Building Bayesian Network based Expert Systems from Rules

Saravanan Thirumuruganathan and Manfred Huber
Dept. of Computer Science and Engineering
University of Texas at Arlington
saravanan.thirumuruganathan@mavs.uta.edu, huber@cse.uta.edu

***Abstract*—Combining expert knowledge and user explanation with automated reasoning in domains with uncertain information poses significant challenges in terms of representation and reasoning mechanisms. In particular, reasoning structures understandable and usable by humans are often different from the ones used for automated reasoning and data mining systems. Rules with certainty factors represent one possible way to express domain knowledge and build expert system that can deal with uncertainty. Although convenient to humans, this approach has limitations in accurately modeling the domain. Alternatively, a Bayesian Network allows accurate modeling of a domain and automated reasoning but its inference is less intuitive to humans. In this paper, we propose a method to combine these two frameworks to build Bayesian Networks from rules and derive user understandable explanations in terms of these rules. Expert specified rules are augmented with importance parameters for antecedents and are used to derive probabilistic bounds for the Bayesian Network's conditional probability table. The partial structure constructed from the rules is fully learned from the data. The paper also discusses methods for using the rules to provide user understandable explanations, identify incorrect rules, suggest new rules and perform incremental learning.**

***Index Terms*—Bayesian Networks, Certainty factors, expert systems**

I. Introduction

An ideal framework for performing reasoning in domains with uncertain information must have convenient mechanisms to perform theoretically consistent and automated reasoning. If human users are involved, it must also be able to generate user understandable explanations for its predictions. In addition, if the expert system also involves a domain expert, it should have mechanisms to elicit and store the expert's domain knowledge.

It is well known that experts are most comfortable in specifying their domain knowledge using sets of rules because rules model experts' decision making process naturally [1]. Also, explanation of the diagnosis in terms of the expert specified rules is much more intuitive to the user than a set of probability values. By adding certainty factors to rules, we can enable them to model uncertainty in the domain. As a result, such a rule based system will have convenient mechanisms to elicit expert knowledge and explain the reasoning process in a user understandable way using the expert specified rules [1].

On the other hand, rule based systems have limited ability to completely model the domain. Since a typical rule base is not exhaustive, only a small number of the dependencies that exist in the domain are modelled. Rule based expert systems rely entirely on the rule base for prediction and hence modelling of the domain remains inexact. Moreover, certainty factors as a means of modelling uncertainty expect all rules to be specified in the exact direction in which they will be reasoned. It is not possible to use a causal rule to perform diagnosis as rule inversion is not defined in this framework. This means that a diagnostic expert system must only have diagnostic rules. Mixing of predictive and diagnostic rules results in inconsistencies. Certainty factors also make additional rule independence assumptions that undermine their formal consistency [2]. In addition, certainty factor based expert systems are static, i.e. they consist of a set of rules and certainty factors which do not change during the operation of the system. This means that the availability of additional data will not result in fine tuning of the system. If the rule base is not exhaustive, the system may not be able to make any meaningful diagnosis when provided with partial observations [2].

Bayesian Networks, in contrast, are an example of a probabilistic graphical model that has mechanisms to accurately model the domain's dependencies and perform fully automated reasoning. They have a mathematically consistent way to specify uncertainty in the system and can represent a combination of expert knowledge and data where an expert can specify dependencies among nodes or partial network structure and the complete structure and parameters are learned from data. Moreover, if the expert is not available to provide the structure, it can also be learned from the data. Given a network, it is very easy to perform automated reasoning and for the network to make predictions based on a given set of evidence. In addition it is possible to make predictions about any node in the network based on the evidence, thus allowing diagnostic inference even if the network structure is causal and vice versa. There are multiple algorithms that adapt the network based on additional data so that it can fit the data very well [3].

While Bayesian Networks provide a good mechanism for reasoning and to model knowledge, it is well known that, in practice, experts have problems specifying their domain knowledge using probability and likelihood. This means that without data it is very hard for the expert to encode his entire domain knowledge in a Bayesian Network. Also, generating user understandable explanations of the inference is hard.

In this paper, we propose a mechanism that combines the advantages of both frameworks. The expert specifies his domain knowledge using rules and certainty factors. These rules are used to bootstrap a partial Bayesian Network that is

then fully learned from the available data. Using the Bayesian Network, we can perform fully automated reasoning in a consistent way and also explain the prediction using initially specified rules as a substitute for the expert's reasoning.

