A Bayesian Network Approach for Causal Action Rule Mining

Pirooz Shamsinejad and Mohamad Saraee

***Abstract*— Actionable Knowledge Discovery has attracted much interest lately. It is almost a new paradigm shift toward mining more usable and more applicable knowledge in each specific domain. Action Rule is a new tool in this research area that suggests some actions to user to gain a profit in his/her domain.**

Up to now some methods have been devised for action rule mining. Decision Trees, Classification Rules and Association Rules are three learner machines that already have been used for action rule mining. But when we want to suggest an action we need to know the causal relationships among parameters and current methods can't say anything about that. So that we use here Bayesian Networks as one of the most powerful knowledge representing models that can show the causal relationships between variables of interest for extracting action rules. Another benefit of new method is about the background knowledge. Bayesian Networks are very powerful at integrating the background knowledge into model.

At the end of this paper an action rule mining system is proposed that can suggest the most profitable action rules for each case or class of cases.

***Index Terms*— Actionable Knowledge Discovery, Action Rule Mining, Bayesian Networks, Causal Action Rule.**

I. Introduction

Data Mining is "the process of discovering patterns in data. The process must be automatic or semiautomatic. The pattern discovered must be meaningful in that they lead to some advantage, usually an economic advantage. " [1]. Up to now most of the researches in this area have focused on finding different types of patterns from different data but a few of them have paid enough attention to usability of mined patterns. Subsequently, there is a noticeable gap between delivered patterns and business expectations that is the final goal of data mining.

Actionable Knowledge Discovery is almost a new paradigm shift toward mining more usable and more applicable knowledge in their corresponding domains. The AKD concept can be illustrated well by an example in CRM, involving a bank loan system. We can define two types of useful knowledge in this system. First, "How much is the probability of a customer pay back his loan?" and second, "How we can increase the probability of a customer pay back his loan?" The first question is more informative and less actionable and it is the concern of traditional data mining, but the second one is more actionable and AKD aiming for answering it.

Up to now, two main categories of approaches in AKD have been reported in the literature. First, those who define actionability as an interesting measure for filtering patterns which have been mined using traditional data mining methods [2]-[5]. The second one is those who try to extract new type of patterns namely "action rules" from data sets [6]-[11]. The work presented is in this paper placed in second category.

Action rule is a rule that suggests an action to user to gain a profit in his/her domain. Each action rule usually contains some attribute value changes that could be applied to specific customer or class of customers. For example in our bank loaning system an action rule could be like this: "If we can change marital status of male customers from single to married in some way then the probability of they pay back their loan will be more ".

Up to now some learner machines have been utilized for action rule discovery. In [6], [7] Yang et al have used decision tree in two steps: first a decision tree is made from data set and then set of attribute value changes that give the maximum net profit are found as action rules. Ras et al have used Classification Rules in [8]-[11]at following two phases. First, classification rules are mined from data using traditional methods and next, action rules are generated by combining classification rules with different class attribute values. They also proposed a method for finding action rules without finding classification rules in advance[12].

One of the most important drawbacks of current methods comes from a well-known principle in statistics that states "association doesn't imply causation". So that inferring action rules from associative models and not causal models may result in non applicable action rules while the main goal of action rules – as a part of actionable knowledge discovery- is to be applicable. Our contribution is using Bayesian Networks (BN) for finding action rules to remedy the problem of insane action rules. BN is one of the most powerful modeling tools that can represent the causality relationships between attributes of a domain. Causality is the pivot of actions in real world; For example we drink water because it quenches our thirst. Therefore if we can find actions directly using causalities, they would be more accurate and more applicable in their corresponding domains. Our aim is to find action rules using causality relationships that are conveyed by Bayesian Networks.

The rest of paper is as follows: In Section II Bayesian Networks and its features as a representing model is discussed. Action Rule is defined in Section III. In Section IV we go through the process of action rule discovery using Bayesian Networks. The conclusion of paper will come in

Manuscript received September 20, 2011.

Pirooz Shamsinejad is a Ph.D. candidate at Electrical and Computer Engineering Department of Isfahan University of Technology, Isfahan, Iran (e-mail: p_shamsinejad@ec.iut.ac.ir).

Mohammad Saraee is the founder and Director of the Intelligent Databases, Data Mining and Bioinformatics Research Centre.

Section V and finally some future works are presented in Section VI.

