Copyright

by

Sowmya Ramachandran

1998
Theory Refinement of Bayesian Networks with Hidden Variables

by

Sowmya Ramachandran, B.Tech., M.S.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

May 1998
Theory Refinement of Bayesian Networks with Hidden Variables

Approved by
Dissertation Committee:

____________________

____________________

____________________

____________________

____________________
To my grandfather
Acknowledgments

Getting a doctorate is an exciting, challenging, and at times difficult process. I have been fortunate to have an extensive network of family, friends, and professional colleagues to support me throughout this process. I am grateful to Ray Mooney for supervising my research efforts, and all the invaluable advice he has given me through the years on technical matters and on presentation skills. I am also grateful to him for supporting me through the following grants: NSF grant number IRI-9704943, and NSF grant number IRI-9310819. My research was also supported by teaching assistantships in the Department of Computer Sciences. I would also like to thank Ben Kuipers for financial support and advice on what research is all about. I would like to thank Greg Provan, Risto Miikkulainen, Joydeep Ghosh, and Ben Kuipers for serving on my committee and giving me insightful feedback on my research. I am also grateful to Haym Hirsh for all the support and encouragement through the years.

I am very grateful to Bobby Blumofe, Lorenzo Alvisi, Mike Dahlin, Calvin Lin and other folks at the LESS lab for letting me use their computing facilities for research. The multiprocessor computing facilities in this lab were made available through a generous equipment donation from Sun Microsystems. But for these machines, I would still be a graduate student.

My tenure as a graduate student was enlivened and enriched by all my fellow graduate students, who have shown me that there is more to life than graduate school. I would like to thank James Lester, Charles Callaway, Tim Collins, and Bradley Richards for all the juggling and dessert, to Jeff Rickel, and Siddarth Subramanian for enlivening our graduate student lives with pranks and spiders. It has been very nice sharing an office with Cindi Thompson and Mike Hewett, and knowing that I can always count on them for support, friendship, and proof-reading. Thanks also to Marty and Coquis Mayberry for Pictionary, Scrabble, and several interesting discussions. I am also very grateful to Ajita John, Gita Mani, Ashok Balivada, and Tom Lee for being such good friends. Other students who have supported in my research include Jeff Mahoney, John Zelle, Paul Baffes, Rich Mallory, Tara Estlin, Dave Moriarty, Mary Elaine Califf, Shailesh Kumar and many others. I would also like to take this opportunity to remember Bert Kay, who passed away recently, and his contributions to my life as a graduate student. Bert was a wonderful role model, a patient teacher, and a very kind friend. I am very grateful to have had a chance to know him. I am especially thankful to Rick Froom for standing by me these last few years, for showing me a lighter and brighter side of life, and for all the support and encouragement.

Finally, I would like to thank my parents and grandparents for their patience and understanding, and for providing me with a place to call home. Without their love and support, I would not be where I am today. Finally, I would like to thank my sister and her two beautiful children for making me laugh and reminding me that life is delightful.
Theory Refinement of Bayesian Networks with Hidden Variables

Publication No. __________________

Sowmya Ramachandran, Ph.D.
The University of Texas at Austin, 1998

Supervisor: Raymond J. Mooney

Research in theory refinement has shown that biasing a learner with initial, approximately correct knowledge produces more accurate results than learning from data alone. While techniques have been developed to revise logical and connectionist representations, little has been done to revise probabilistic representations.

Bayesian networks are well-established as a sound formalism for representing and reasoning with probabilistic knowledge, and are widely used. There has been a growing interest in the problem of learning Bayesian networks from data. However, there is no existing technique for learning or revising Bayesian networks with hidden variables (i.e. variables not represented in the data), that has been shown to be efficient, effective, and scalable through evaluation on real data. The few techniques that exist for revising such networks perform a blind search through a large space of revisions, and are therefore computationally expensive.