To further increase the descriptive power of the rules, we augment the rules specified by the expert with importance parameters and use them to estimate the residual evidence of the related rules that can be inferred from the original rule. This allows us to make maximum use of the expert specified rules to bootstrap the network. Based on the constructed network, we also introduce mechanisms for performing inference in the Bayesian Network and use the expert specified rules to generate user understandable explanations for the network's predictions. In addition to this, the proposed approach also makes it possible to perform incremental learning as new data arrives. Furthermore, using the information obtained from the data, the system provides mechanisms to identify bad rules and suggest new rules that can better explain the predictions.

II. Related Work

There have been multiple attempts in the past to build expert systems that can perform automated uncertainty reasoning and generate user understandable explanations.

Rule based expert systems represent the knowledge of a domain using a set of rules. The rules can be causal or diagnostic. The certainty-factor (CF) model, which was originally designed for the MYCIN expert system [1], is a convenient framework to manage uncertainty in a rule-based system.

In a diagnostic rule-based system, rules are of the form *if e then h*. Here, *e* denotes evidence and *h* denotes the hypothesis. An expert expresses uncertainty by attaching a single certainty factor to each rule. In the original interpretation, Certainty factors represent the expert's change in belief in the hypothesis *h* given the evidence *e*. A positive value indicates that the expert's belief in *h given e* increases and a negative value indicates that the expert's belief decreases. It must be noted that a certainty factor does not represent a person's absolute degree of belief and hence is not the same as a probability value. The certainty factor model contains mechanisms to combine rules. More details can be found in [4] and [2].

A Bayesian Network [3] is a directed acyclic graph (DAG) whose nodes represent the variables in the problem domain and edges represent direct probabilistic dependencies among nodes. Bayesian Networks are a rigorous framework that allows accurate modelling of the domain and provides mechanisms for fully automated reasoning. As opposed to rule-based system where causal rules only support causal reasoning, it is possible to use a Bayesian Network to represent causal relationships between nodes in the domain and then use it to perform diagnostic problem-solving. In addition to being a theoretically consistent framework, Bayesian Networks can also perform fully automated reasoning. Moreover, they allow a combination of expert knowledge and data where partial structure can be obtained from an expert and the complete structure with parameters is learned from data. Well known algorithms exist to do incremental learning and fast inference even with partial observations [3]. However, it is hard to explain the inference performed by a Bayesian Network in a human understandable way and studies have shown that formulating knowledge in terms of probabilities is non intuitive even for experts [5].

To bridge the gap between certainty factors and formal uncertainty reasoning, Gordon and Shortliffe [6] show a possible interpretation of certainty factors based on the evidence framework. They consider a rule to be an evidence which supports only one hypothesis to a certain degree. If a rule confirms a hypothesis with degree *s*, this can be interpreted as the rule assigning evidence *s* to the single hypothesis corresponding to the consequent of the rule and *1-s* to the remaining hypotheses. In this interpretation, the certainty factors associated with the rules can be viewed as basic probability assignment functions. Under this interpretation, the combination of rules with certainty factors can be considered as a generalization of Dempster's rule of combination [6].

Heckerman [7] introduced a probabilistic interpretation of certainty factors that is also consistent with the framework of probability. The new formulation gives a probabilistic interpretation of certainty factors based on the likelihood ratio,

$$\lambda(h,e) = \frac{p(e|h,\xi)}{p(\neg e|h,\xi)} \tag{1}$$

Likelihood ratio is one way to describe the change in "belief" when new evidence for the hypothesis is observed. Heckerman [7] showed that any monotonic transformation of the likelihood ratio produces a valid probabilistic interpretation. For this paper, we use the following transformation [7]:

$$CF(h \rightarrow e|\xi) = \frac{\lambda(h,e) - 1}{\lambda(h,e) + 1} \tag{2}$$

Substituting (1) in (2), we get the following probabilistic interpretation that is used in the paper :

$$CF(h \rightarrow e|\xi) = \frac{p(h|e,\xi) - p(h|\xi)}{p(h|\xi)(1 - p(h|e,\xi)) + p(h|e,\xi)(1 - p(h|\xi))} \tag{3}$$

However this interpretation does not handle dependencies between rules or rules with inexact certainty factors. To handle this, the work in this paper treats the value from (3) as a bound on the conditional probability for the individual rule.