II. Bayesian Network

A. Definitions

A Bayesian Network (BN) is a graphical model that represents probabilistic relationships among variables of interest. It was first introduced by Judea Peal[13] and afterwards it used in many domains including artificial intelligence, statistic, and philosophy. Although, there are a number of methods for representing the mined knowledge from data including decision trees, rule based systems, Artificial Neural Networks, BN and Support Vector Machines, but BN has the following properties that make it unique for knowledge representing [14].

1. It is well suited to deal with missing values in data that is a major problem in DM applications.
2. BN makes it possible to learn about causal relationships which are very important for understanding the problem domain.
3. By combining the BN with Bayesian statistical techniques we can integrate the prior domain knowledge with data and find more applicable patterns in real world through this.
4. Forth, BN in conjunction with Bayesian methods can be immunized against over fitting problem.

Here we use Bayesian Network knowledge representing model to discover action rules from data. The process will be discussed in detail in subsequent sections. Nonetheless, before we explain the different parts of using a BN for modeling and inferring, it can be useful if we shed some light on the differences between a BN and Causal Networks. Technically, BNs do not show the causal relationships. They are just a graphical and compact representation of a joint probability distribution. In other words a BN just states that if we know the values of parents of a node in BN, we can then compute the probabilities of different values of that node without any need to know the value of other nodes in BN. But Causal Networks are more restrictive than BN in this way that the edges in CNs must show the direct causation. So that a CN states that parents of a node are direct causes of it in the corresponding domain. It is clear that CNs convey more knowledge than BNs but creating them is also more complicated and sometimes impossible. For more arguments about this topic refer to [15]. Despite the mentioned subtle difference between BNs and CNs we consider BNs in this paper as a causal model.

There are three major issues that we must deal with when we want to use BNs as a modeling tool: First, how to learn the structure of the BN from data; Second, how to estimate the parameters of the BN from data; Third, how to perform fast enough necessary inferences from the learnt BN. In the next subsections we argue these issues concisely.

B. Structure Learning

As it mentioned before BN is a graphical model. More accurately BN is a Directed Acyclic Graph that its nodes are variables of interest in the domain. Its edges show a property about nodes in BN that it is known as Markov Condition in literatures because of its similarity to Markov chains' property. We bring the definition of Markov Condition according to [16]:

> *Assume we have a joint probability distribution P of set of random variables V and a DAG $G=(V,E)$. We say that (G,P) satisfies the Markov Condition if for each variable $X \in V$, $\{X\}$ is conditionally independent of the set of all its nondescendents given the set of all its parents.*

It is the condition that allows us to estimate the value of a variable in a BN only by knowing the values of its parents.

By structure learning, we mean finding a DAG from data that meets the Markov Condition. It is worth noting that the type of data in AKD is historical that is the usual type of data in DM applications and therefore are non-experimental (aren't collected from designated experiments) in spite of experimental data which are used in some statistical domains for evaluating some specific hypothesizes about data.

Up to now many works have been done for learning structure of BN from data. The structure learning methods can be divided to these three types:

1) Constrain-Based learning methods: These methods use conditional independencies among variables and Markov property to find the structure of network. IC algorithm [15], PC algorithm and SGS [17] are some of the most prominent algorithms of this type. But the main drawback of this type of methods is about their time complexity. Almost all of them need exponential time in number of variables and this makes them prohibitive for DM applications with medium or large number of attributes.
2) Score-Based learning methods: These methods assign a score to each candidate BN structure and try to find a structure with maximum or local maximum score using some heuristic searches. Greedy Equivalence Search [18] is one of these methods. These heuristic searches can do fast but sometimes fall in local optimum points.
3) Hybrid methods: This type of methods uses both constraint-based and score-based techniques to learn the structure. Max-Min Hill Climbing method [19] is one of the best methods of this type and also of all structure learning methods. It is shown in [19] that this method is applicable on large databases with hundreds of variables.

C. Parameter Learning

After finding the structure of BN it then must be parameterized. That is estimating the values of conditional probability table cells for each variable in the BN. In simple words the goal is assigning a value to each parameter in a way that the network can reconstruct the data with minimum error.

There are some methods reported including Maximum Likelihood method that can be used for doing such estimating. Very simple and general version of ML method assigns a random variable to each parameter and then tries to change each parameter a bit to the direction of a point with minimum (or local minimum) error in the space of all parameters. The complete details and variations of ML method can be found in [20].