This dissertation presents BANNER, a technique for using data to revise a given Bayesian network with Noisy-Or and Noisy-And nodes, to improve its classification accuracy. Additionally, the initial network can be derived directly from a logical theory expressed as propositional Horn-clause rules. BANNER can revise networks with hidden variables, and add hidden variables when necessary. Unlike previous approaches to this problem, BANNER employs mechanisms similar to those used in logical theory refinement techniques for using the data to focus the search for effective modifications to the network. It can also be used to learn networks with hidden variables from data alone. We also introduce BANNER-Pr, a technique for revising the parameters of a Bayesian network with Noisy-Or/And nodes, that directly exploits the computational efficiency afforded by these models.

Experiments on several real-world learning problems in domains such as molecular biology and intelligent tutoring systems demonstrate that BANNER can effectively and efficiently revise networks to significantly improve their accuracies, and thus learn highly accurate classifiers. Comparisons with the Naive Bayes algorithm show that using the theory refinement approach gives BANNER a substantial edge over learning from data alone. We also show that BANNER-Pr converges faster and produces more accurate classifiers than an existing algorithm for learning the parameters of a network.
Contents

Acknowledgments v

Abstract vii

Chapter 1 Introduction 1

Chapter 2 Background 7
  2.1 Bayesian Networks: An Overview ................................................. 7
    2.1.1 Polytree Vs. Loops ......................................................... 9
    2.1.2 Noisy-Or and Noisy-And Models ....................................... 9
  2.2 Multi-layered Feed-Forward Neural Networks .................................. 13
  2.3 Theory Refinement ........................................................................ 15
    2.3.1 Logic-Based Approaches .................................................. 16
    2.3.2 Hybrid Approaches ................................................................ 18
  2.4 Naive Bayes .................................................................................. 20
  2.5 Learning Bayesian Networks ........................................................... 22

Chapter 3 BANNER: An Overview 27
  3.1 Overview of the Algorithm .............................................................. 27
  3.2 Limitations of BANNER, BANNER-Pr and C-APN ................................ 28
  3.3 Converting from a Logical Theory to a Bayesian Network ...................... 31

Chapter 4 Parameter Revision: Learning Conditional Probabilities 33
  4.1 BANNER-Pr .................................................................................. 33
    4.1.1 Overview of BANNER-Pr .................................................... 33
    4.1.2 Backpropagation ................................................................ 34
    4.1.3 Stopping Criteria .................................................................. 35
  4.2 APN and C-APN ............................................................................. 35
  4.3 Experimental Evaluation .................................................................. 38
    4.3.1 DNA Promoter Recognition .............................................. 38

Chapter 5 Structure Revision 44
  5.1 Augmenting the Network with Leak Nodes ...................................... 48
  5.2 Blame Assignment ......................................................................... 48
  5.3 Revision Operators ........................................................................ 51
Chapter 6 Empirical Evaluation

6.1 Experimental Methodology ........................................ 60
6.2 DNA Promoter Recognition ........................................ 61
   6.2.1 Experiments with the 106-example Data Set .................. 62
   6.2.2 Experiments with the 468-example Data Set .................. 66
6.3 DNA Splice-Junction ............................................... 67
6.4 C++ Tutor ...................................................... 70
6.5 Chess End-games: Induction from Many Examples ................. 75
6.6 Brain Disorders: Revising an Abductive Theory ................ 78
6.7 Evaluation of the Structure Revision Component ................. 81
   6.7.1 Deleting Intermediate Concept Minus_35 .................... 82
   6.7.2 Deleting Intermediate Concept Contact ..................... 85
6.8 Summary of Results .............................................. 86

Chapter 7 Related Work ............................................. 91

7.1 Learning Bayesian Networks ...................................... 91
   7.1.1 Parameter Learning ........................................ 92
   7.1.2 Structure Learning ......................................... 93
7.2 Theory Refinement ................................................ 95
   7.2.1 Logic-Based Approaches ................................... 95
   7.2.2 Hybrid Approaches ......................................... 96

Chapter 8 Future Work ............................................... 97

8.1 Experimental Study .............................................. 97
8.2 Extensions to BANNER-Pr ......................................... 98
8.3 Extension to General Nodes ....................................... 99
8.4 Integrating other Bayesian Techniques .......................... 99
8.5 Extension to Multi-valued Variables .............................. 100