A number of expert systems [8] [9] [10] have been constructed based on Bayesian Networks due to their mathematically consistent way of handling uncertainty. [11] describes the conversion of HEPAR-RB, a rule based expert system, to HEPAR-BN which is based on Bayesian Networks. In contrast to our approach, the rules were used to construct a partial Bayesian Network which was completed using domain knowledge. The conditional probability entries were estimated from medical literature instead of from rules. A qualitative comparison of the two systems can be found in [9].

Neuro-Fuzzy [12] is another approach to build expert systems. In this approach, the structured knowledge is encoded using fuzzy logic rules and the unstructured information is encoded using neural networks. Despite the potential to have

both human understandable explanation and precise inference, in practice the system developer is force to choose one [13].

III. From rules to Bayesian Networks

In this section, we introduce a method to combine the certainty factor and Bayesian Network frameworks. To increase the descriptive power of the rules, these are augmented here with additional importance parameters for antecedents which allow to estimate evidence of related rules representing the rationale of the original rule but lacking partial antecedents. A partial Bayesian Network whose structure is guided by the rules is constructed and assigned with a consistent set of parameters based on the rules' evidence. Given a set of data, the complete network is learned and used to perform automated reasoning. We discuss ways to generate user understandable explanations of the inference using the expert specified rules. Furthermore, we introduce mechanisms to identify incorrect rules, identify new rules and perform incremental learning.

A. Augmentation of Rules

A typical rule based expert system using the certainty factor framework consists of rules *If A AND B THEN C (CF)* where *CF* represents the certainty factor associated with the rule. In this paper, an importance parameter is added to each of the antecedents in the rule, resulting in rules of the form:

$$If\ A\ (S1)\ AND\ B\ (S2)\ THEN\ C\ (CF) \tag{4}$$

There are two additional parameters, *S1* and *S2*, which provide information about the importance of the antecedents *A* and *B* respectively. Intuitively, importance of an antecedent is the degree to which it has to be present for the rule to be applicable. These values represent the absolute importance of individual antecedents and do not depend upon the values of the other antecedents in the same rule. This means that the importance values need not add up to 1 and also do not represent the probability of the corresponding antecedents.

Given a single rule with importance parameters for its antecedents, we can now estimate the residual evidence of the expert specified rule given that antecedents are not true (or have not been evaluated) and thus to provide evidence bounds on negated antecedents. As discussed in Section II, a rule with certainty factor can be considered as representing a belief function that adds a piece of evidence for its conclusions given the antecedents. With the addition of the importance parameter, the rules become more expressive. Now, they become a belief function that can potentially provide evidence given each of the subsets of the hypothesis space formed by the rule's premises.

B. Derivation of Residual Evidence

Consider the following rule:

$$If\ A\ (0.7)\ AND\ B\ (0.8)\ THEN\ C\ (0.9) \tag{5}$$

The values 0.7 and 0.8 reflect how important the antecedents *A* and *B* are to apply the rule (or more precisely, the rationale underlying the rule). A rule with importance parameters can be considered as a belief function that can provide evidence for the related rules whose antecedents are variations of the original rule's antecedents. Using the interpretation that a rule with importance parameters is a belief function and importance of an antecedent is the degree to which the antecedent has to be present in the rule, it is possible to estimate the residual evidence that a rule provides if some of its antecedents do not hold. In (5) for example, the expert has provided the importance of *A* as 0.7. If the antecedent does not hold, then the evidence provided by the rule is diluted by 0.3 $(1 - 0.7)$. In other words, the expert believes that in the absence of the antecedent *A*, the rule provides only 30% of the evidence it provided when all its antecedents hold.

An alternative is to consider the rule as a line of reasoning. A rule provides a rationale for the consequent given its antecedents. Now, if part of the antecedents are not true, then the line of reasoning might still hold but is proportionally weaker. Thus the evidence provided by a derived rule which incorporates partial antecedents must be proportionally reduced based on the degree of belief for the original antecedents. This leads to the creation of a derived rule from (5) of the form,

$$If\ \neg A\ AND\ B\ THEN\ C\ (0.27) \tag{6}$$

The certainty factor for (6) is derived by subtracting *A*'s importance from 1 and multiplying it with (5)'s certainty factor. Considering the certainty factor as evidence, we proportionally reduce the evidence provided by (5) by the degree of belief we had in antecedent *A* for (5). A similar method applies when *A* is a discrete, non binary variable.

It is important to note here that the certainty factors in these derived rules represent solely the residual evidence from rule (5) and not the evidence that the expert would assign to the new rule if he/she were to provide it manually. As such it is more clearly interpreted as a minimum of the evidence that an explicitly specified rule should provide.