D. Inference

After creating the BN we can use it for answering some queries about the learnt knowledge. There are four types of general queries that can be posed respect to a BN that we explain them here briefly, one can refer to [6] for complete definitions and examples:

1) Probability of Evidence: This is the simplest form of query and it means asking about the probability of some variable instantiation. In simple words if **e** be the evidence we are looking for Pr(**e**).
2) Prior and Posterior Marginal: This is the most common query form. Given a joint probability distribution $\Pr(x_1,...,x_n)$, (prior) marginal distribution $\Pr(x_1,...,x_m)$ where $m \le n$ is defined as follows :
$$\Pr(x_1,...,x_m) = \sum_{x_{m+1},...,x_n} \Pr(x_1,...,x_n)$$
In other words marginal distribution is a projection of joint probability distribution on a smaller subset of its variables $x_1,...,x_m$. Posterior marginal distribution shows marginal probability given some evidence **e** and it is defined as follows:
$$\Pr(x_1,...,x_m|\mathbf{e}) = \sum_{x_{m+1},...,x_n} \Pr(x_1,...,x_n|\mathbf{e})$$
This type of query is used when we are interested to know the probability distribution of some network variables given the values of other variables.
3) Most Probable Explanation (MPE): The aim of this type of query is to find the most probable instantiation of network variables given some evidence **e**. In simple words it identifies the most probable state of the network when we know the values of some parts of it.
4) Maximum a Posteriori Hypothesis: This is a special case of MPE in this way that it tries to find the most probable instantiation of a subset of network variables in spite of all variables. It is more common that MPE and easier to compute.

In this paper, the BN modeling tool is explained to the degree needed for our work. One can refer to [13]-[17], [20] for detailed discussions about BN. In the next section we will use the BN concepts for finding causal action rules.

III. Action Rules

Action rule is a rule that suggest an action to user to gain a profit in his/her domain. It is despite traditional rules that only give information about underlying knowledge in data. Each action rule usually contains some attribute value changes that could be applied to specific customer or class of customers. Attributes typically divide into two main types: input attributes and goal attribute. Input attributes also divide into two types: flexible and stable attributes. Value of flexible attributes can change at a reasonable cost in spite of stable attributes. By action we mean a change in an attribute value of an instance. Therefore each action rule is a set of some actions.

Let's consider a simple example for illustrating the concepts. We will use this example through the paper for other discussions. Let's assume a company has a database of its customers' data that contains five attributes as follows:

- Service Type (T), the type of service that customer uses with values: T1, T2, T3
- Using rate (R), the rate of using the service by customer with values: Low (L), High (H)
- Sex (S) : Male (M), Female (F)
- Advertisement Sent (A), has advertisement been sent to customer : Yes (Y), No (N)
- Customer Loyalty (C), is customer loyal or attritor: Low (L), High (H)

In this example "Customer Situation" is goal attribute because it has a direct effect on profit at corresponding domain. Low loyalty and high loyalty result low profit and high profit respectively. "Service Type", "Using Rate" and "Advertisement Sent" are flexible attributes. "Sex" is a stable attribute.

If there is a customer record like $I_1(t_1, l, m, n, l)$ where the order of attributes is as above list. Then a sample action rule for this instance could be like:

$$\mathrm{T}: \mathrm{t}_1 \to \mathrm{t}_2, \mathrm{A}: \mathrm{n} \to \mathrm{y} \Rightarrow \mathrm{C}: \mathrm{l} \to \mathrm{h}$$

which states if the attribute T changes from t_1 to t_2 and attribute A changes from "n" to "y" then it implies attribute C - the goal attribute – probably will change from "l" to "h". In other words this action rule suggest to domain expert to change the value of attributes T and A for customer I_1 according to the rule and then he/she can expect a probable change in the value of goal attribute. The method for finding and evaluating action rules for an instance will be described next. As brevity the name of attributes have been omitted from the examples but it will be added everywhere it may cause confusion.

IV. Action Rule Discovery using Bayesian Networks

Up to now many methods have been proposed for mining actions instead of patterns from data. The goal of these actions is to gain a profit in their corresponding domain. Here an action discovery method based on Bayesian networks is proposed. The method works in two phases. In the first phase namely modeling phase it takes data about the instances as input and then creates a Bayesian Network for modeling causality relationship between attributes of instances in database. In the second phase namely discovering phase method takes each time an instance and generates the highest profitable actions for that case. Before we explain the details, presumptions must be clarified. In the next subsection we present our assumptions about the problem.

