are listed in Table 3.

About	No. of Rules	*cf*			*support*			*cf / prof*
		mean	max	min	mean	max	min	mean
Diag-nosis	2	45.6%	51.4%	39.8%	9.2%	10.0%	8.4%	1.6
Opera-tion	8	42.6%	74.0%	28.0%	5.4%	16.2%	3.2%	2.0
Stay	7	71.1%	81.1%	47.0%	4.5%	8.7%	3.1%	2.5

Table 3: Summary of the rules for the fracture database

The learning process can uncover knowledge about the age effect on fracture, the relationship between diagnoses and operations, and the effect of diagnoses and operations on lengths of staying in the hospital.

The results have been evaluated by the medical experts. Previous analysis on fracture patterns only gave an overall injury pattern. Our system automatically uncovered relationships between different attribute values. The learned rules provide interesting patterns that were not known before. The rules revealed some interesting treatment patterns and rules. It can provide a good monitor of change of pattern if the data mining process is continued longitudinally over the years. It also provides the information for setting up a knowledge-based instruction system to help young doctors in training.

6.2. Scoliosis Database

6.2.1. Results of Bayesian Network Learning

The learning of network structures and discretization policies are alternated for 20 iterations. For the learning of network structures using MDLEP, we have used a population of 50 to run for 100 generations. In each iteration of the learning of discretization policies using GA, the population size is 50 and the number of generations is 10. The discovered Bayesian network structure learned from this data set is shown in Figure 3. The age is divided into 0-12 (child), 13-16 (adolescence), 17-21 and over 22. The degrees and Mens are divided into different ranges.

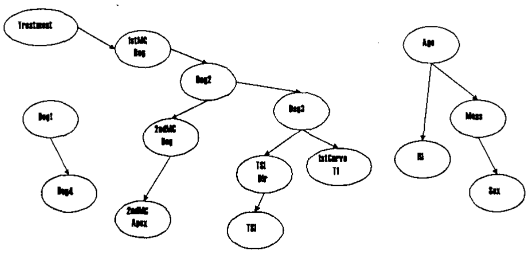

Figure 3: The discovered network structure for the Scoliosis database

The discovered Bayesian network shows some physical relationships among attributes. For example, the network shows that 1stMCDeg and 2ndMCDeg are related with Deg2. The two major curves are defined as the curves with the largest degrees among the four curves, and most likely Deg2 are involved. Deg1 and Deg2 can imply Deg4 and Deg3 because if the degree of first or second curves are small, the degree of the remaining curves are either zero or small. The network also reveals some patterns; Age affects Mens and RI (the maturity), and the value of Mens affects Sex. In addition, the following relationships are observed:

- The value of Operation affects the value of 1stMCDeg. If Operation equals to observation, the value 1stMCDeg is smaller. If Operation equals to surgery, the value of 1stMCDeg is large.
- The value of Deg3 affects the value of 1stCurveT1. If Deg3 is large, the spine has three or more curve, and most likely the first curve starts at the first vertebra T1.
- The value of Deg3 affects the value of TSIDir. If Deg3 is small, most of the time the direction of trunk shift is null.
- The value of Treatment affects the value of 1stMCDeg. If treatment is bracing, most likely the degree of the first major curve is small. In contrast, if operation is needed, the degree of the first major curve is usually large.

6.2.2. Results of Rule Learning

The medical experts are interested in inducing knowledge about classification of Scoliosis. Scoliosis can be classified as Kings, Thoracolumbar(TL) and Lumbar(L), while Kings can be further subdivided into K-I, II, III, IV and V. This domain knowledge has been incorporated in the design of the rule grammar.

The population size used in the rule learning step is 100 and the maximum number of generations is 50. For each class of Scoliosis, a number of rules are obtained. The results are summarized in Table 4.

The rules discovered are generally consistent with the knowledge of medical experts. However there is an unexpected rule for the classification of King-II. Under the conditions specified in the antecedents, our system found a rule with a confidence factor of 52% that the classification is King-II.

However, the domain expert suggests the class should be King-V! After an analysis on the database, we revealed that serious data errors existed in the current database and that some records contained an incorrect Scoliosis classification. The rules for TL and L also show something different in comparison with the rules suggested by the clinicians. According to our rules, the classification always depends on the location of the *first major curve*, while according to the domain expert, the classification always depends on the *larger major curve*. After discussion with the domain expert, it is agreed that the existing rules are not defined clearly enough, and our rules are more accurate than theirs. Thus, our rules provide hints to the clinicians to re-formulate their concepts.