C. Construction of a Bayesian Network from Rules

To construct a Bayesian Network from rules, it must be possible to map the rules to the network's structure and parameters. A rule indicates a statistical dependency between the antecedent and the consequent in either a causal or diagnostic relation. Intuitively, if the expert provides rules, a partial network with the antecedents as parent nodes and the consequents as the child nodes can be constructed. If all rules are causal, the result is a partial causal Bayesian Network. It is possible here to use a mix of causal and diagnostic rules, leading to a partially causal, partially diagnostic network. To resolve conflicts, structure learning algorithms determine conflicting parent and child nodes using statistical independence tests. The expert's domain knowledge allows him to identify some of the nodes (and connections). The Bayesian Network constructed this way then not only encodes the expert's reasoning process but also allows the system to explain the inference performed in terms of the expert's reasoning through rules.

D. From Certainty Factors to Probabilities

Once we have a partial Bayesian Network and a set of related rules with evidence, the next step is to fill conditional

probability entries for the nodes. The probabilistic interpretation given in (3) provides a way to convert a certainty factor for a rule to a conditional probability value.

$$p(h|e,\xi) = \frac{p(h|\xi)[1 + CF(h \rightarrow e|\xi)]}{1 + CF(h \rightarrow e|\xi)[2p(h|\xi) - 1]} \quad (7)$$

It must be noted that these probabilities only reflect the evidence from the original rule and might therefore underestimate the actual conditional probability. To address this, the approach presented in this paper takes a more evidence based stance [6] to derive the conditional probability estimates from the rules.

Another issue that the evidence based stance addresses is that even though it is possible to calculate conditional probabilities from each rule using Heckerman's formula, it is unlikely that the calculated probabilities are accurate. One reason is that the probabilities are highly sensitive to the certainty factors provided by experts which are subjective in nature. Also, to calculate an exact conditional probability for a term such as $P(C|A,B)$, we need an exhaustive set of rules involving C. Furthermore, it is possible that the rule base contains rules which overlap and rules that are generalizations of others. To identify the exact conditional probability, the dependence relations between rules have to be known. This relation can be analyzed from data, but if data is not available then there is no reliable mechanism to determine the relation between two overlapping rules and deduct the common influence.

Even if the probability calculated from each rule may not be accurate, it is possible, however, to find the bounds for the conditional probability. If one of the rules provides conditional probability (or belief) x, then we know that the correct conditional probability is at least x. It is possible that some other unspecified rule exists which increases the conditional probability, but given that the rule is correct, the ultimate conditional probability should lie between x and 1. The evidence (either explicitly specified or residual) forms the belief for the hypothesis and is used as a lower bound for the derivation of the actual conditional probability values. We cannot use a similar approach for the calculation of upper bounds as individual rules do not specify an estimate of the counter evidence. The system presented therefore assigns the default value of 1 for the upper bound for each entry in the conditional probability table unless we have additional knowledge to refine the bound.

Assuming *P(C)=0.5* (as estimated from data) and using equation (7), the bound for $P(C|A,B)$ can be calculated from rule (5) to be $(0.95, 1)$. From residual rule (6) the bound for $P(C|\neg A,B)$ can be estimated as $(0.635, 1)$. From the other residual rules derived from (5), the bounds for $P(C|A,\neg B)$ and $P(C|\neg A,\neg B)$ can be derived as $(0.054, 1)$ and $(0.18, 1)$.

Intuitively, it can be argued that the upper bound is unlikely to be 1. If any entry in the conditional probability table has a value of 1, then we have potentially a rule with stronger predictive power than the existing rules in the rule base. This also means that the expert would have most likely specified this rule. But since it cannot be assumed that the expert will always specify the complete set of highly predictive rules, we have to look for other concrete ways to refine the upper bound. From an evidence standpoint, the plausibility in a belief interval of a hypothesis is given by subtracting the belief we have in its complement from 1. If two hypotheses are complementary, we can use one of them to determine the plausibility of the other.

E. Constructing a Consistent Conditional Probability Table

Once we have a set of probabilistic bounds for entries in the conditional probability table for the partial Bayesian Network, the next step is to determine a set of probability assignments that are consistent with the individual bounds and the axioms of probability. All the nodes of the Bayesian Network which are not involved in any rules are assigned a probability distribution with uniform Dirichlet priors.