A. Assumptions

Because of wide range of assumptions in actionable knowledge discovery literatures it is necessary to settling on the assumptions first.

- Input data are set of instances which each instance is composed of fix number of <attribute,value> pairs. We show this type of data in tabular format.
- One of the attributes is known as goal attribute. The value of goal attribute for an instance directly affects the profit that could be gained by the instance. For simplicity we assume goal attribute has two values "Low" and "High" that represent their profit. Then our final goal is to change value of goal attribute of an instance from L to H.

- There is no information about actions as input data; neither about actions nor about their effects on some instances. So that the method that presented here is different from methods which simply search through some actions to find best of them.

It will be described how the method works in each phase in the following subsections.

B. Modeling Phase

The heart of this phase is to learn a Bayesian network from input data. Learning a Bayesian network includes three main tasks [20]:

1. Specifying the set of relevant attributes and their values. It means defining the domain of problem.
2. Constructing the structure of BN by connecting each pair of attributes which there is a cause and effect relationship between them. The resulting structure is a Directed Acyclic Graph (DAG) that its nodes are attributes and each causal relationship shows by an edge starting from cause node pointing to effect node.
3. Parameter learning which means finding the values of conditional probabilities for each attribute. It is the quantitative part of the learning process.

In section II we described the final two steps in more detail. We can use any BN structure learning algorithm or causal learning algorithms if we want to be conservative about causal relationships. It is clear that as much as the discovered relationships among variables of interest are more causal, the resulted action rules will be more applicable in corresponding domain.

C. Discovering Phase

In this phase constructed BN is used to find the highest profitable action rules for a specific instance. By highest profitable action rule we mean an action rule that with the highest probability will change the goal attribute of a specific instance from L to H. It is noteworthy that one of our important assumptions is that there is no prior knowledge about actions or their influence on attributes.

We must first create action rules and then compute their power to changing the value of goal attribute. Here, the modeled BN used to estimate the power of an action rule. Discovering phase for a specific instance can be done in three following steps:

- Step1: Finding the candidate action rules for the instance
- Step2: Estimate the power of each action rule to changing the goal attribute
- Step3: Ranking the action rules based on their power and selecting the most profitable ones.

We will describe these steps through an example for better understanding.

Let's assume the BN that is depicted in fig.1 has been constructed through the modeling phase for our company example in section III. The BN shows the existing causal relationships between attributes. It states that the value of attributes T and S have direct effect on value of attribute R and the value of attributes R and A have direct effect on the value of attribute C.

For simplicity, we assume that there is no hidden cause in the Bayesian network and the network is exhaustive. So that the only causes of attribute R are S and T and there is no hidden attribute that can influence on the value of attribute R. This assumption implies that it isn't possible to change attribute R directly and we must change it only through changing the value of its parents T and S. At the other hand S is a stable attribute and can't be changed so that R only can be changed using attribute T.

It can be concluded from the above discussion that only those attributes can be directly changed which are flexible and have no parents in the BN; we call them mutable attributes. In our example T and A are the mutable attributes.

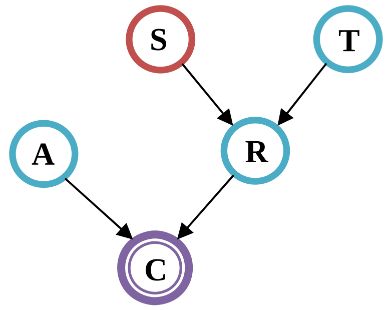

Fig. 1. The BN structure for company example. C is goal attribute, A, R and T are flexible attributes and S is a stable attribute.

To locate, the most profitable action rules for an instance, we must first determine candidate action rules. As mentioned above only the mutable attributes can change directly, so that the candidate action rules must be constructed from the mutable attributes. For implement this task, we must consider all combinations of changing the value of each mutable attributes to its other values for each specific instance.

In our example, the following candidate action rules are all possible action rules for instance I_1.

$$a_1 : (T : t_1 \to t_2 \Rightarrow C : l \to h)$$
$$a_2 : (T : t_1 \to t_3 \Rightarrow C : l \to h)$$
$$a_3 : (A : n \to y \Rightarrow C : l \to h)$$
$$a_4 : (T : t_1 \to t_2, A : n \to y \Rightarrow C : l \to h)$$
$$a_5 : (T : t_1 \to t_3, A : n \to y \Rightarrow C : l \to h)$$

After finding candidate action rules it is necessary to define a strategy to rank them and select the highest profitable of them. Here we define profitability of an action rule regarding a specific instance. In other words one action rule may have different effects on different instances. We define the profitability of an action rule as follows:

"If a be a candidate action rule for instance I and I' be the resulting instance of applying action rule a on I then the *profitability* of a defines as the probability of goal attribute of new instance I' has value of high".