Chapter 9 Conclusion ................................................ 101

Appendix A Derivation of the Gradients for BANNER-Pr ............ 105
   A.1 Gradient for a Noisy-Or Node ................................. 105

Appendix B Initial Domain Theories .................................. 109
   B.1 DNA Promoter Recognition .................................... 109
      B.1.1 Input Features .......................................... 109
      B.1.2 Categories ............................................. 109
      B.1.3 Domain Theory .......................................... 109
B.2 Splice Junction Recognition .............................................. 110
  B.2.1 Input Features ................................................... 110
  B.2.2 Categories ...................................................... 110
  B.2.3 Domain Theory .................................................. 110
B.3 C++ Tutor .............................................................. 112
  B.3.1 Input Features ................................................... 112
  B.3.2 Categories ...................................................... 112
  B.3.3 Domain Theory .................................................. 112
B.4 Brain Disorders ......................................................... 114
  B.4.1 Input Features ................................................... 114
  B.4.2 Categories ...................................................... 115
  B.4.3 Domain Theory .................................................. 115

Bibliography ................................................................. 121

Vita ..................................................
Chapter 1

Introduction

Theory refinement, or theory revision, is an area of research that has grown out of work on inductive and explanation-based learning. It is based on the idea that, when only limited data is available, biasing an inductive learner with prior knowledge can improve learning by focusing the search space of possible hypotheses. The more complex the domain, the more the advantage of such a bias. Thus, theory refinement systems assume that the learner has an initial imperfect knowledge base (usually obtained from a domain expert) which is then inductively revised to fit the data. Many techniques have been developed for revising knowledge bases represented in various languages such as propositional Horn-clause logic (Ourston & Mooney, 1994; Koppel, Feldman, & Segre, 1994) and first-order Horn-clause logic (Cohen, 1992; Wogulis & Pazzani, 1993; Wogulis, 1994; Richards & Mooney, 1995; Brunk, 1996). Even within the connectionist framework there are techniques, such as KBANN (Towell & Shavlik, 1994; Opitz & Shavlik, 1993), that explicitly bias a neural network with an initial theory. Experiments on real-world data have demonstrated that revising an approximate domain theory produces more accurate results than learning from training data alone (Ourston & Mooney, 1990; Thompson, Langley, & Iba, 1991; Towell, Shavlik, & Noordewier, 1990).

Although existing techniques for revising theories differ in their approaches, they have many things in common. One such common feature is that most of them are geared towards learning theories for the specific task of classification. They use the classification accuracy of the theory as an indication of the correctness of the current theory, and make revisions to a theory only if it misclassifies some of the data. A second, very important feature that they share in common is that, rather than searching blindly through the entire space of possible revisions, they analyse the classification errors resulting from the theory, and use this information to restrict the space of possible revisions to be considered. These features combine to make these algorithms directed and, therefore, efficient.

Research in theory refinement has, however, been limited to logical and connectionist representations. Little has been done to address the problem of revising probabilistic knowledge. Intelligent systems need mechanisms to represent and reason with uncertain knowledge. Uncertainty in a domain or a task may arise due to incomplete knowledge about the state of the world or due to true randomness in the domain. Examples of tasks involving uncertainty include medical diagnosis, and plan recognition, to name but a few. Thus, we need languages for representing uncertainty. It is also desirable to have techniques for learning and revising knowledge represented
Several approaches have been developed for representing and reasoning with uncertainty. These include *default reasoning* (Reiter, 1980), *fuzzy logic* (Zadeh, 1965, 1981), *certainty factors* (Buchanan & Shortliffe, 1984), *Bayesian networks* (Pearl, 1988), and *Dempster-Shafer calculus* (Dempster, 1968; Shafer, 1976). Of these, default reasoning uses a symbolic representation of uncertainty, while the rest use numeric representations. The Bayesian network approach stands out as the only one that is directly grounded in probability theory, which has long been a widely accepted formalism for representing uncertainty. The rest of the approaches invent their own calculi, which makes their semantics less clearly defined.