A consistent value for a conditional probability entry lies between the lower and upper probabilistic bound (inclusive) derived from the rules. Additionally, the sum of the entry and its negation must be 1. E.g. $P(C|A,B) + P(\neg C|A,B) = 1$ must be true. If there are two rules, one of which is the superset of other we will have an additional constraint to enforce the marginalization rule, i.e. for rules like *IF A AND B AND C THEN D* and *IF B AND C THEN D* we have an additional constraint to ensure that

$$\text{P(D|B,C)=P(D|A,B,C)P(A)+P(D|}\neg\text{ A,B,C)P(}\neg\text{ A)}$$

where $P(A)$ and $P(\neg A)$ are constants calculated from data. For any rules that overlap, such as *IF A AND B THEN D* and *IF B AND C THEN D*, we have two additional constraints:

- P(D|A,B)=P(D|A,B,C)P(C)+P(D|A,B,¬ C)P(¬ C)
- P(D|B,C)=P(D|A,B,C)P(A)+P(D|¬ A,B,C)P(¬ A)

The values of $P(A), P(\neg A), P(C)$ and $P(\neg C)$ are again calculated from data. If any of the prior probabilities cannot be learned from the data they are assumed to be uniformly distributed. Additionally, to ensure that the consistent values fall as much in the middle of the probabilistic bound as possible, some slack variables δ_X are introduced whose absolute values lie in the range of half of the width of the probabilistic bound.

Thus, from the rule base we can derive a set of constraints for the conditional probability entries. The next step is to assign consistent values that satisfy these constraints. This can be achieved using either linear programming or by sampling the probability space. The linear programming formulation for rule (5) using the estimated bounds in III-D is given by:

Maximize:

$$P(C|A,B) + P(\neg C|A,B) + P(C|\neg A,B) + P(\neg C|\neg A,B) + P(C|A,\neg B) + P(\neg C|A,\neg B) + P(C|\neg A,\neg B) + P(\neg C|\neg A,\neg B)$$

Subject to:

- $P(C|A,B) + P(\neg C|A,B) = 1$
- $P(C|\neg A,B) + P(\neg C|\neg A,B) = 1$
- ...
- $1 - P(C|A,B) + \delta_{P(C|A,B)} = P(C|A,B) - 0.9$
- $1 - P(C|\neg A,B) + \delta_{P(C|\neg A,B)} = P(C|\neg A,B) - 0.27$
- ...

F. Learning the Complete Bayesian Network

From the previous steps, a partial Bayesian Network structure and a consistent conditional probability table for each node involved in the rules are obtained. If a node is not used in any rule, then it is initialized with uniform Dirichlet priors. For each rule, the consequent becomes a child and the antecedents become the parents. To further complete and adjust the network structure, available data can be used in the context of any structure learning algorithm [3], bootstrapped with the partial Bayesian Network obtained from the rules.

IV. Explanation of Predictions

Once a Bayesian Network has been constructed, it can be used for predictions. For example, the system can be used for classification where, given a set of records, inference is used to find the most likely value for a target random variable. Similarly, the method can be used interactively where the user enters observations and is interested in the distribution of specific unobserved variables.

Once the system has a distribution of the variable, it has to explain the prediction to the user. Doing this purely from the Bayesian Network is very difficult as explanations in terms of the network's probability distribution are exceedingly difficult for users (and even experts) to understand. On the other hand, knowledge encoded in the rules should be understandable to the user. As a consequence, the system presented here derives explanation for Bayesian Network predictions using the original rules in the system. The system performs the inference and finds the rules in the rule base whose consequent matches the most likely value of the predicted variable. The system finds the Bayes factor [14] for each of the matching rules. A higher value of the Bayes Factor means that the rule is strongly supported by the data and has a higher likelihood of occurrence than its complement. The Bayes Factor (BF) for a rule of the form *If A then B* is defined as:

$$BF(A \rightarrow B) = \frac{P(B|A)}{1 - P(B|A)} \tag{8}$$

The Bayes Factor is calculated for each matching rule and the rules are sorted in descending order based on their Bayes Factor. The rule with the highest Bayes Factor can be considered as the strongest rule. The system will use the rules (from strongest to weakest) in explaining the rationale of the prediction. In the case of overlapping rules, it is possible that multiple rules apply to a single record. The system takes the rule with highest Bayes Factor and gives the user the percentage of the data explained by the rule. After that, all the records where the strongest rule applied are removed. The other rules are considered to augment the primary rule and the process is continued.