If a be an action rule regarding an instance I we can show it like this:

$$a_1 : v_{11} \to v_{12}, a_2 : v_{21} \to v_{22}, \dots, a_n : v_{1n} \to v_{2n} \Rightarrow g : l \to h$$

Then the Profitability of a regarding instance I is the value of following probability.

$$\text{profitability}_{a,I} = \Pr(H \,|v_{12}, v_{22}, \dots, v_{n2}, v_{c_1}, v_{c_2}, \dots, v_{c_m})$$

where H in the above formula is the value of the goal attribute. v_C shows value of stable attributes or attributes of instance I that no change suggests for them in action rule a.

Now, we use the BN constructed in modeling phase to compute the profitability of an action rule. For doing this we can making query from learnt BN. It is clear from explanations given in section II that profitability is a query of form posterior marginal where we want to find the probability of goal attribute take the value "High" given the value of other attributes. So that we can select from different algorithms which exist for performing this type of query [20] for computing the profitability of an action rule. After computing the profitability of all candidate action rules we can sort them and select the most profitable ones. For illustrating the method we compute the profitability of candidate action rules for our example. In these examples for clarity and simplicity we just compute the profitability using the basic concepts of BNs and not the sophisticated algorithms. Necessary conditional probability tables for the BN of our example are shown in table I and table II.

Table I shows values of $\Pr(R|T,B)$ for different values of R,T and B. Likewise Table II shows values of $\Pr(C|R,A)$ for different values of C, R, and A. For brevity only probabilities for "h" value of goal attribute are shown in tables. The probability of "l" value can be computed easily using the corresponding row for "h" value.

It is noteworthy that based on Markov conditions for BN, the value of each attribute only depends to the value of its parents, so that it isn't necessary that the values of other attributes come in conditional probability table.

The profitability of action a_1 regarding instance $I_1(T_1, L, M, Y, L)$ defined as follows:

$$\text{Profitability}_{a_1,I_1} = pr(h|t_2, m, y)$$

TABLE I: CONDITIONAL PROBABILITY TABLE FOR ATTRIBUTE R

R	T	B	pr(R | T , B)
h	t_1	m	0.7
h	t_1	f	0.2
h	t_2	m	0.3
h	t_2	f	0.9
h	t_3	m	0.5
h	t_3	f	0.4

TABLE II: CONDITIONAL PROBABILITY TABLE FOR GOAL ATTRIBUTE C

C	R	A	pr(C | R , A)
h	l	n	0.2
h	l	y	0.5
h	h	n	0.3
h	h	y	0.7

To compute the above probability we use BN structure, values of conditional probabilities and conditionally independency property of non-descendant nodes like the following:

$$pr(C:h|T:t_2, S:m, A:y) =$$
$$pr(C:h|A:n, R:l) \times pr(R:l|T:t_2, S:m) +$$
$$pr(C:h|A:n, R:h) \times pr(R:h|T:t_2, S:m)$$
$$= 0.2 \times 0.7 + 0.3 \times 0.3 = 0.23$$

The profitability of other actions can be computed like the following calculations:

$$\text{Profitability}_{a_2,I_1} = pr(c:h|T:t_3) = 0.25$$
$$\text{Profitability}_{a_3,I_1} = pr(c:h|A:y) = 0.5$$
$$\text{Profitability}_{a_4,I_1} = pr(c:h|A:y, T:t_2) =$$
$$= pr(c:h|A:y, R:l) \times pr(R:l|S:m, T:t_2) +$$
$$pr(c:h|A:y, R:h) \times pr(R:h|S:m, T:t_2)$$
$$= 0.5 \times 0.7 + 0.7 \times 0.3 = 0.56$$
$$\text{Profitability}_{a_5,I_1} = pr(c:h|A:y, T:t_3) =$$
$$= pr(c:h|A:y, R:l) \times pr(R:l|S:m, T:t_3) +$$
$$pr(c:h|A:y, R:h) \times pr(R:h|S:m, T:t_3)$$
$$= 0.5 \times 0.5 + 0.7 \times 0.5 = 0.6$$

According to the above calculations and the definition of profitability it is shown that a_5 is the most profitable action rule for instance I_1. a_4, a_3, a_2 and a_1 are the next profitable ones respectively. In the next subsection we bring some complexity issues and some solutions for them.