Among the numeric representations, *certainty factors* have played a significant role in the history of uncertain reasoning. The use of *certainty factors* is exemplified in MYCIN, an expert system to recommend treatment for bacterial infection (Shortliffe & Buchanan, 1975; Buchanan & Shortliffe, 1984). MYCIN is a rule-based system, where the rules are augmented with *certainty factors* or numbers that indicate their credibility. The certainty of a conclusion of a rule is computed as a function of the certainty of the premises and the credibility of the rule. Although they have been successfully applied in a number of domains, the rules for combining certainty factors are *ad hoc* and provide no mathematical guarantees, unless unrealistic independence assumptions are made (Heckerman, 1986).

Bayesian networks (Pearl, 1988) represent uncertainties as probabilities of events in the world, and provide ways for representing dependencies between variables explicitly. A Bayesian network is a directed acyclic graph (DAG), where the nodes are random variables representing events, and the links represent dependencies between the variables. Associated with the DAG are numeric parameters that define the dependencies precisely. A variety of techniques for reasoning with such networks have been developed that use theoretically sound mechanisms for combining probabilities, and accounting for dependencies. Their strong grounding in probability theory makes Bayesian networks a particularly attractive formalism for representing knowledge. Many real-world applications, especially in medical diagnosis, now use this representation (Pradhan, Provan, Middleton, & Henrion, 1994; Burnell & Horovitz, 1995; Fung & Del Favero, 1995). However, like all numerical representation schemes, Bayesian networks suffer from the knowledge acquisition problem. Not only is it difficult to formulate the underlying structure of the DAG, it is especially difficult to specify the dependencies between the variables in precise numeric terms. Therefore, it would be useful to have efficient techniques that learn such networks from data. Not surprisingly, there is now a growing interest in this problem.

The task of learning a Bayesian network can be divided into two subtasks: one of learning the DAG structure of the network, and the second of determining the parameters. Within the general framework of inducing Bayesian networks, we can envision the following scenarios.

1. **Known structure, complete data**: In this scenario, the structure of the network is given and assumed to be correct and the data includes observations of all the variables in the network. The task here is to learn the parameters of the network from data.

2. **Known structure, incomplete data**: Here, the structure of the network is given and assumed to be correct. The data, however, does not include observations of every variable in the network. This includes situations where some variables are missing values for part of the data, and
situations where the values of some variables are never specified in the data. Variables whose values are never included in the data are called hidden variables. Again, the task is to induce the parameters of the network. However, the task is more complicated in this case due to the presence of hidden variables and variables with missing values.

3. **Unknown structure, complete data:** In this case, neither the structure, nor the parameters of the network are known. We can, however, assume that the data is complete, and there are no hidden variables. The task here is to learn both the structure and the parameters of the network.

4. **Unknown structure, incomplete data:** This is the most general learning scenario where the structure of the network is unknown and there are hidden variables.

The first of these is fairly straightforward. A common approach is to use the maximum likelihood estimates for the parameters, which in the case of complete reduces to a function of the relative frequencies of occurrences of the values of the variable (Spiegelhalter & Lauritzen, 1990).

The problem of learning the parameters for a network with a known structure, given incomplete data, has also received some attention. Many statistical techniques like Gibbs sampling (Geman & Geman, 1984) and EM (Dempster, Laird, & Rubin, 1977; Lauritzen, 1995) can be used in the context of Bayesian networks. Russell, Binder, Koller, and Kanazawa (1995) have proposed an approach that attempts to optimize the probability of the data given the network using a gradient descent algorithm.

The learning problem addressed by Cooper and Herskovits (1992) falls in the third category. Their technique, implemented in a system called K2, uses a scoring metric to hill climb through a space of possible Bayesian networks to find one that is the most probable given the data. A number of variations and improvements to this approach have since been proposed (Buntine, 1991; Heckerman, Geiger, & Chikering, 1994; Provian & Singh, 1994).