V. Rule Evaluation and Mining

A. Detecting Potential Incorrect Rules

The incorporation of data into the network structure and parameters can lead to conditional probability entries that are inconsistent with the bounds derived from the rules. This, in turn, can indicate that the certainty and importance factors or, in extreme cases, even a rule might be incorrect. To use this, the learned Bayesian Network structure and parameters are compared with the original Bayesian Network that is used as a seed. In particular, the learned probability values are compared to the bounds that were derived from the rules. If the learned values fall outside the bounds by more than a tolerance value, this could indicate a potential problem with the corresponding rule which could be verified by the expert. This is an inexpensive way to determine any issues in rules as it uses the original probability bounds to determine deviations.

An alternate, but expensive, method which would identify rule conflicts in more detail and could attempt to automatically resolve them is to perform structure learning for different subsets of rules. For each subset this corresponds to ignoring some subset of the rules and learning a new structure with the reduced set of rules. If any of the structures learned have a higher score than the structure learned with all the rules, the potentially incorrect rule can be identified whose veracity can then be verified with the expert.

B. Suggesting New Rules

In a typical scenario, most of the evidence can be explained by the rules available in the dataset. If for some prediction the system was not able to use rules to explain a significant amount of the evidence, this indicates a potential scenario to mine for new rules. In this case, the system uses lack of sufficient rules for explanation to mine new rules. Association rule mining [15] is used on the relevant records to identify new rules which can explain the prediction.

The new rules that are mined are compared with the existing rules to make sure that they are not a subsets or supersets of the existing rules. The system also calculates the certainty factor of the new rules using Heckerman's formula and assigns default importance values of 1 to the antecedents. The set of rules which have a certainty factor above some threshold are suggested to the user who may potentially accept or reject them. Accepted rules are added to the rule set and are used for future explanations. They do not result in any immediate alteration of the network structure or parameters.

C. Incremental Learning

Some of the domains in which the approach presented in this paper is going to be applied have a steady stream of incremental data. In this case, the system uses rules, certainty factors and the available data to learn the initial structure. As more data becomes available, the existing Bayesian Network must be adjusted so that it still fits the data. This incremental learning happens periodically. The new data can result in a modification of parameters and potentially even the structure. Currently, the system performs parameter estimation of the conditional probability table for the existing structure using the data. In other words, it does not start the structure learning from scratch each time new data arrives. However, if there is a large divergence between the model before and after learning from the incremental data, we can make a decision to perform structure learning again.

VI. Experiments

The approach was implemented in the *R* language using the *bnlearn* package [16]. The dataset used is *Alarm network* which was introduced in [10] and designed to monitor patients in an intensive care unit. The network consists of 37 discrete variables whose cardinality ranges from two to four. The variables represent observations from an intensive care unit, including the patients' history and device readings.

The rules for the alarm domain were mined from the data using association rule mining [15]. For each of the rules, the values of support, confidence and certainty factor were calculated using Heckerman's formula [7]. The values of prior and conditional probability used in Heckerman's formula were obtained from the data. The rules with the highest values for these factors were randomly added to the rule base. If the data is not available, a uniform prior is assumed. For the alarm network, a dataset of $20,000$ records is used for training. A separate dataset of $10,000$ records sampled from the original alarm network [10] was used for testing.

Two measures were used to compare the desirability of the candidate networks. The first measure used is the BIC score [3] of the candidate network using the testing dataset. The second is the classification error (CE) [3] which is defined as the ratio of incorrectly classified records to the total number of records.

A. Bootstrapping Bayesian Network from Rules

The first experiment demonstrates bootstrapping of an initial network from rules that were specified with valid certainty factors and importance parameters. Consider a simple rule, *IF Left Ventricular Failure(LVF)=FALSE(0.99) AND Hypovolemia(HYP)=FALSE(0.8) THEN Stroke Volume(STKV)=NORMAL(0.9)*. Using the methods described here, the derived related rules and their residual evidence are:

- *If LVF=FALSE AND HYP=TRUE THEN STKV=NORMAL (0.18)*
- *If LVF=TRUE AND HYP=FALSE THEN STKV=NORMAL (0.009)*
- *If LVF=TRUE AND HYP=TRUE THEN STKV=NORMAL (0.018).*

Probability bounds for the conditional probability table entries and a consistent assignment found using Linear Programming are shown in Table I. The network learned is shown in Figure 1. We can see that the structure of the Bayesian Network learned from data is influenced by the rule specified.