D. Complexity Issues

Although this new method can find more sensible and applicable action rules but it is necessary to consider the complexity bottlenecks of this method and fast enough algorithms in each step must be devised. The first complexity issue emerges in modeling phase where we must to create the BN structure from databases. As it was described in section II there are some methods for BN structure learning that can approximate the structure of BN in reasonable time. Although it is shown in [21] that learning the structure of some sort of networks is an NP-Hard problem.

The next complexity issue is about the number of candidate action rules which can be too large when the number of attributes increases. Therefore computing the profitability for them would be a bottleneck for our method. We can remedy this problem by computing the profitability of some candidate action rules at the same time. It is possible by making other types of query from the BN like Maximum a Posteriori Hypothesis.

E. Action Rule Discovery System

Based on the new method presented here an abstract system has been devised for mining action rules using Bayesian networks. The system on first step takes data and background knowledge as input and then builds the BN from data. It is worth noting that one of the advantages of the BN over other learning methods is its ability to integrate background knowledge in learning process. On second step, system frequently takes instances as input, creates the candidate actions, Find and send out the most profitable action rules. The Action Rule Discovery System is depicted in fig. 2.

V. CONCLUSION

Action Rules are new tools for actionable knowledge discovery. Up to now, many methods have been developed for mining action rules. Decision trees, classification rules and association rules are three types of learner machines that have been used for this task till now.

Our contribution is to consider causal relationships between variables for mining action rules. We use BN for as one of the most powerful knowledge representing models that can show causal relationship and also can be integrated with background knowledge very well. It matches well with human reasoning system and this point makes the resulting action rules more applicable in problem domain. It can handle missing data as well.

The Profitability measure for action rules based on probability is presented in this paper and a method for computing profitability of action rules is developed. Some complexity issues of the method along with solutions to them have been presented. Finally an Action Rule Discovery System has been suggested that takes data, background knowledge and some instances as input and send out the most profitable action rules for each instance.

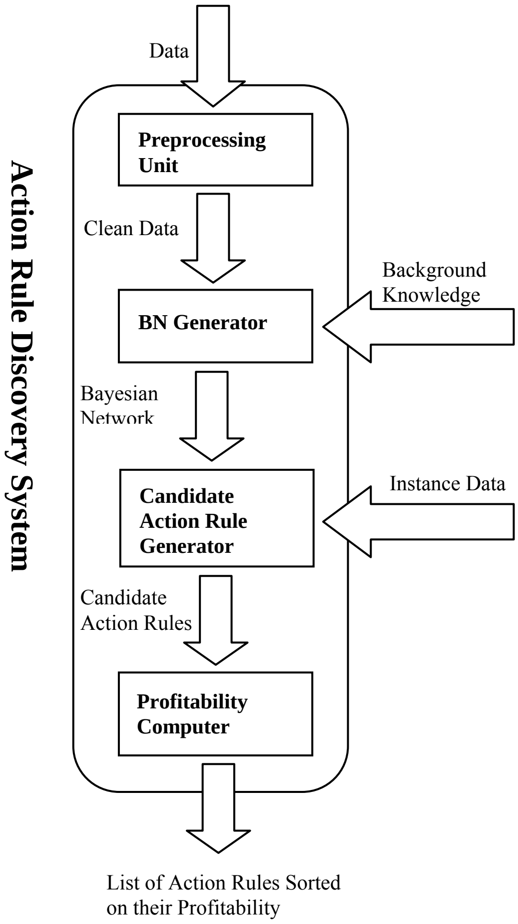

Figure 2: Action Rule Discovery System based on Bayesian Networks

VI. Future Works

Using BN for action rule mining is a new territory in Data Mining and therefore it is an open area for ground breaking research works. The future works can be listed as follow:

- Implementing the Action Rule Discovery System and evaluating it in real domains.
- Devising fast and robust algorithms for finding causal relationships from large databases.
- Defining profitability of an action rule for a class of instances and not only a specific instance.
- Devising high performance and robust algorithms for finding action rules from Bayesian Networks.
- Using Action Rule Discovery System on some real domains and compare the results with other methods.
- Extending the Action Rule Discovery System to accept also some background knowledge about action rules.