	No. of	*cf*			*support*			*prof*
Class	Rules	mean	max	min	mean	max	min	mean
King-I	5	94.84%	100%	90.48%	5.67%	10.73%	0.86%	28.33%
King-II	5	80.93%	100%	52.17%	6.61%	14.38%	1.07%	35.41%
King-III	4	23.58%	25.87%	16.90%	1.56%	2.58%	0.86%	7.94%
King-IV	3	24.38%	29.41%	19.35%	1.18%	1.29%	1.07%	2.79%
King-V	5	54.13%	62.50%	45.45%	0.97%	1.07%	0.86%	6.44%
TL	1	41.18%	41.18%	41.18%	1.50%	1.50%	1.50%	2.15%
L	3	54.04%	62.50%	45.45%	2.00%	2.79%	1.07%	4.51%

Table 4: Results of the rules for Scoliosis classification

The biggest impact on the clinicians from the data mining analysis of the Scoliosis database is the fact that many rules set out in the clinical practice are not clearly defined. The usual clinical interpretation depends on the subjective experience. Our data mining effort revealed quite a number of mismatches in the classification on the type of Kings curves. After a careful review by the senior surgeon, it appears that the database entries by junior surgeons may not be accurate and that the data mining rules discovered are in fact more accurate! The classification rules must therefore be quantified. The rules discovered can therefore help in the training of younger doctors and act as an intelligent means to validate and evaluate the accuracy of the clinical database.

7. Conclusion

We have presented our approaches for knowledge discovery from two medical databases. Firstly, rules are learned to represent the interesting patterns of the data. Secondly, Bayesian networks are induced to act as causality relationship models among the attributes. The Bayesian network learning process is divided into two phases. In the first phase, a discretization policy is learned to discretize the continuous variables, and then Bayesian network structures are induced in the second phase. We employ advanced evolutionary algorithms such as Generic Genetic Programming,, Evolutionary Programming, and Genetic Algorithm to conduct the learning tasks.

From the fracture database, we have discovered knowledge about the patterns of children fracture. From the Scoliosis database, we have discovered knowledge about the classification of Scoliosis. We also have found unexpected rules that led to discovery of errors in the database. These results demonstrate that the knowledge discovery process can find interesting knowledge about the data, which can provide novel clinical knowledge as well as suggest refinements of the existing knowledge.

Acknowledgments

The work was partially supported by Hong Kong RGC CERG Grant CUHK 4161/97E. The authors wish to thank Mr. Po Shun Ngan, Ms. Chun Sau Lau and Ms. King Sau Lee for preparing, analyzing and implementing the rule learning system and the Bayesian network learning system.

References

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 International Conference on Management of Data (SIGMOD 93)*, pages 207-216, 1993.

[2] U. M. Fayyad, G. Piatesky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. *AI Magazine*, pages 37-54, Fall 1996.

[3] L. Fogel, A. Owens, and M. Walsh. *Artificial Intelligence through Simulated Evolution*. New York: John Wiley and Sons, 1966.

[4] N. Friedman and M. Goldszmidt. Discretizing continuous attributes while learning Bayesian networks. In *International Conference on Machine Learning*, pages 157-165, 1996.

[5] D. Heckerman and M. P. Wellman. Bayesian networks. *Communications of the ACM*, 38(3):27-30, March 1995.

[6] J. H. Holland. *Adaptation in Natural and Artificial Systems*. Bradford/MIT Press, 1992.

[7] J. R. Koza. *Genetic Programming: on the programming of computers by means of natural selection*. Bradford/MIT Press, 1992.

[8] W. Lam and F. Bacchus. Learning Bayesian belief networks – an approach based on the MDL principle. *Computational Intelligence*, 10(3):269-293, 1994.

[9] K. S. Leung, Y. Leung, L. So, and K. F. Yam. Rule learning in expert systems using genetic algorithm: 1, concepts. In *Proceedings of the 2nd International Conference on Fuzzy Logic and Neural Networks(Iizuka, Japan)*, pages 201-204, 1992.

[10] R. S. Michalski, A theory and methodology of inductive learning. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning – An Artificial Intelligence Approach*, chapter 4. Los Altos, Calif., 1983.

[11] J. Rissanen. Modeling by shortest data description. *Automatica*, pages 465-471, 1978.

[12] M. L. Wong, W. Lam, and K. S. Leung. Using Evolutionary Programming and Minimum Description Length Principle for Data Mining of Bayesian Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(2):174-178, 1999.

[13] M. L. Wong and K. S. Leung. Evolutionary program induction directed by logic grammars. *Evolutionary Computation*, 5:143-180, 1997.