TABLE I
Bootstrapped Conditional Probability Table for STKV

LVF	HYF	Certainty Factors	Probabilistic bounds for STKV	Consistent value of probability	Probability after learning
False	False	0.9	[0.85,1]	0.93	0.89
False	True	0.18	[0.54,1]	0.77	0.51
True	False	0.009	[0.08,1]	0.53	0.04
True	True	0.018	[0.11,1]	0.56	0.01

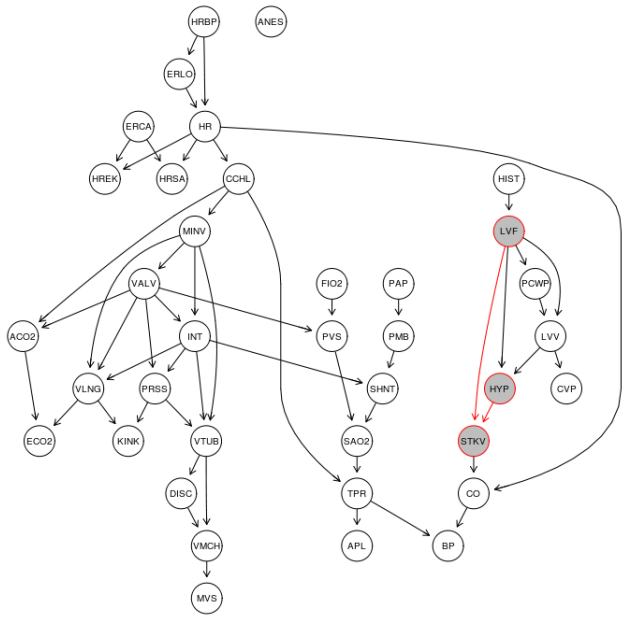

Fig. 1. Network learned from single rule

B. Impact of Rules on Training Dataset Size

The second experiment evaluates the benefits of bootstrapping the network using rules. We show that compared to a network that learns its entire structure from data, the bootstrapped network needs a significantly smaller training dataset before it can make meaningful predictions. The impact of rule base size is analyzed by comparing networks learned from rule bases of different sizes.

To evaluate the effect of additional data, 1000 records were randomly selected from the training dataset and used to learn two networks - one whose structure is bootstrapped from the rules (network1) and another with an empty initial structure (network2). The networks were compared according to their BIC score, size and classification error on the testing dataset. This process was repeated 26 times and the average value of the scores is noted. The whole experiment was repeated for partial datasets whose size varies from $2,000$ to $20,000$. The detailed results in terms of BIC score and classification error are shown in Table II and Table III, respectively. It can be seen that when the network is bootstrapped with rules, it needs significantly less data to learn the final structure and provide predictions that have classification error comparable to the other Bayesian Network that was seeded with empty structure and used the entire dataset for training.

C. Identifying Incorrectly Specified Rules

In the final experiment, we show how the system can identify potentially incorrect rules based on the deviation between initial and learned Bayesian Networks.

Consider a rule base with a single incorrect rule: *IF Pulmonary Artery Pressure(PAP)=NORMAL(0.9) AND Analgesia(ANES)=TRUE (0.9) THEN Left Ventricular Failure(LVF)=TRUE(0.9)*. The probabilistic bounds and estimated

TABLE II
SCORE AND COMPLEXITY OF NETWORKS

Dataset Size	Network bootstrapped with rules		Network seeded with empty structure	
	# Edges	BIC Score	# Edges	BIC Score
1000	44	-112605.4	46	-113218.3
2000	45	-111472	47	-112666.2
3000	47	-111324.3	48	-112601.8
4000	46	-110896.3	47	-112117.9
5000	47	-110855.9	47	-112117.9
10000	48	-110486.3	52	-111246.5
20000	50	-110500	53	-111281.5

TABLE III
CLASSIFICATION ERROR OF NETWORKS

Dataset Size	Mean CE for network1	Mean CE for network2	Statistics		
			Mean of difference in CE	Std Dev of difference in CE	p-value for t(25)
1000	0.176	0.269	0.093	0.021	2.2E-16
2000	0.173	0.263	0.091	0.023	2.2E-16
3000	0.173	0.220	0.047	0.017	1.9E-13
4000	0.173	0.186	0.013	0.031	0.05
5000	0.175	0.176	0.001	0.01	0.6011
10000	0.173	0.174	0.001	0.009	0.8448
20000	0.174	0.175	0.001	0.003	0.0562

and learned values of corresponding conditional probability table entries for the rule and its derived related rules are shown in Table IV. It can be seen that the learned probability for the rule after incremental learning with the training data differs significantly from the bounds. This provides an indication that the rule might be incorrect (e.g. that it has an incorrect certainty factor or represents an incorrect rationale). Similarly, deviations from the bounds of derived rules can indicate potentially incorrect importance factors for a rule. Combined with incremental learning, this allows the approach to derive a fine-tuned model that can transcend inexactness in the originally specified rule parameters and provide feedback to the expert.