The fourth scenario above, namely that of learning a Bayesian network with an unknown structure, given incomplete data, is by far the most difficult, especially when there are hidden variables. Most of the above techniques could be adapted to discover hidden variables, but at a great cost involving brute force search. Connolly (1993) has proposed using clustering techniques (Fisher, 1987) to discover hidden variables. However, this technique can only learn tree-structured networks. Very recently, Friedman (1997) has proposed a technique, called MS-EM, that extends EM (Dempster et al., 1977; Lauritzen, 1995) to learn the structure of a network with hidden variables. However, it is not flexible because it requires that the number of hidden variables to be considered by the network be specified ahead of time. BKD (Ramoni & Sebastiani, 1997) is another recent algorithm, based on K2, for learning Bayesian networks from incomplete data. However, it cannot discover hidden variables.

Thus, while researchers have a grasp on some aspects of learning Bayesian networks, the problem of learning Bayesian networks with unknown structures and hidden variables still poses a tough challenge. However, theory refinement techniques like those proposed by Ourston and Mooney (1994), Opitz and Shavlik (1993), Mahoney and Mooney (1993) have been successful in addressing similar issues. The bias that such techniques provide in the form of prior knowledge, and their techniques for using the data to focus the search for accurate theories, make them very
efficient. We had mentioned earlier that there has been very little research into using theory refinement techniques to learn Bayesian networks. Lam and Bacchus (1994b) have a technique for incrementally refining a Bayesian network using the Minimum Description Length principle (Rissanen, 1978). Buntine (1991) has proposed a technique for revising a Bayesian network efficiently, using scoring metrics similar to that proposed by (Cooper & Herskovits, 1992). However, neither of these techniques can revise networks with hidden variables. While algorithms like K2 (Cooper & Herskovits, 1992), BKD (Ramoni & Sebastiani, 1997) and MS-EM (Friedman, 1997), that build a Bayesian network iteratively, can also be used for theory refinement, they perform a blind search through all possible single revisions to the network, and are thus computationally expensive. Mahoney and Mooney (1993) have a system for revising uncertain knowledge bases, but expressed in the form of certainty factors. Their system, RAPTURE maps the rules of the knowledge base into a neural network and uses connectionist methods to revise the certainty factors associated with the rules as well as the rules themselves. RAPTURE can also discover hidden variables not specified in the data.

All of the Bayesian network learning techniques described above optimize the probability assigned by the network to the data as a whole. However, in many learning situations the objective is to acquire knowledge for a particular task such as classification. In such situations, it is better to learn theories that are optimized for the particular task that it is meant to perform. Thus, it is desirable to have a learning algorithm that optimizes the classification accuracy of the network, when the objective is to build a Bayesian network for the purpose of classification. Friedman and Goldszmidt (1996) have experimentally demonstrated the advantage of learning to maximize classification accuracy. Specifically, the learning algorithm should try to learn a network that best estimates the probability distribution of the class variables conditioned on some evidence variables. Currently, there are very few existing techniques for learning Bayesian networks that optimize for classification.

The goal of our research is to use a theory refinement approach to learning Bayesian network classifiers with hidden variables. From the perspective of the four learning scenarios outlined earlier, this problem lies somewhere between scenarios 2 and 4. Thus, we assume that the learner is given a Bayesian network that may be incomplete or incorrect. We also assume that the learner is provided with data that may not include all of the variables in the network. The task is to use the data to improve the predictive accuracy of the network, modifying both its parameters and structure (adding hidden variables when needed). Unlike the previous approaches to revising Bayesian networks, we are specifically interested in developing mechanisms, similar to those employed by logical theory refinement techniques (Ourston & Mooney, 1994; Koppel et al., 1994; Cohen, 1992; Wogulis & Pazzani, 1993; Richards & Mooney, 1995; Brunk, 1996), for using the data to focus the search for effective modifications to the network.