We are working on some of them now in our data mining laboratory.

References

[1]. I. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann Pub, 2005.

[2]. K. McGarry, "A survey of interestingness measures for knowledge discovery". *The Knowledge Engineering Review*, vol. 20, no. 1, pp. 39-61, 2005.

[3]. B. Liu, W. Hsu, and Y. Ma, "Identifying non-actionable association rules", in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, 2001, pp. 329-334.

[4]. H. Kaur, "Actionable rules: issues and new directions", *Transactions on Engineering, Computing and Technology*, pp. 61–64, 2005.

[5]. Z. Lei, L. Bing, j. Benkler and j. Chi. "Finding Actionable Knowledge via Automated Comparison". in *IEEE 25th International Conference on Data Engineering,* 2009, pp. 1419-1430.

[6]. Q. Yang, J. Yin, C. Ling and R. Pan, "Postprocessing decision trees to extract actionable knowledge", *IEEE Transactions on Knowledge and Data Engineering*, pp. 43-56, 2007.

[7]. Q. Yang, J. Yin, C. Ling and R. Pan. "Extracting actionable knowledge from decision trees". *IEEE Transactions on Knowledge and Data Engineering*, pp. 43-56, 2007.

[8]. Z. Ras and A. Wieczorkowska, "Action-rules: How to increase profit of a company". *Lecture Notes in Computer Science*, pp. 587-592, 2000.

[9]. Z. Ras and L. Tsay, "Discovering extended action-rules (System DEAR)". in *Proceedings of international Intelligent Information Systems,* 2003, vol. 20, p. 293–300.

[10]. L. Tsay and Z. Ras, "Action rules discovery: System DEAR2, method and experiments". *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 17, no. 1-2, pp. 119-128, 2005.

[11]. L. Tsay and Z. Ras, "Action Rules Discovery System DEAR_3". *Journal of Foundations of Intelligent Systems,* 2006, pp. 483-492.

[12]. Z. Ras, E. Wyrzykowska and H. Wasyluk. "ARAS: Action rules discovery based on agglomerative strategy". in *Proceedings of the 3rd ECML/PKDD international conference on Mining complex data,* pp.196-208, 2007.

[13]. J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Morgan Kaufmann, 1988.

[14]. D. Heckerman, "A tutorial on learning with Bayesian networks", *Journal of Innovations in Bayesian Networks*, pp. 33-82, 2008.

[15]. J. Pearl, *Causality : models, reasoning, and inference*, Cambridge University Press, 2000.

[16]. R. Neapolitan, *Learning bayesian networks*, Pearson Prentice Hall, 2004.

[17]. P. Spirtes, C. Glymour, and R. Scheines, *Causation, prediction, and search*, The MIT Press, 2000.

[18]. D.M. Chickering, "Learning equivalence classes of Bayesian-network structures", *The Journal of Machine Learning Research*, vol. 2, pp. 445-498, 2002.

[19]. I. Tsamardinos, L.E. Brown and C.F. Aliferis, "The max-min hill-climbing Bayesian network structure learning algorithm", *Journal of Machine Learning*, vol. 65, no. 1, pp. 31-78, 2006.

[20]. A. Darwiche, *Modeling and reasoning with Bayesian networks*, Cambridge University Press, 2009.

[21]. D.M. Chickering, D. Heckerman and C. Meek, "Large-sample learning of Bayesian networks is NP-hard". *The Journal of Machine Learning Research*, vol. 5, pp. 1287-1330, 2004

Pirooz Shamsinejadbabaki has born in Tehran, Iran, 1984. He has a B.Sc. in Computer Software Engineering from Isfahan University of Technology, 2004. His M.Sc. is in Computer Architecture from IUT, 2006. He is a Ph.D. candidate in Computer Science at IUT from 2007.
His research interest is actionable knowledge discovery from large databases, domain driven data mining, text mining and genetic algorithms.

Dr Mohamad Saraee received his PhD from University of Manchester in Computation, MSc from University of Wyoming, USA in Software Engineering and BSc from ShahidBeheshti University, Iran.
His main areas of research are Intelligent databases, Mining advanced and complex data including medical and Bio, Text Mining and E-Commerce. He has published extensively in each of these areas and served on scientific and organizing committee on number of journals and conferences.