TABLE IV
CONDITIONAL PROBABILITY ENTRIES FOR LVF

PAP	ANES	Certainty Factors	Probabilistic bounds for LVF	Consistent initial probability	Probability after learning
Normal	True	0.9	[0.6,1]	0.84	0.04
Normal	False	0.09	[0.059,1]	0.48	0.05
Low	True	0.09	[0.059,1]	0.48	0.05
Low	False	0.09	[0.0508,1]	0.48	0.05
High	True	0.09	[0.0508,1]	0.48	0.03
High	False	0.01	[0.0508,1]	0.48	0.06

VII. CONCLUSION

Certainty factor based expert systems and Bayesian Networks are two popular frameworks to perform uncertain reasoning. In this paper, we have proposed an approach that combines the ability to efficiently elicit expert knowledge and generate user understandable explanations using rules with the automated reasoning capabilities of the Bayesian Network. The domain knowledge is expressed using rules with certainty factors that were augmented with importance parameters for antecedents. Using the techniques described in the paper, the residual evidence for rules that are variations of the expert specified rules were derived and used to construct a partial network structure and its corresponding conditional probability tables. This partial network is used to bootstrap a Bayesian Network learned from data. This network can then be used to perform automated inference and generate explanations using the expert specified rules. Other issues like incremental learning, identifying incorrect rules and proposing new rules are also addressed, resulting in a consistent framework that combines the advantages of rule based systems with the data driven capabilities of Bayesian Networks. The experiments show that with this approach a properly bootstrapped Bayesian Network needs substantially less data to learn the final structure and provide inference. The experiments also show identification of incorrectly specified rules using the training dataset.

ACKNOWLEDGMENT

This work was supported in part by grant number 1G08LM009262 from the National Library of Medicine.

REFERENCES

[1] E. H. Shortliffe and B. G. Buchanan, Eds., *Rule-Based Expert Systems - The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison–Wesley, 1984.
[2] D. Heckerman, "The certainty-factor model," in *Encyclopedia of Artificial Intelligence*, 1992.
[3] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
[4] D. Heckerman and E. H. Shortliffe, "From certainty factors to belief networks," in *Artificial Intelligence in Medicine 4:3552*, 1992.
[5] M. Henrion, "Should we use probability in uncertain inference systems?" in *Proceedings of the Cognitive Science Society Meeting*, 1986.
[6] J. Gordon and E. H. Shortliffe, "The dempster shafer theory of evidence," in *Readings in Uncertain Reasoning*, G. Shafer and J. Pearl, Eds., 1990.
[7] D. Heckerman, "Probabilistic interpretations for mycin's certainty factors," in *Uncertainty in Artificial Intelligence*, 1986.
[8] F. J. Dez, J. Mira, E. Iturralde, and S. Zubillaga, "Diaval, a bayesian expert system for echocardiography," *AI in Medicine*, vol. 10, 1997.
[9] A. Onisko, P. J. F. Lucas, and M. J. Druzdzel, "Comparison of rule-based and bayesian network approaches in medical diagnostic systems," in *AI in Medicine in Europe*, 2001.
[10] I. Beinlich, G. Suermondt, R. Chavez, and G. Cooper, "The alarm monitoring system," in *Sec. European Conf. on AI and Medicine*, 1989.
[11] M. Korver and P. J. F. Lucas, "Converting a rule-based expert system into a belief network," *International Journal of Medical Informatics*, 1993.
[12] J. S. R. Jang and C.-T. Sun, "Neuro-fuzzy modeling and control," *Proceedings of The IEEE*, vol. 83, 1995.
[13] R. Fuller, "Introduction to neuro-fuzzy systems," 1999.
[14] C. Yuan and T.-C. Lu, "A general framework for generating multivariate explanations in bayesian networks," in *AAAI*, 2008.
[15] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *SIGMOD*, 1993.
[16] M. Scutari, "Learning Bayesian Networks with the bnlearn R Package," *Journal of Statistical Software*, 2010.