Since general Bayesian networks are impractical for many large problems because the size of the conditional probability tables grows exponentially in the fan-in of a node, we focus on the problem of learning networks with Noisy-Or and Noisy-And nodes (Pearl, 1988; Pradhan et al., 1994). These are specialized models for representing dependencies that only require a linear number of parameters. We specifically chose these models because they are semantically close to logical ors and logical ands, thus making it possible to use knowledge expressed in the form of logical rules as an initial theory. Using such nodes, a knowledge base originally expressed as rules can
be mapped to an analogous Bayesian network and refined to improve its accuracy. Many existing knowledge bases are written in the form of rules, and many experts have become comfortable with this formalism. However, results in theory refinement show that the accuracy of such rule bases can be dramatically improved by mapping them to a representation that employs some form of uncertain reasoning or numerical summing of evidence (Towell & Shavlik, 1994; Kopp et al., 1994; Baffes & Mooney, 1993; Mahoney & Mooney, 1993).

Here, we present a system called Banner that, given a set of data and an initial approximate domain theory, produces a revised theory in the form of a Bayesian network with Noisy-Or/And nodes that accurately classifies the data. The initial theory may be in the form of propositional Horn-clause rules, or in the form of a Bayesian network with Noisy-Or/And nodes. Rather than search through the space of all possible revisions, it uses information in the data to select specific nodes and links in the network to be revised, which makes it efficient. Although designed for theory refinement, it can also be used to learn Bayesian networks with hidden variables inductively from data using a default initial network.

Banner uses a two-tiered approach, similar to that used by Rapture (Mahoney & Mooney, 1993, 1994). Given some data, Banner first tries to improve the Bayesian model by revising the parameters of the network. If the network still does not fit the training data, the structure of the network is modified to find the network with the highest predictive accuracy. Thus, Banner has two components: one for revising the parameters of the networks, and one to revise the structure.

In the following chapters, we will present the details of our technique, and provide experimental evaluations of its effectiveness. We will evaluate our system on the following real-world learning problems, some of which are standard benchmark problems used to evaluate theory refinement and inductive learning techniques: recognizing DNA promoters (Noordewier, Towell, & Shavlik, 1991), recognizing DNA splice-junctions (Noordewier et al., 1991), learning student models for a C++ tutor (Baffes, 1994), diagnosing brain disorders in human patients (Tuhrim, Reggia, & Goodall, 1991), and predicting the outcome of a chess end-game from board configurations (Shapiro, 1983, 1987). The first four of these problems have associated domain theories, represented as propositional Horn-clause rules, that do not have good predictive accuracies on the data and, therefore, need to be revised. The fifth problem does not have an initial domain theory and is intended to study the effectiveness of Banner in learning networks from scratch. In addition, we will evaluate the structure revision component on Banner by performing experiments on corrupted versions of the theory for recognizing DNA promoters. Through our experiments, we will demonstrate that our technique is effective in learning fairly large Bayesian networks with high classification accuracy. We will show that the performance of Banner on these domains is comparable to the best results obtained with other learning techniques. We will also show that the strategy of starting with an initial, approximately correct theory gives Banner an edge over systems that learn from scratch. The experiments with corrupted theories will demonstrate the effectiveness of our technique in revising such theories so as to significantly improve their accuracies.

The research presented here makes contributions to the field of Bayesian network learning as well as the field of theory refinement. From a Bayesian network learning perspective, Banner is a novel technique for revising a class of Bayesian network classifiers with hidden variables that can also add hidden variables when necessary. It can also be used to learn networks with hidden variables inductively from incomplete data. Our experiments on real-world data sets demonstrate
that this approach can efficiently revise large networks and produce highly accurate classifiers. Whereas previous techniques for revising Bayesian networks searched through the space of all possible revisions, our technique uses novel mechanisms for using the information in the data to guide the search for useful revisions, thus eliminating a large number of irrelevant revisions from consideration. Since the initial theory given to BANNER may be in the form of propositional Horn-clause rules, it also provides a direct mechanism for incorporating knowledge expressed as propositional Horn-clause rules into a Bayesian network. We also introduce a new technique for revising the parameters of a network with Noisy-Or/And nodes that directly exploits the efficiency afforded by these models, and is targeted towards learning classifiers by trying to optimize the conditional distribution of the class variables given the evidence. We show that this technique converges faster and produces more accurate classifiers than an existing algorithm for learning the parameters of a network.

From a theory refinement perspective, this dissertation presents a novel hybrid theory refinement system that combines good performance with comprehensibility. Early research in theory refinement focussed on purely symbolic representations, which are very comprehensible, but show poor accuracies on several real-world learning problems. Techniques that revise rules-bases by mapping them into representations that combine logical rules and some form of uncertain reasoning, or numerical summing of evidence, have been shown to produce significantly improved classifiers for many domains. Successful hybrid theory refinement systems include KBANN, a system for learning a class of multi-layered feedforward neural networks called Knowledge-Based neural networks, whose structures are determined by an initial logical theory (Towell & Shavlik, 1994), and RAPTURE, a system that revises logical theories by mapping then into certainty factors rule-bases (Mahoney & Mooney, 1993; Buchanan & Shortliffe, 1984). Although these techniques have been shown to learn highly accurate classifiers, such hybrid representations lack well-defined semantics and cannot be easily understood. Bayesian networks, on the other hand, are attractive as hybrid representations because they combine sound mechanisms for representing probabilities with a well-founded qualitative representation of the correlations between variables, and are grounded in probability theory. Several sound inference mechanisms have been developed to reason with such networks. Thus, BANNER can be viewed as a hybrid theory refinement system that learns representations with more clearly defined semantics than the representations learned by the other hybrid theory refinement systems mentioned above, while maintaining a comparable level of performance in terms of learning accurate classifiers, as will be demonstrated by our experiments.

This dissertation is organized as follows: Chapter 2 presents a brief introduction to various concepts that are central to our research, Chapter 3 presents an overview of BANNER, Chapter 3 discusses the parameter revision component, Chapter 4 discusses the structure revision component, Chapter 6 presents experiments that demonstrate the effectiveness of BANNER, Chapters 7 and 8 discuss related work and future directions respectively, and finally Chapter 9 summarizes our research and its contributions.
Chapter 2

Background

Our research builds upon several ideas and techniques. This chapter presents some of this background knowledge at a level of detail that will contribute to an easier understanding of later chapters. We will begin with an overview of Bayesian networks. Then we will look at multi-layered feedforward neural networks, followed by an overview of some techniques for theory refinement, and techniques for learning Bayesian networks.

2.1 Bayesian Networks: An Overview

Bayesian networks (Pearl, 1988) provide a formalism for representing probabilistic knowledge. They provide theoretically sound mechanisms for representing and reasoning with probabilistic dependencies between events. In general, a Bayesian network is a directed acyclic graph, whose nodes represent to random variables. Here, we use the convention of using upper-case letters to represent variables and lower-case letters to represent values associated with the variable. The links in the network represent dependencies between variables, such that two variables are assumed to be independent of each other if there is no undirected path between them, or if any of their common ancestors are instantiated. Associated with each node is a conditional probability table (CPT), which gives the probability of each value of the variable given each possible combination of values of its parent nodes. The CPTs define how the influences of the parents of a node interact to produce a combined influence on the node. Given a network with $n$ nodes and the associated CPTs, the probability of a conjunction of a particular assignment of values to the variables, i.e. $P(x_1, \ldots, x_n)$, can be calculated using the following formula:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(X_i))$$

where $P(x_i \mid Parents(X_i))$ is obtained from the CPT associated with variable $X$.

Figure 2.1 (Pearl, 1988) shows an example of a Bayesian network. Such a network may be used by a person to decide whether or not to respond to an alarm in his house. The nodes in the network represent the various events that are of relevance to the decision. Node $E$ represents the occurrence of an earthquake, node $R$ represents the announcement of an earthquake on the radio, and node $D$ represents the event of the person’s daughter calling him about the alarm. The links between the nodes represent the dependencies between the various events. For instance, the